Transaction-Relationship Oriented Log Division for Data Recovery from Information Attacks

Satyadeep Patnaik, Subex Technology, Inc., USA
Brajendra Panda, University of Arkansas, USA

ABSTRACT

The survivability of database systems in case of information attacks depends exclusively on the logging mechanism. The recovery methods specifically designed for recovery from information attacks require that the log must record all operations of every transaction and that the log should never be purged, thus incurring enormous growth of the log. In this paper, we have developed a model for accurate and fast data recovery from information attacks in order to reduce denial-of-service while providing consistent values of data items. Our model divides the log based on transaction relationships and stores each segment as a separate file, which can then be accessed independently as required. We have proved that only one of these segments will be accessed during damage assessment and recovery process. Appropriate damage assessment and recovery methods are also presented. Through simulation we have validated that our model significantly reduces recovery time.

Keywords: transaction dependency, log segmentation, information attacks, data recovery

INTRODUCTION

The importance of information sharing for a successful existence of an organization is paramount. Recent boom in Internet technologies has made this process even faster and easier. When we use computer systems to share information with others via network, we leave the door open for hackers to invade our systems. It is extremely difficult to make computer systems absolutely free of security flaws. Smart hackers often find a way to break into systems and perform unauthorized activities. Password sniffing and session hijacking, for example, are among various means of intruding into a system. In such cases, the system under attack will not be able to detect an attacker from a legitimate user. Therefore, any such malicious transaction will be executed and committed just as any other legitimate transaction.
Accepting the fact that these attacks are possible, the system should be designed to survive the hostile activities and be able to provide uninterrupted service. As noted in a recent article (Panda and Giordano, 1999), protection of the system from unauthorized users, detection of any hostile activities, and recovery from any damage caused by intruders are the three major activities in building secure systems. Extensive research has been carried out to protect computer systems from unauthorized users. A few of the references are Bell and LaPadula (1973), Grohn (1976), Denning (1982), Denning and Lunt (1987), and Lunt, Denning, Schell, Heckman, and Shockley (1990). The next step in the process of building secure systems is intrusion detection. There has been a lot of research performed in the area of intrusion detection. Some of the notable products available today are the Automated Security Incident Measurement (ASIM), the Network Intrusion Detector (NID), the Joint Intrusion Detection System (JIDS), Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD) developed by SRI International (Porras and Valdes, 1998), and Distributed Intrusion Detection System (DIDS) (Mukherjee, Herelein, and Letitt, 1994). LaPadula (1999) provides an excellent account on state-of-the-art anomaly detection tools. Since system protection and intrusion detection issues are outside the scope of this research, we would not discuss them here.

Graubart, Schlipper, and McCollum (1996) pointed out that detection of a hostile transaction is not always immediate. In several cases, as stated earlier, the transaction executes as a valid one and commits after modifying some sensitive data in the database. As per the ACID properties of transactions, these updates on data items are made permanent and the values of these affected data items are made available to other transactions. Thus, as stated by Amman, Jajodia, McCollum and Blaustein (1997), the damage can spread to other parts of the database through legitimate users as they update fresh data after reading any damaged data. So, timely and faster recovery is required to stop further spreading of damage. Since an attack may get detected a long time after its occurrence and the log is the only source for recovery, the log must never be purged. Moreover, to identify affected transactions using transaction dependency relationships, all operations of every transaction need to be stored in the log. Therefore, the size of the log becomes massive, incurring a prolonged log access time during damage assessment and recovery process (Lala and Panda, 2001).

In this paper, we have developed a model to divide the log into several segments based on transaction dependency relationships. We proved that only one of these segments would be accessed during damage assessment and recovery process. We also have presented a technique to divide large segments resulting from merger of multiple segments, into mini-segments. Necessary damage assessment and recovery algorithms have been provided which access transaction operations in an affected segment in case of attack detection and then determine affected transactions and subsequently recover the database. A simulation model was used to test the performance of our method. The result shows dramatic improvement over the time required to perform damage assessment and recovery using the traditional approach.

The rest of the paper is organized as follows. In the following section, we describe the motivation for this research. The segmentation algorithm and its proof of correctness are discussed next, followed by an algorithm for the post-segmentation
Adaptation of an Agile Information System Development Method
www.igi-global.com/chapter/adaptation-agile-information-system-development/28433?camid=4v1a

Multilevel Databases
www.igi-global.com/chapter/multilevel-databases/11177?camid=4v1a