The Effective Utilization of Audit Logs in Information Security Management

Werner Olivier & Rossouw von Solms
Department of Information Technology

Port Elizabeth Technikon

Private Bag X6011

Port Elizabeth 6000

SOUTH AFRICA

Key words: Information Security Management, Audit Logs, Security Policies, Security Procedures, Security Monitoring, Security Agents, GPALM

Abstract: As more computers are connected to each other via enterprise wide networks and the Internet, information security and the implementation of security policies and procedures are becoming of paramount importance. All security information is logged in security audit logs. Currently, information security is being enforced via enterprise application suites based on platform specific security agents. These agents are installed at every workstation where security has to be enforced. These agents are however, only a small part of vast application suites and have many limitations. There is clearly room for improvement. The Generic Proactive Audit Log Model does away with agents to rather duplicate security audit logs on a dedicated log server. On this server extensive and intelligent audit log analysis can be performed to verify and enforce security policies to a much greater extent.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35575-7_19

J. H. P. Eloff et al. (eds.), *Information Security Management & Small Systems Security*

1. INTRODUCTION

As information technology expands in the business world, more and more people within an organisation utilise computers and related information systems. With the advent of the Internet, especially the World Wide Web (WWW), many companies have connected their computer based information systems to the Internet, in an attempt to gain a competitive advantage over their competitors. Connecting to the Internet results in a company potentially exposing their computer systems to the rest of the world. As a result of more people gaining access to the companies' information systems and the interconnection of many company networks via the Internet, computer related illegal activities are increasing continuously from within and outside the organisation. To counter this, the introduction of effective information security controls, within the company, is of imperative importance. Audit logs record all activities taking place in any computer system and have the potential to provide a clear history of all happenings, including security violations. Every activity of the user should be recorded in some form of an audit log. The challenge today is the effective utilisation of the audit logs in assisting the company in enforcing high-level security policies effectively, in order to maintain and enhance its information technology security in a proactive manner. Currently, the utilisation of audit logs, for security purposes, is integrated in enterprise product suites.

Complex enterprise application suites exist, attempting to fill the appetite of very large companies for integrated software solutions that integrate functions such as application management, business process management, Internet management, network management, desktop and server management, storage management, software distribution and security management (Computer Associates, 1998). Security management, in the context of these application suites, is the inspection of the audit logs to find any series of entries (activities), which are deemed a security violation by the company. Security auditing via audit logs is, however, only a minor component of these very large application suites. Most products in this category of application suites implement security auditing in a similar manner.

The objective of this paper is to study and analyse the utilisation of audit logs for security purposes, in these enterprise application products, and to suggest a specialised model for the effective utilisation of audit logs in information security management. This paper reports mainly on research in progress.

2. THE ENTERPRISE APPLICATION MODEL

These enterprise application suites attempt to solve as many computer related administration functions as possible in one product. The said application suites perform many functions, such as software distribution, custom security enhancement, custom log files, network maintenance, inventory-tracking, etc. These application suites attempt to be everything a company needs to maintain its computer environment, wrapped into one complete software solution. Some suite appear as a complete integrated software solution while others consist of modules fitting onto a basis application suite framework onto which one adds individual modules, as required by the company for their specific requirements. One common fact to all these application suites is that security auditing is only a minor component of any of these suites (Axent, 1998 and Computer Associates, 1998). The security modules within these application suites are not the primary function, and as such will be referred to as the secondary agent based security model.

3. SECONDARY AGENT BASED SECURITY MODEL

The security model found within these enterprise suites is all based on an agent. The agent is a software module that resides in memory on every relevant computer and scans all new lines added to the audit log files. The agents are also platform specific, thus different versions of agents exist for different computer platforms and operating systems. Typically, these agents can be remotely administrated from the information security officer's workstation. The agent scans for entries in the audit log files of the relative computer system that violate low level technical policies of the company, as implemented into the security model by the information security officer. An example of a low-level technical information security policy is; the company states that after three failed login attempts on a specific user name, the security officer must be informed. This will be implemented into the security model by creating a rule searching for failed login attempt entries in the audit logs, and finding three in a row on the same user name, will trigger a response being sent to the information security officer. The information security officer can then act accordingly, as it is the information security officer's duty to enforce an acceptable level of information security (Von Solms, 1993, p.19).

