Improving Security Through Analysis of Log Files Intersections

Kazimierz Kowalski and Mohsen Beheshti
(Corresponding author: Kazimierz Kowalski)

Computer Science Department, California State University Dominguez Hills
1000 Victoria Str., Carson, CA 90747 (Email: kkowalski@csudh.edu)

(Received June 27, 2006; revised and accepted Sep. 13, 2006)

Abstract

The paper discusses our research in development of general and systematic methods for intrusion prevention. The key idea is to use data mining techniques to discover repeated patterns of system features that describe program and user behavior. Server systems customarily write comprehensive activity logs whose value is useful in detecting intrusion. Unfortunately, production volumes overwhelm the capacity and manageability of traditional approach. This paper discusses the issues involving large-scale log processing that helps to analyze log records. Here, we propose to analyze intersections of firewall log files with application log files installed on one computer, as well as intersections resulting from firewall log files with application log files coming from different computers. Intersections of log files are substantially shorter than full logs and consist of records that indicate abnormalities in accessing single computer or set of computers. The paper concludes with some lessons we learned in building the system.

Keywords: Data mining, intrusion prevention, log files, security architectures

1 Introduction

In today's business environment almost all companies have their computers connected to the public Internet. As the number of companies with computers and services accessible to the Internet increases, a corresponding increase in the number of attacks against these businesses is also observed. Network-based attacks on business computers have been increasing in frequency and severity over the past several years. Consequently, many research efforts have concentrated on network intrusion detection techniques whose goal is to identify such attacks. For example, reports generated from the Computer Emergency Response Team Coordination Center [7] databases illustrate dramatic growth in reported incidents of security breach over the past years. Due to the fact that the numbers of attacks on the global Internet are increasing, it is critical for companies to secure their network and computers. This is especially true for corporations with businesses that are dependent on the Internet. In severe cases of security breach companies may lose business, and eventually become bankrupt, as a result of one successful attack [6].

Security attacks (or more neutrally security treats) come from different sources. Natural forces such as earthquakes, floods, etc can ruin essential information. Similarly, accidents such as water pipes breaks, fire, etc can damage business data. In those cases prevention of data loses, deal with fairly predictable scenarios of natural disasters or accidents. Completely different treats come from people known as intruders, e.g. unauthorized users of computers or services on some computers. There are external intruders, who are unauthorized users of the machines they attack, and internal intruders, who have permission to access the system with a number of restrictions [1]. Several techniques have been used to prevent unauthorized access to business data; some suitable to prevent the access by external and internal intruders, while others only prevent the access by external intruders. Users' authentication and data encryption are examples of techniques appropriate for both, external and internal intruders, while firewalls can prevent the access by external intruders. In this paper we concentrate on prevention access from external intruders.

It is understood that it is too risky to omit a firewall that separates a private local area network from the unrestricted global Internet; Majority of businesses have some sort of security policy in effect to prevent possible misuse of their system. The inclusion of intrusion detection systems is becoming more common in many organizations. According to the annual computer crime and security survey released by the Computer Security Institute (CSI) and Federal Bureau of Investigation (FBI), the number of companies that have intrusion detection systems has increased from 42 to 73 percent between the years 1999 and 2003. During this same period, the number of companies that have firewalls in place has increased from 91 to 98

percent [7].

The security of a computer system is compromised when an intrusion takes place. An intrusion can be defined [11] as "any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource". Intrusion prevention techniques, such as user authentication (e.g. using passwords or biometrics), avoiding programming errors, and information protection (e.g., encryption) have been used to protect computer systems as a first line of defense. The idea of the perfect system, that stops all attacks before they reach their target, is theoretically possible only with preventive solutions rather than just detective ones. Intrusion prevention alone is not sufficient because as systems become ever more complex, there are always exploitable weaknesses in the systems due to design and programming errors, or various "socially engineered" penetration techniques. The policies that balance convenience versus strict control of a system and information access also make it impossible for an operational system to be completely secure.

