A REAL-TIME INTRUSION PREVENTION SYSTEM FOR COMMERCIAL ENTERPRISE DATABASES

Ulf T. Mattsson
Protegrity, 201 Shannon Oaks Cir, Suite 205, Cary, NC 27511

Keywords: Isolation, Intrusion Tolerance, Database Security, Encryption, GLBA, HIPAA

Abstract: Modern intrusion detection systems are comprised of three basically different ap-proaches, host based, network based, and a third relatively recent addition called pro-cedural based detection. The first two have been extremely popular in the commercial market for a number of years now because they are relatively simple to use, understand and maintain. However, they fall prey to a number of shortcomings such as scaling with increased traffic requirements, use of complex and false positive prone signature databases, and their inability to detect novel intrusive attempts. This intrusion detection system interacts with the access control system to deny further access when detection occurs and represent a practical implementation addressing these and other concerns. This paper presents an overview of our work in creating a practical database intrusion detection system. Based on many years of Database Security Research, the proposed solution detects a wide range of specific and general forms of misuse, provides detailed reports, and has a low false-alarm rate. Traditional commercial implementations of database security mechanisms are very limited in defending successful data attacks. Authorized but malicious transactions can make a database useless by impairing its integrity and availability. The proposed solution offers the ability to detect misuse and subversion through the direct monitoring of database operations inside the database host, providing an important complement to host-based and network-based surveil-lance. Suites of the proposed solution may be deployed throughout a network, and their alarms man-aged, correlated, and acted on by remote or local subscribing security ser-vices, thus helping to address issues of decentralized management.

1 INTRODUCTION

Most companies solely implement perimeter-based security solutions, even though the greatest threats are from internal sources. Additionally, companies implement network-based security solutions that are designed to protect network resources, despite the fact that the information is more often the target of the attack. Recent development in information-based security solutions addresses a defense-in-depth strategy and is independent of the platform or the database that it protects. As organizations continue to move towards digital commerce and electronic supply chain management, the value of their electronic information has increased correspondingly and the potential threats, which could compromise it, have multiplied. With the advent of networking, enterprise-critical applications, multi-tiered architectures and web access, approaches to security have become far more sophisticated. A span of research from authorization (P. P. Griffiths et al., 1976), (F. Rabitti et al., 1994), (S. Jajodia et al., 1997), to inference control (M. R. Adam, 1989), to multilevel secure databases (M. Winslett et al., 1994), (R. Sandhu et al., 1998), and to multi-level secure transaction processing (V. Atluri et al., 1999), addresses primarily how to protect the security of a database, especially its confidentiality. However, limited solutions has been presented on how to practically implement a solution to survive successful database attacks, which can seriously impair the integrity and availability of a database. Experience with data-intensive applications such as credit card billing, has shown that a variety of attacks do succeed to fool traditional database protection mechanisms. One critical step towards attack resistant database systems is intrusion detection, which has attracted many researchers (D.E.Denning, 1987), (T. Lunt et al., 1992), (R. Jagannathan et al., 1993), (P. Helman et al., 1993), (T.F. Lunt, 1993), (B. Mukherjee et al., 1994), (Teresa Lunt et al., 1998), (T. Lane et al., 1998), (Wenke Lee et al., 1999). Intrusion detection systems monitor system or network activity to

T. Mattsson U. (2004).
A REAL-TIME INTRUSION PREVENTION SYSTEM FOR COMMERCIAL ENTERPRISE DATABASES.
In *Proceedings of the First International Conference on E-Business and Telecommunication Networks*, pages 275-280
DOI: 10.5220/0001381102750280

discover attempts to disrupt or gain illicit access to systems. The methodology of intrusion detection can be roughly classed as being either based on statistical profiles (H. S. Javitz et al., 1991), (H. S. Javitz et al., 1994), (D. Samfat et al., 1997) or on known patterns of attacks, called signatures (K. Ilgun, 1993), (T.D. Garvey et al., 1991), (P.A. Porras et al., 1992), (K. Ilgun et al., 1995), (S.-P. Shieh et al., 1997). Intrusion detection can supplement protection of network and information systems by rejecting the future access of detected attackers and by providing useful hints on how to strengthen the defense. However, intrusion detection has several inherent limitations: Intrusion detection makes the system attack-aware but not attack-resistant, that is, intrusion detection itself cannot maintain the integrity and availability of the database in face of attacks. Achieving accurate detection is usually difficult or expensive. The false alarm rate is high in many cases. The average detection latency in many cases is too long to effectively confine the damage. To overcome the limitations of intrusion detection, a broader perspective is introduced, saying that in addition to detecting attacks, countermeasures to these successful attacks should be planned and deployed in advance. In the literature, this is referred to as survivability or intrusion tolerance. In this paper, we will address a useful technique for database intrusion prevention, and present the design of a practical system, which can do attack prevention.