These security agent modules, typically work on the basis that for every security policy to be implemented, the information security officer has to create a new rule in the management section of the agent. For instance, using the mentioned example, the information security officer has already created a rule enforcing the "three failed login attempt" policy. If another policy must be enforced which states that for any failed login attempt as administrator or super user, the information security officer must immediately be notified; a complete new rule must be added. Thus, the "three failed login attempts" rule cannot be extended by merely adding a second parameter to it. This demonstrates that even for closely related technical security policies, separate rules have to be created (Axent, 1998 and Computer Associates, 1998).

4. LIMITATIONS OF THE AGENT BASED SECURITY MODEL

- The number of technical policies or procedures that are actively being monitored influences the performance of the computer system on which the agent resides. The greater the number of low level policies being checked, the greater the memory usage and as a result the performance of the computer can deteriorate dramatically. A worsening factor is that even for related policies, separate rules have to be implemented by the information security officer.

- The agent typically only search for lines in audit log files as specified by the information security officer with no added analysis of the audit logs being performed. This reflects a reactive approach of audit log utilization for security purposes. Little evidence of a proactive approach, where agent rules will start to interact with one another, creating more intelligent audit log analysis enabling the system to warn the information security officer of potential security violations. For example, if a few failed login attempts are recorded during the day and suddenly, after a period of time, unexpected access to important system files occur, which do not normally get accessed, it could be a hacker that managed to log on to the system and is busy modifying the system files in order to hide his/her presence in the system.

- These agents work autonomous from each other in the sense that agents for different operating systems and platforms do not communicate or even realize each other's existence. This is especially important in finding activities on various platforms, that when combined, indicate quite clearly that illegal activities are occurring within the system.

- Currently, even if you require just basic audit log security monitoring you have to purchase a major component of the entire enterprise suite at prohibitive costs. These products can typically only be afforded by very large companies, resulting in the situation that small to medium sized companies do not have the facilities or funds to implement audit log security monitoring.

- The audit log security software currently available do no or very little activity and trend analysis. Trend analysis, in the sense of analyzing the audit logs over longer periods of time to find hidden long-term trends showing possible security violations or violation attempts.

- The agent based security model mainly searches for specific entries in the audit logs, as specified by the information security officer, this approach addresses the implementation of low level technical policies as specified by the company. The effective enforcement of high-level security policies cannot ideally be implemented by merely studying specific entries in audit logs. A more integrated, analytical approach, spread over all the different platforms, is required to enforce high-level policies.

- When a potential security breach is identified, a logical backtrack analysis cannot easily be established. This is the process whereby the steps of the intruder are backtracked in an attempt to pinpoint the initial action where the intrusion first started. This can be a very useful action to determine where security should be enhanced within the company's computer systems.

It is clear that the secondary agent based security model does have significant limitations. The ideal situation would be to completely remove the processing overhead induced by the agent from the application servers and workstations which actually point to a scenario where the agent component will not be needed any more. Due to the large size of the audit logs and the complexity of searches and analysis to be performed on the audit log data, it would be beneficial to have a dedicated log server where all audit log processing could take place. Currently, most agent-based systems only consider a specific line in the audit logs and do not analyze text areas or fields within a line. There is no evidence of any form of trend analysis currently being used by agent based products. In conclusion, it can be said that much more intelligent analysis tools could and should be added to make the enforcement of security policies more effective. The process at the moment is purely reactive and the ideal is to become proactive to make information security implementation more effective and useful for the company. What is required, is a model that has a primary function of proactive information security management. This model will be referred to as the Primary Agent Based Security Model.

5. PRIMARY AGENT BASED SECURITY MODEL

The Generic Proactive Audit Log Model (GPALM) is a prototype of a primary agent based security model. This prototype has been developed at the Port Elizabeth Technikon and has the objective of utilizing audit log files in a proactive manner to effectively monitor all user activities for possible security breaches.