2 Related Work

In recent years, with the widespread use of Intranet and Internet, users have become more and more dependent on the services provided by networked systems where computer programs and potentially sensitive information are kept in (geographically) dispersed systems and exchanged over telecommunication facilities. Distributed systems have emerged to provide the means through which networked systems cooperate to process users' tasks in a seamless and efficient fashion. Such systems provide tremendous benefits to their users but also raise new challenges specifically, access control and instruction detection [12].

Access Control:
Security access control mechanisms play a key role in the overall structure of any security system. They are responsible for controlling the access permissions to system resources; i.e. determining who has access to which resource and with what type of access. Access control mechanisms rely on the authentication mechanisms to identify the users and ensuring that they are actually who they claim to be. The most common authentication method used to date is the user ID and password (or PIN number) combination, though other methods, such as bio-metric identification, have been used with varying degrees of success [12].

The authentication system is clearly the cornerstone of current security systems. It also constitutes one of their main weaknesses. Indeed, current security systems are built on the premise that once a user presents valid credentials to the authentication system (e.g. valid ID and password), they are granted access permission to all resources assigned to the user that they claim to be. Numerous studies [12], however, have shown that a large number (if not most) of security breaches are done by unauthorized users impersonating as authorized users (by guessing passwords or stealing them through various means). Other security breaches occur by circumventing the authentication system altogether, by exploiting security "holes" in the system. Once the authentication system is broken, the system and the information kept in it become wide open to unauthorized access and malicious usage. Moreover, because of the interdependencies among the various (computer and telecommunication) components of a distributed system, a security breach to one component can have repercussions throughout the system.

Some access control mechanisms differ from others in that they are based on heuristic information about the user making the request, the sensitivity level of the resources that may be affected by the request, and the organization's tolerance to the type of losses that may be inflicted by granting the requested service. For example, a service request from a remote installation would be treated differently if the remote installation requesting the access is known not to provide a specific security service (e.g. secure authentication or firewalls). A security risk assessment would have to be performed by the local host (or its security guard) taking into account such factors as the remote host's security safeguards, the type of operations/services being requested, and the sensitivity of the information that may be affected by the remote access operations to determine whether the remote request should be serviced. The security risk analysis can be applied to any component of the distributed system (e.g. a user, an end-system, a communication link, a LAN, etc.) and would allow the local host to determine the level of security/hostility of the component [5, 8].

Intrusion Detection:
As computer attacks become more and more sophisticated, the need to provide effective intrusion detection methods increases. Current best practices for protecting networks from malicious attacks are to deploy a security infrastructure that includes network intrusion detection systems. While those systems are useful for identifying malicious activity in a network, they generally suffer from several major drawbacks: inability to detect distributed or coordinated attacks, high false alarm rates, and producing large amount of data that is difficult to analyze. A major concern is the high rate of false alarms produced by current Intrusion Detection Systems which undermine the applicability of such systems. Effective protection of networks from malicious attacks remains a problem in both the research and network administering communities. Monitoring intrusion detection of multiple network systems requires the existence of multiple intrusion detection systems and a framework for integration.

Currently, a research project [8] is considering the potential benefits of distributed network intrusion detection systems by addressing two problems: first, how to

combine data from multiple intrusion detection sensors distributed in several subnets in a network (data fusion problem), and second, how to identify the most important data provided by multiple sensors in a network. The goal is to investigate a method to combine data from diverse distributed sub networks in order to improve false-alarm rates and timeliness in detecting attacks. As part of the project a series of analytic and simulation models are being developed to evaluate the potential benefits of distributed sensor based intrusion detecting systems for reducing false alarms and improving timeliness of detection for different fusion strategies.

The goal of the project is to accumulate the existing knowledge about the methods to monitor and detect attacks, and examine and analyze systems logs. Another related work emphasizes a revisit of database design to allow data fusion from multiple databases [8].