2 PROBLEM FORMULATION

In order to protect information stored in a database, it is known to store sensitive data encrypted in the database. To access such encrypted data you have to decrypt it, which could only be done by knowing the encryption algorithm and the specific decryption key being used. The access to the decryption keys could be limited to certain users of the database system, and further, different users could be given different access rights. Specifically, it is preferred to use a so-called granular security solution for the encryption of databases, instead of building walls around servers or hard drives. In such a solution, which is described in this paper, a protective layer of encryption is provided around specific sensitive data-items or objects. This prevents outside attacks as well as infiltration from within the server itself. This also allows the security administrator to define which data stored in databases are sensitive and thereby focusing the protection only on the sensitive data, which in turn minimizes the delays or burdens on the system that may occur from other bulk encryption methods. Most preferably the encryption is made on such a basic level as in the column level of the databases. Encryption of whole files, tables or databases is not so granular, and does thus encrypt even non-sensitive data. It is further possible to assign different encryption keys of the same algorithm to different data columns. With multiple keys in place, intruders are prevented from gaining full access to any database since a different key could protect each column of encrypted data.

2.1 New Requirements

The complexity of this task was dramatically increased by the introduction of multi-platform integrated software solutions, the proliferation of remote access methods and the development of applications to support an increasing number of business processes. In the "good old days", files and databases contained fewer types of information (e.g., payroll or accounting data) stored in centralized locations, which could only be accessed, by a limited number of individuals using a handful of controlled access methods. As more types of information were migrated to electronic formats (and ever more databases proliferated, often with little planning), there was a simultaneous increase in the number of users, access methods, data flows among components and the complexity of the underlying technology infrastructure. Add to this the demand from users forever more sophisticated uses of information (data mining, CRM, etc.), which are still evolving, and the management's enhanced awareness of the value of its information. Database intrusion tolerance can mainly be enforced at two possible levels: database level and transaction level. Although transaction level methods cannot handle database level attacks, it is shown that in many applications where attacks are enforced mainly through malicious transactions transaction level methods can tolerate intrusions in a much more effective and efficient way. Database level intrusion tolerance techniques can be directly integrated into an intrusion tolerance framework with the ability to back out from a malicious database transaction. Two levels of intrusion response behavior may be deployed; an intrusion into the database system as such, or an intrusion to the actual data. In the first case focus is on preventing from further malicious activities, i e you have had an attack but it is handled by next layer of security. In the second the behavior is a rollback of the data written, to handle the attack afterwards.

3 PROBLEM SOLUTION

In the above-mentioned solutions the security administrator is responsible for setting the user permissions. Thus, for a commercial database, the security administrator operates through a middle-ware application, the access control system (ACS), which provides authentication, encryption and decryption services. The ACS is tightly coupled to the database management system (DBMS) of the database. The ACS controls access in real-time to the protected elements of the database. Such a security solution provides separation of the duties of a security administrator from a database administrator (DBA). The DBA's role could for example be to perform usual DBA tasks, such as extending tablespaces etc, without being able to see (decrypt) sensitive data. The SA could then administer privileges and permissions, for instance add or delete users. For most commercial databases, the database administrator has privileges to access the database and perform most functions, such as changing password of the database users, independent of the settings by the system administrator. An administrator with root privileges could also have full access to the database. This is an opening for an attack where the DBA can steal all the protected data without any knowledge of the protection system above. The attack is in this case based on that the DBA impersonates another user by manipulating that users password, even though a hash algorithm enciphers the user's password. An attack could proceed as follows. First the DBA logs in as himself, and then the DBA reads the hash value of the users password and stores this separately. Preferably the DBA also copies all other relevant user data. By these actions the DBA has created a snapshot of the user before any altering. Then the DBA executes the command "ALTER USER username IDENTIFIED BY newpassword". The next step is to log in under the user name "username" with the password "newpassword" in a new session. The DBA then resets the user's password and other relevant user data with the previously stored hash value. Thus, it is important to further separate the DBA's and the SA's privileges. The DBA attack prevention described here is specific to databases with internal authentication. Databases that utilizes external (OS level) authentication provides a level of separation of duties, and the database encryption system, or intrusion prevention system, can verify that the database session is properly authenticated by the external authentication system before any decryption of sensitive data is allowed.

