Chapter 34

REAL-TIME MONITORING AND AUDITING

INTRODUCTION

The database community has begun to realize its role in a robust information security infrastructure. Traditionally there has been little concern for the security of relational database systems, and for the vast amounts of data they house. In the early days of relational databases, gaining access to a database was so difficult that the need for complex security features was irrelevant. Databases were housed in mainframes not accessible directly from the network. Slowly they were ported to other networked systems such as UNIX, Linux, and Windows, but even when this happened, the databases were kept far behind the firewall out of the reach of the typical Black Hat surfing the Internet [1].

As digital information has become more and more critical to how businesses and customers interact, firewalls, which once served as the walls of the fortress, have now become an archaic means of defense providing little real-world protection. Even the smallest of organizations have numerous entry points into their networks – wireless access, remote employees connecting through VPN, customers connecting through web applications, etc. As well, employees take home laptops, hook up to a home cable modem connection, get infected with a Trojan, and then return to work. This becomes a virtual "store and forward" attack that effectively gives the Black Hat a backdoor into an organization's internal network [1].

As the threats have changed, the responses must change as well. Rather than focusing protection solely on perimeter security, it is imperative you start looking at protecting data at the source – inside the database. The goal is to provide protection where it is most effective and cannot be circumvented. The database has come to be the de facto standard for housing an organization's most valuable information. The logical extension of these facts is that the database is where your strongest security needs to be in place [1].

Given the flurry of new information security related legislature that has arisen, there is a strong call to take extra steps to provide strong security at the database level. The need to protect sensitive information no longer needs to be driven by corporate strategy, federal

laws are forcing organizations to put the proper defenses in place, with consequences for those who do not comply. Information security is focused on preserving the confidentiality, integrity, and availability (CIA) of the information system. Without maintaining these basics tenets, a database does not measure up to the requirements of handling commercial data [1].

Without real-time auditing and monitoring of data, CIA is impossible to maintain. While there have been many discussions on the need to provide some level of auditing and monitoring, there is little information to help organizations define what the appropriate auditing and monitoring strategy is for them. This chapter very briefly focuses on "theoretical best-practices" combined with "real-world practicality" to define a usable policy for the real-time auditing and monitoring of databases. By following the policies outlined in this chapter, you can properly implement a database system that will work well, and provide adequate security for the data it houses [1].

THE IDEAL MONITORING SOLUTION

Monitoring can be a complex task. Collecting data is the simple part of the job. An ideal solution needs to handle the tricky aspects of the job, as well as the simple ones. Some of the more difficult issues that need to be addressed by a monitoring solution include:

1. Deciding what to monitor for
2. Handling the volume of data that needs to be monitored
3. Detecting when something malicious has occurred
4. Ensuring the integrity of the audit data [1]

WATCHING THE DATABASE ADMINISTRATOR

Modern database technology is designed to be managed by database administrators, who are the unrestricted owners of the database. This architecture leaves an organization's most critical information entirely exposed to and controlled by a small handful of technologists, the Database Administrators or DBAs. This leaves both the DBA and the entire organization in a precarious position. The DBA is afraid that she or he will be blamed for any information leak, while the organization is forced to almost blindly trust a small number of people in the technology group [1].

One way to mitigate this risk is by properly auditing and monitoring the activities of the DBA. The amount of work that a DBA does on a production server is very limited. Auditing and monitoring this activity does not add significant overhead to the system [1].

How do you properly audit activity in a database? Native auditing fails here because it is fully under the control of the DBA. He or she can easily turn off auditing, clear the audit logs, manipulate an audit record, or even reconfigure auditing to filter out their own malicious activity. Auditing should ultimately provide a separation of duties. An ideal audit system would be intelligent enough to distinguish database administrative accounts, filter out noise and irrelevant events, and succinctly illustrate their activities. Auditing data should be