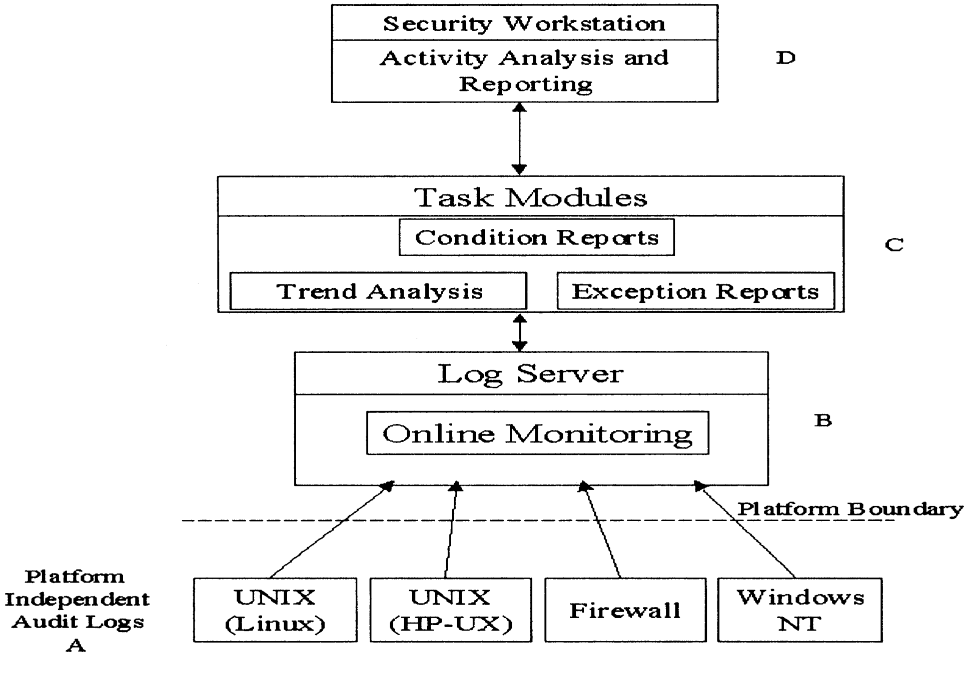

Figure 1. Figure 1. The GPALM Model (Krige, 1999, p.94)

GPALM deviates from the agent theme, as used in the secondary agent based security model, in order to lessen the burden of processing overheads on the application servers or workstations where the agent would normally operate in addition to normal processing performed by the computer. In order to eliminate this effect, GPALM proposes the usage of a dedicated log server, as seen in Figure 1. All platform independent audit logs created by computers in the company will be duplicated and integrated on the log server where all audit log analysis will be performed. The proactive audit log analysis on the log server is performed by a number of task modules from where the security workstation will be notified of any security violations. The following specific proactive functions are performed by the task modules (Krige, 1999, pp.96-98):

- exception reporting
- trends analysis
- condition reporting

These task modules form only one component of GPALM. GPALM actually consist of four components (Krige, 1999, pp.94-100):

1. The **platform independent audit logs,** that perform the actual data logging activity and duplicating all platform independent log data onto a dedicated log server. This involves configuring the different platforms to log remotely to the log server as well as locally. As previously stated, the reason for a dedicated log server is that analysis of log data on a live system (by agents) may introduce severe performance implications, thus all monitoring and analysis will be performed solely on the dedicated log server.
2. The actual **log server** is used as a central repository of the audit log data and an online monitoring and analysis tool. The online monitoring component that searches and analyses the duplicated audit log data for security exceptions and violations while in the process of enforcing the high level security policies of the organisation. In the event of a security violation, an exception report will be generated and sent to the appropriate person, for example, the information security officer.
3. The **task modules** enable the proactive usage of the audit logs. Activities of the task modules can be either real-time and/or non real-time. The task modules perform activities such as: condition reports, exception reports and trend analysis.

   Condition reports are created by querying the audit logs for a specific piece of information such as: on which dates a specific user logged into a computer. Exception reports enable the information security officer to view specific or all the security exceptions generated. Trend analysis identifies related security activities over a very long period of time, normally not noticed in the every day working environment. This method of analysis is very resource intensive and is as such not performed in real-time.
4. The **security workstation** is typically a computer with a graphical user interface (GUI), which allows easy interaction with the audit logs. The security workstation uses the appropriate task module for the query

needed. All messages created by the online monitoring component will appear on the screen of the security workstation.

GPALM strives towards an analysis friendly tool with a powerful and easy to use query language, resulting in the ability to perform regular and once-off analysis tasks on the security audit logs. What is envisaged is an 'intelligent' system with the ability to interpret activities and notify the security personnel of perceived possible security violations that are occurring or which may occur in future.

6. SECURITY ENHANCEMENT TO GPALM

The practical implementation of GPALM in its current form, combines audit logs from various UNIX platforms onto a central log server. Basic online monitoring, exception reporting and condition reporting are performed on the log server. The GPALM model can be further enhanced by adding more features, which will be discussed in more detail below.

6.1 Log file integration and cleansing.

Integrating audit log files from various operating systems pose challenges as well. Log data from different computer platforms differ significantly, since there do not seem to be any common, well defined standards for creating audit logs between the various operating systems. Audit logs tend be very large in size and cumbersome to work with, therefore another important factor is data cleansing. A large amount of unimportant data written to the audit logs are of no value whatsoever, therefore any data being written to the log server must first be cleansed to ensure that only useful data resides on the log server. This task is performed by audit reduction tools, which remove audit records with no security relevance (NIST, 1995, p.219). In the GPALM context, the audit reduction tool must firstly, be able to distinguish between various platform independent log entries and secondly, whether it can be discarded or not.