3.1 A New Approach

Within the framework, the Intrusion Detector identifies malicious transactions based on the history kept (mainly) in the log. The Intrusion Assessor locates the damage caused by the detected transactions.

3.2 Intrusion Prevention Solution

The method allows for a real time prevention of intrusion by letting the intrusion detection process interact directly with the access control system, and change the user authority dynamically as a result of the detected intrusion. The hybrid solution combines benefits from database encryption toolkits and secure key management systems. The hybrid solution also provides a single point of control for database intrusion prevention, audit, privacy policy management, and secure and automated encryption key management (FIPS 140 Level 3). The Database Intrusion Prevention is based on 'context checking' against a protection policy for each critical database column, and prevents internal attacks also from root, DBA, or 'buffer overflow attacks', by automatically stopping database operations that are not conforming to the Database Intrusion Prevention Policy rules. The Database Intrusion Prevention and alarm system enforces policy rules that will keep any malicious application code in a sand box regarding database access. The policy enforcement system, integrated with an external network authentication system, perform the following basic checking: Session Authentication and Session Encryption, Software Integrity, Data Integrity, and Meta Data Integrity, Time of Access, and related policy rules. In database security, it is a well-known problem to avoid attacks from persons who have access to a valid user-ID and password. Such persons cannot be denied access by the normal access control system, as they are in fact entitled to access to a certain extent. Such persons can be tempted to access improper amounts of data, by-passing the security. Such persons can be monitored and controlled by this database intrusion prevention system and automatically be locked out from database operations that are not conforming to the Database Intrusion Prevention Policy rules. Other solutions in this problem area have been suggested:

Network-Based Detection - Network intrusion monitors are attached to a packet-filtering router or packet sniffer to detect suspicious behavior on a network as they occur. *Server-Based Detection* - These tools analyze log, configuration and data files from individual servers as attacks occur, typically by

placing some type of agent on the server and having the agent report to a central console. *Security Query and Reporting Tools* - These tools query NOS logs and other related logs for security events or they glean logs for security trend data. Accordingly, they do not operate in real-time and rely on users asking the right questions of the right systems.

3.3 Inference Detection

A variation of conventional intrusion detection is detection of specific patterns of information access, deemed to signify that an intrusion is taking place, even though the user is authorized to access the information. A method for such inference detection, i.e. a pattern oriented intrusion detection, is disclosed in US patent 5278901 to Shieh et al.
None of these solutions are however entirely satisfactory. The primary drawback is that they all concentrate on already effected queries, providing at best information that an attack has occurred.

3.4 Intrusion Prevention Profile

By defining at least one intrusion detection profile, each comprising at least one item (column access) access rate, associating each user with one of the profiles, receiving a query from a user, comparing a result of the query with the item access rates defined in the profile associated with the user, determining whether the query result exceeds the item access rates, and in that case notifying the access control system to alter the user authorization, thereby making the received request an unauthorized request, before the result is transmitted to the user. According to this method, the result of a query is evaluated before it is transmitted to the user. This allows for a real time prevention of intrusion, where the attack is stopped even before it is completed. This is possible by letting the intrusion detection process interact directly with the access control system, and change the user authority dynamically as a result of the detected intrusion. The item access rates can be defined based the number of rows a user may access from an item, e.g. a column in a database table, at one time, or over a certain period of time. In a preferred implementation, the method further comprises accumulating results from performed queries in a record, and determining whether the accumulated results exceed any one of the item access rates. The effect is that on one hand, a single query exceeding the allowed limit can be prevented, but so can a number of smaller queries, each one on its on being allowed, but when accumulated not being allowed. It should be noted that the accepted item access rates not necessarily are restricted to only one user. On the contrary, it is possible to associate an item access rate to a group of users, such as users belonging to the same access role (which defines the user's level of security), or connected to the same server. The selective activation of the intrusion detection will then save time and processor power. According to another implementation of the method, the intrusion detection policy further includes at least one inference pattern, and results from performed queries are accumulated in a record, which is compared to the inference pattern, in order to determine whether a combination of accesses in the record match the inference policy, and in that case the access control system is notified to alter the user authorization, thereby making the received request an unauthorized request, before the result is transmitted to the user. This implementation provides a second type of intrusion detection, based on inference patterns, again resulting in a real time prevention of intrusion.