3 Intrusion Prevention Design Issues

An intrusion can be defined as an action aimed at compromising "Confidentiality, Integrity or Availability" of data. This includes unauthorized attempts to access data, manipulate data or make the system not viable [1].

Firewalls are probably one of the most important components of computer networks designed to protect against the "network elements" like intrusions, denial of service attacks, etc. [3]. Access Lists are the fundamental form of firewall protections; although an access list is not by itself a firewall. The access list compares both the source and destination IP address and port numbers of an incoming packet and decides if blocking or not blocking of some data traffic should take place. An access list can be placed on the dedicated router to block certain types of data packets from entering and exiting the network. An access list can also be placed in the "firewall software" residing on an individual computer thus protecting this machine only. There are situations in which a host with specific IP address is on the block list. Rules that govern the actions in a particular situation have to be defined [10], and Configuring an access list to deny packets from the remote host can solve the problem. An access list does not keep track of the data packet flow and therefore it can be matched with other software components to form a firewall. The main purpose of the firewall is to protect the inside of the network from the outside world while allowing traffic to go out. The firewall examines and matches stored source and destination IP addresses with the packet's addresses and port numbers. If the information matches the incoming packets, they are allowed to pass.

Basically, firewalls need to allow some outside traffic to enter the network. Well-configured firewalls are certainly an important part of any security strategy. However, if there were open ports to the Web, as it is in the case with commercial Web servers, firewalls would not stop an attack through those open ports. The three most common types of traffic allowed to the network are Web servers, DNS servers, and e-mail servers. Thus, the firewall must allow access to network using (for example) port 80 on Web server. If vulnerability is discovered on port 80, then the server's operating system could potentially get accessed. Hence, it makes sense to protect the Web server with its own firewall.

There are two types of intrusion prevention systems: Host-based intrusion prevention systems (HIPS) - software applications that protect individual servers; and network-based intrusion systems (NIPS) - placing sensors throughout the network to protect network segments [16, 18]. Traditional intrusion detection technologies are passive and merely notify users that an attack has occurred, whereas new technologies in host and network intrusion prevention are active and preclude malicious activities before damage can occur. Application firewalls are special cases of intrusion prevention systems; generally they are targeted to protect Web applications thus preventing attacks from the ports 80 and 443. Application firewalls normally work as a proxy examining every HTTP request and respond before sending them to the server. However, if HTTP and/or HTTPS connections to Web servers are the only means of communication to and from the Internet, then application firewalls could give very good protection against the attacks. Their great advantage is that they understand Web application logic because they work at the application level. This enables them to protect both Web applications and the system itself against attacks launched using HTTP/HTTPS connections [17].

There is a need for effective networks protection through a proactive stance to security, so that the newly discovered vulnerabilities are not exploited by malicious hackers.

Our research is related to proactive protection of machines that host Web servers, and ftp and telnet servers. We believe that the best way to protect those servers is by banning access to them from specific IP hosts that exhibit suspicious behaviors. Analyses of log files coming from firewalls and Web server access files allow us to update access list by adding suspicious IP addresses to the "banned IP addresses."

4 Log Files Significance

Many researchers have proposed and implemented different models which define different measures of system behavior, with an ad hoc presumption that normalcy and anomaly (or illegitimacy) will be accurately manifested in the chosen set of system features that are modelled and measured. Intrusion detection techniques can be categorized into ***misuse detection***, which uses patterns of well-known attacks or weak spots of the system to identify intrusions; and ***anomaly detection***, which tries to determine whether deviation from the established normal

usage patterns can be flagged as intrusions [9]. Misuse detection systems encode and match the sequence of "signature actions" (e.g., change the ownership of a file) of known intrusion scenarios.