It is of great importance to determine whether all the data needed for high-level policy implementation is actually available in the audit logs. This leads to another area of research in that if the data needed does not exist, what steps will be taken to create the custom data logging features.

6.2 Advanced Reporting

All the enterprise models do have reporting features, but all at a basic level. 'Intelligence' is, therefore, needed to determine what action must be performed, who should be notified and the method of notification. As the severity of the security violation increases, the appropriate method of notification of the relevant personnel should be used. For instance, if the information security officer is not in his/her office and the severity of the security violation is high, a SMS message will be sent directly to his/her cellular phone. The system should have the 'intelligence' to find the security officer if he/she is logged in anywhere on the network, else move to the next level of notification.

6.3 High Level Security Policies

Currently, using any software based on the enterprise model, only low-level policies can be implemented within a company. Tools are needed to implement high-level security policies, with special emphasis on the relation between high-level policies and procedures, and what mechanisms are available to implement security policies in the real world.

6.4 Hacking Templates.

In most hacking attempts, there exist certain series of characteristics that are common to any hacker attempting to break into a system. Although there are many forms of hacking attempts to illegally enter into a system, for most types of hacking attempts, certain tell tale characteristics or signs will appear in the security audit logs. Obviously, the common characteristics will hold true for separate operating systems since different operating systems have different weaknesses that are exploited by hackers. These characteristics can be converted to a trend consisting of a number of steps that a hacker will follow to gain initial access to a system. Based on this knowledge, it is possible to analyze a security audit log and search for a series of events that approximate the theoretical trend in order to realize in advance that a hacking attempt is in progress. If these steps of a hacking attempt are known, a template type of script can be created to provide a warning in advance, that a possible breach of security is imminent. The script will search in the audit logs for the steps, as laid out by the template created, and the more steps found in the audit logs, the better the correlation to the template and the better the chance that an actual hacking attempt is in progress.

6.5 Advantages of GPALM after enhancements

GPALM solves the problem of agents taking away resources from the application servers and workstations that it reside on, by moving all processing to a dedicated log server. The only price to be paid using this method is the network traffic increase to duplicate the audit logs of all the application servers to a log server. GPALM will have highly advanced analysis tools capable of trend analysis etc., including an easy to use query language. GPALM will have some form of 'intelligence' built in to report more than just the obvious line errors as per the secondary agent based security model. Because of the integrated audit logs on the log server, GPALM will be capable to find trends and security violation events that span across more than one physical server or workstation. One error in a single audit log on its own may be considered trivial, but the same error found in audit logs of more than one server or workstation will point more definitely to a possible security violation. GPALM will also have advanced reporting procedures and features that will intelligently inform the relevant parties based on the severity of the security violation.

7. CONCLUSION

Current secondary agent based security models mostly act in a reactive way to implement low-level technical policies. GPALM is potentially capable of verifying and enforcing high-level policies through 'intelligent' interpretation of integrated audit log data on a single dedicated log server. Ad hoc queries can easily be performed by an easy to use querying facility. Furthermore, it is also capable of actually generating new data for policies not yet implemented in the company or policies perceived not needed by the company. Backtracking analysis will result in identifying the steps that were used to perform some illegal action. These steps are actually the relevant information needed to formulate a hacking template for future implementation. Taking this a step further, the ultimate analysis tool will be capable of generating new data through backtrack analysis and using that data to automatically create a new hacking template or new policy without any user intervention. This form of analysis is moving into the realm of metadata. This is a field where a huge amount of research still needs to be done. Software capable of performing these functions will be able to identify unknown security violations, create new security rules and react to them without any human intervention.

8. BIBLIOGRAPHY

Axent. (1998). Intruder Alert. [online]. Available from Internet: URL http://www.axent.com/product/smsbu/ITA/default.htm

Computer Associates. (1998). Unicenter TNG: Total Enterprise Management. US0498EN U.S.A.

Krige, W. (1999). The usage of audit logs for effective information security management. Unpublished master's thesis. Port Elizabeth Technikon, Port Elizabeth, South Africa.

US. Department of Commerce. (1995). An introduction to computer security: The NIST handbook. Washington: U.S. Government Printing Office.

Von Solms, R. (1993). A process approach to information security management. IFIP'93, WG 11.1, Toronto, Canada, 1993.