The main shortcomings of such systems are: known intrusion patterns have to be hand-coded into the system; they are unable to detect any future (unknown) intrusions that have no matched patterns stored in the system. Anomaly detection systems establish normal usage patterns (profiles) using statistical measures on system features, for example, the CPU and I/O activities by a particular user or program. The main difficulties of these systems are: intuition and experience is relied upon in selecting the system features, which can vary greatly among different computing environments; some intrusions can only be detected by studying the sequential interrelation between events because each event alone may fit the profiles. It is very wise to remove all unnecessary features of the systems [13]. Configure system security settings as tight as possible, install the latest security patches and monitor logs on the entire system are essential in intrusion attempts [21].

With the use of data mining on large data files it is possible to produce detection models. Such models are produced off line, because an algorithm had to process huge amount of archived data [14]. These models can obviously be used for off-line intrusion detection. The idea of the system, that stops all attacks before they reach their target, is theoretically possible with preventive solutions rather than just detective ones. Off line behavior-based algorithms are the ideal tools to create intrusion prevention system, and log files contain the data that can useful in finding patterns of abnormal use of system's resources.

Logs are append-only, time stamped records representing some event that occurred in some computer or network device. Logs were used by programmers and system administrators to figure out "what's going on"' inside systems, and weren't of much value to take some important decisions related to denial of access to some business people. That's all changed with the rise of internet-based communication, and the need to archive traffic and to protect privacy. Unfortunately, tools to manage log data haven't kept up with the rise in traffic, and people have reverted to building custom tools. There are many reasons that traditional data management solutions cannot effectively manage log data, but the first one that users typically experience is in the sheer volume of log data [20], although in recent years several data mining techniques have been constructed to deal with such voluminous data.

5 Firewall's and Server's Logs Analysis

Individual log files that records activities related to a particular application although useful in many developments contain a lot of data that might not be particularly useful in intrusion prevention systems. However, comparative analysis of different types of log files, coming from different applications run on the same host can reveal useful interrelations that can be used in intrusion prevention systems. The process producing intersections of log files can be defined as a process dealing with the association, correlation, combining data coming from different sources, and has been known in the literature as data fusion. Llinas [15] defines it as a "process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates for observed entities, and to achieve complete and timely assessments of situations and threats, and their significance".

The key advantage of our approach is that it can automatically generate concise and accurate detection models from large amount of audit data. The methodology itself is general and mechanical, and therefore can be used to build intrusion prevention systems for a wide variety of computing environments.

In our project we have concentrated on intersections of log files coming from different applications run on the same host, as well as intersections of logs coming from different (or the same) applications run on different hosts during the same period of time. Intersections of log files are much smaller in size than individual logs and their analysis can be more complete [2]. We have used firewalls' log files coming from Web server machine and from regular desktop computer (in both cases coming from the same period of time), and Web server's access-log file from the same time period.

Project's Environment:
In our project we have collected data coming from two different machines. One machine runs Windows 2000 Server operating system with Neowatch firewall software, ftp server, and Java Web Server as an application. The other machine runs Windows XP Professional operating system with NeoWatch firewall software, ftp (FileZilla) server, regular Microsoft Office applications software and Web browser installed. The data available through the Web server consists of mostly course syllabi of courses thought in Computer Science department, corresponding PowerPoint presentation materials, example of exams, as well as on-line advising system.

As mentioned earlier we had three log files: 2 firewall's log files, coming from 2 separate machines, and one Web server access log file. Authorized user of the desktop machine was allowed to use (remotely) only ftp server. On the other machine (Web server) there were authorized ftp server users, and Web server administrator as remote users. All other users were authorized to use port 80 of Web server. Over the period of our experiment the largest log file was Web server access log file (about 120 thousands of entries). Most of the recorded entries were associated with our students accessing syllabi and other educational materials, as well as on-line advising system, and were coming from our (internal to the university) LAN network. Although there is a possibility that some

Table 1: Firewall's entries

Date	Time	Host IP	SPort	DPort
5/17/04	5:24:25	201.135.208.254	3289	445
5/17/04	5:24:25	201.135.208.254	1028	139

of our students could be "internal intruders", for our study (in order not to "cloud" the data) we have removed all entries coming from local IP addresses (Intranet).

Statistics:
During the initial preprocessing all log entries that were related to Intranet, were removed, leaving those entries that came from outside of our LAN.

After removing internal entries there were 17977 entries in Web server's firewall log, 17247 entries in Web server's access log, and 10517 entries in desktop's firewall log. Entries have been recorded during the period of 6 months. Web server's firewall has been set up in such a fashion that it did not recorded legitimate entries accessing "permitted" ports 80, and 21, so all Web server's firewall log's entries can be considered intrusive activities. Similarly desktop's firewall has not recorded legitimate entries through the port 21. Although the number of entries in Web server firewall log and Web server access log are similar (about 17 thousands) this similarity is accidental, as original number of entries (before removing Intranet's entries) in access log was about 10 times larger.

Also it is worth mention that although we have different goal in mind, a large number of entries in desktop's firewall log file proves that "honeypots" may play significant role in intrusion detection (and prevention), as our desktop could be considered a honeypot. Data taken from firewall and server logs has been processed by custom made program written in C. Files has been read and IP Address and the date/time from each log has been extracted, while the rest of information has been ignored. Even though each log file contains similar information, the format in which the information is recorded is different. For example, the Firewall Log file saves first the date (year/month/day), then the time and lastly the IP Address. The Server Log file saves first the IP Address, then the date (day/month/year) and finally the time. Even something as minute as how the date is written (in the Firewall Log file the month is recorded numerically, while on the Server Log file is recorded by name) could cause a problem on data abstraction from a file. Due to this, and in order to accurately acquire the data from both files, separate algorithms (within the same function) had been implemented to extract data from each file [19].

Analysis:
Entries in firewall's log have been recorded as in Table 1.

Entries in Web server log have been recorded with Host_IP, Date_Time, and Options (that specify method, file accessed, and protocol).

Comparison of Web server firewall log and Web server access log have produced intersections consisting of 48 different host IP addresses. As we can conclude from this intersection of logs, there were 48 hosts that have accessed Web server (on its standard port 80), but have also probed other ports on this server, and these probes have been recorded in the firewall's log.

Our initial assumption was that, intruder might probe port 80 first, and when they find it open (meaning that the machine hosting it is a server) intruder probes other ports trying to find vulnerabilities in the system. In many instances Web servers are not protected by individual firewall installed on the same host as Web server, so those machines are easier to "hijack" than others.

In 33 cases we found that the intruder in fact tried port 80 first and than probed other ports too. We could not, however, take these cases as an indicator of possible intrusion, as there were more than ten thousands cases of legitimate access to Web pages. We could however isolate IP addresses of intruders, as those were hosts having one successful access to Web server, and numerous (up to 50) probes of other ports. Below is the portion of the intersection of logs where the first line is access log entry and consecutive lines are entries from firewall log.

61.234.250.250	18/Sep/2004 : 23 : 27 : 59
61.234.250.250	2004/09/1823 : 28 : 03
61.234.250.250	2004/09/1823 : 28 : 03
61.234.250.250	2004/09/1823 : 27 : 59
61.234.250.250	2004/09/1823 : 28 : 00
61.234.250.250	2004/09/1823 : 28 : 01
61.234.250.250	2004/09/1823 : 28 : 06
61.234.250.250	2004/09/1823 : 28 : 06
61.234.250.250	2004/09/1823 : 28 : 04
61.234.250.250	2004/09/1823 : 28 : 06
61.234.250.250	2004/09/1823 : 28 : 03
61.234.250.250	2004/09/1823 : 28 : 05
61.234.250.250	2004/09/1823 : 28 : 03
61.234.250.250	2004/09/1823 : 28 : 02
61.234.250.250	2004/09/1823 : 28 : 04
61.234.250.250	2004/09/1823 : 28 : 06
61.234.250.250	2004/09/1823 : 28 : 06
61.234.250.250	2004/09/1823 : 28 : 05
61.234.250.250	2004/09/1823 : 28 : 06
.....	

From the above portion we can conclude that intruder has accessed Web server first (the earliest time) and than probed other ports. Such conclusion, however, could also be derived from analysis (data mining) of firewall log alone. We then analyzed intersection of log files coming from Web server installed on one machine, and log file coming from firewall installed on another machine (regular desktop). We have found that intersection of these log files consists of 106 different IP addresses. Detailed analysis has revealed that most of these hosts have accessed Web server several times and they probed only port 80 on desktop machines. When they realized that port 80

is not open those hosts never probed other ports. They were however some hosts that probed other ports on desktop machine. As before, we can say that those hosts were intruders trying to gain access to desktop's operating system. However, as before, this conclusion can be derived from analysis of desktop's firewall log file alone.

Comparison of both intersections (intersection_1: Web server access log and Web server firewall log; intersection_2: Web server access log and other desktop firewall log) has revealed that they contain some of the same IP addresses. They were IP addresses that belong to intruders trying to invade several different computers on our local network.

The existence of the hosts IP addresses in two different intersections of log files has prompted us to create and analyze the intersection of firewall log files coming from different machines on the same local area network. As we had only two firewall logs, we created intersection of these log files and we found 2356 different IP addresses of hosts that tried to access both machines during the period of 6 months. There were 1962 of those hosts that tried port 80 only once, but the rest (396) were (most probably) intruders probing different port numbers. Statistical analysis of these intersections of log files helps us to determine if patterns can be found, to recognize attempts of breaks into the server and so effectively increase security when such patterns are detected.

6 Conclusions

Off-line analysis of intersections of log files has allowed us to identify some host IP addresses that most probably belongs to intruders. As those intruders were able to reach our desktop and server that is behind our university firewall, we can provide a system administrator with those IP addresses and s/he can set the firewall in such a fashion that those IP will be banned from accessing our network. Intersection of firewall log files coming from different machines can be a source for IP addresses that belong to intruders. Having such information we can create a system that will identify those IP addresses in real-time, and will distribute that list to other machines on the LAN to be excluded from their firewall access lists, or it will deliver the list to network firewall.

Although the research for this project has been exploratory in nature, it is a thorough study. However, the current industry dynamics, frequent changes that occur in the computer field, as well as hacker ingenuity indicate a need for ongoing future research. Vigorous and energetic attention applied to current and future security concerns will help keep administrators steps away from vulnerability to attacks that so easily beset those companies who are not up to date in security measures.

The research for this project centered primarily on demonstrating one area of computer networks and security networking concepts e.g. Host Intrusion Prevention System (HIPS), in which we have search for new patterns of malicious behavior based on intersections of two files only. Future development should include analyses of data coming from intersections of more than two files at a time, and so allow the user to compare statistical data from more than one system and more than one application. Such improvement would give us the ability to determine hacking patterns across different systems, so they could be used in Network Intrusion Prevention Systems (NIPS) that provide first line of attacks prevention.

References

[1] A. Abraham, C. Grosan, and C. M. Vide, "Evolutionary design of intrusion detection programs," *International Journal of Network Security*, vol. 4, no. 3, pp 328-339

[2] *Analyst, Securities, Services and Solutions*, Apr. 2004. (http://netsecurity.about.com/gi/dynamic/offsite.htm?zi=1/XJ&sdn=netsecurity&zu=http%3A%2F%2F)

[3] J. S. Beasley, *Networking*, Pearson Education, Inc., Upper Saddle River, NJ, 2004.

[4] M. Beheshti and R. Wasniowski, "Information fusion in sensor-based intrusion detection systems," in *Secure IT Conference 2006*, Anaheim, CA, Mar. 2006.

[5] A. Berrached, M. Beheshti, A. d. Korvin, and R. Alo, "Intelligent access control in distributed systems using fuzzy relation equations," in *Proceedings of the International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet (SSGRR'00)*, Italy, July 2000.

[6] *Buyer's guide for intrusion prevention systems (IPS)*, June 3, 2004. (http://www.juniper.net/ solutions/literatur/buryer_guide/710005.pdf)

[7] CERT Coordination Center, *CERT/CC statistics 1988-2004*, Apr. 15, 2004. (http://www.cert.org/stats/cert_stats.html)

[8] A. deKorvin, A. Berrached, and M. Beheshti, "Active Access Control in Distributed Systems," in *The 2001 International Conference on Internet Computing (IC'01)*, 2001.

[9] Dunigan and Hinkel, "Intrusion detection and intrusion prevention on a large network, a case study," in *Proceedings of the 1st Workshop on Intrusion Detection and Network Monitoring*, Apr. 1999.

[10] J. Han, K. Kowalski, and M. Beheshti, "Detecting network intrusions based on a generalized rough set model," in *Proceedings of the International Symposium on Telecommunications (IST'05)*, pp. 247-252, Sep. 2005.

[11] R. Heady, G. Luger, A. Maccabe, and M. Servilla, *The Architecture of a Network Level Intrusion Detection System*, Technical report, Computer Science Department, University of New Mexico, Aug. 1990.

[12] M. Krause and F. H. Tipton, *Handbook of Information Security Management*, CRC Press LLC, Boca Raton, Florida, 1998.

[13] W. Lee and S. Stolfo, "Data Mining Approaches for Intrusion Detection," in *Proceedings of the Seventh USENIX Security Symposium (SECURITY'98)*, Jan. 1998.

[14] W. Lee, S. J. Stolfo, and P. K.Chan, "Real Time Data Mining-based Intrusion Detection," in *Proceedings Second DARPA Information Survivability Conference and Exposition*, 2001. (http://citeseer.ist.psu.edu/452795.html)

[15] J. Llinas, *Data Fusion Overview*, Jan. 6, 2006. (http://www.infofusion.buffalo.edu/tm/Dr.Llinas'stuff/DataFusionOverview.ppt)

[16] *Network intrusion prevention systems*, May 29, 2004. (http://www.networkintrusion.co.uk/inline.htm)

[17] Nitro Data Systems, *Intrusion Prevention: A White Paper*, July 2004. (http://www.bitpipe.com/ detail/RES/1090433720_603.html?src=TRM_TOPN)

[18] E. Ogren, *Host Intrusion Prevention is the Last Line of Defense for Networks Yankee Group*, 2004. (http://www.csoonline.com/analyst/report1265.html)

[19] G. Orvieto, *PROJECT SINSS: Statistical Informational Network Security System*, Senior Project-CSC495, CSUDH, 2004.

[20] A. Sah, "A new architecture for managing enterprise log data," in *Proceedings of LISA 2002: Sixteen Systems Administration Conference*, pp. 121-132, Berkeley, CA, 2002.

[21] J. Snyder, D. Newman, and R. Thayer, *How We Did it*, Network World, May 20, 2004. (http://www.nwfusion.com/reviews/2004/0216ipshow.html)

Kazimierz Kowalski received his BSc in Computer Engineering, and MSc and Ph.D. degrees in Computer Science from Wroclaw University of Technology (Poland). He is a professor at the California State University, Dominguez Hills where he teaches computer architecture and computer networks related courses. His research interests include information security in an on-line environment.

Mohsen Beheshti is a professor and Chair of Computer Science Department at California State University, Dominguez Hills. His research includes Computer Security,Data Modelling, Object-Oriented Database Systems, Fuzzy Logic, and Data Conversion; He received his Ph.D. from University of Louisiana at Lafayette in May of 1992 in the area of Concurrency Control in Object Oriented Database Management systems. He is a member of ACM, IEEE and Sigma Xi.