4 RELATED WORK

There is a variety of related research efforts that explore what one can do with audit data to automatically detect threats to the host. An important work is MIDAS (M. M. Sebring et al., 1998), as it was one of the original applications of expert systems—in fact using P-BEST—to the problem of monitoring user activity logs for misuse and anomalous user activity. CMDS, by SAIC, demonstrated another application of a forward-chaining expert-system, CLIPS, to a variety of operating system logs (P. Proctor, 1994). USTAT (K. Ilgun, 1993) offered another formulation of intrusion heuristics using state transition diagrams (P. A. Porras et al., 1992), but by design remained a classic forward-chaining expert sys-tem inference engine. ASAX (J. Habra et al., 1992) introduced the Rule-based Sequence Evaluation Language (RUSSEL) (A. Mounji, 1997), which is tuned specifically for the analysis of host audit trails. Recent literature form the RAID conferences, as well as IEEE Security and Privacy, the DARPA program on survivability that concentrated on detecting and surviving attacks, and a large scale DARPA project called DemVal, are dealing with the survivability of a database. The idea of attack prevention, that will not allow access after a threshold is reached, is also discussed in the SRI Appache IDs system. The approach is sometimes also called application level intrusion detection, rather than procedural intrusion detection.

5 CONCLUSION

Our technology and approach fills that gap by providing practical application based intrusion detection and response. We suggest that this gives The Hybrid the unique ability to detect and halt completely novel attacks that have yet to be seen on the Internet, and better yet, we have the ability to protect the first person to see a new attack or exploit. Removing all software vulnerabilities is clearly an unsolvable problem. Providing restrictive and onerous barriers to software use makes the software uncomfortable and difficult to use. Monitoring and controlling program execution at run time through behavioral control is the missing piece in the security puzzle. The complete puzzle has three pieces; data control (encryption), access control, and behavioral control.

REFERENCES

M. R. Adam. *Security-Control Methods for Statistical Database: A Comparative Study.* ACM Computing Surveys, 21(4), 1989.

P. Ammann, S. Jajodia, and P. Liu. *Recovery from malicious trans-actions.*

IEEE Transactions on Knowledge and Data Engineering, 2001. To appear.

V. Atluri, S. Jajodia, and B. George. *Multilevel Secure Transaction Processing.* Kluwer Academic Publishers, 1999.

D. Barbara, R. Goel, and S. Jajodia. *Using checksums to detect data corruption.* In Proceedings of the 2000 International Conference on Extending Data Base Technology, Mar 2000.

P. A. Bernstein, V. Hadzilacos, and N. Goodman. *Concurrency Control and Recovery in Database Systems.* Addison-Wesley, Reading, MA, 1987.

S. B. Davidson. *Optimism and consistency in partitioned distributed database systems*. ACM Transactions on Database Systems, 9(3):456–581, September 1984.

D.E.Denning. *An intrusion-detection model.* IEEE Trans. on Software Engineering, SE-13:222–232, February 1987.

T.D. Garvey and T.F. Lunt. *Model-based intrusion detection*. In Proceedings of the 14th National Computer Security Conference, Balti-more, MD, October 1991.

P. P. Griffiths and B. W. Wade. *An Authorization Mechanism for a Relational Database* System. ACM Transactions on Database Systems, 1(3):242–255, September 1976.

P. Helman and G. Liepins. *Statistical foundations of audit trail analysis for the detection of computer misuse.* IEEE Transactions on Software Engineering, 19(9):886–901, 1993.

K. Ilgun. Ustat: *A real-time intrusion detection system for unix.* In Proceedings of the IEEE Symposium on Security and Privacy,Oak-land, CA, May 1993.

K. Ilgun, R.A. Kemmerer, and P.A. Porras. *State transition analysis*: A rule-based intrusion detection approach. IEEE Transactions on Software Engineering, 21(3):181–199, 1995.

R. Jagannathan and T. Lunt. *System design document: Next generation intrusion detection expert system* (nides). Technical report, SRI International, Menlo Park, California, 1993.

S. Jajodia, P. Samarati, V. S. Subrahmanian, and E. Bertino. *A unified framework for enforcing* multiple access control policies. In Proceedings of ACM SIGMOD International Conference on Management of Data, pages 474–485, May 1997.

H. S. Javitz and A. Valdes. The sri ides statistical anomaly detector. In Proceedings IEEE Computer Society Symposium on Security and Privacy, Oakland, CA, May 1991.

H. S. Javitz and A. Valdes. The nides statistical component description and justification. Technical Report A010, SRI International, March 1994.

T. Lane and C.E. Brodley. Temporal sequence learning and data reduction for anomaly detection. In Proc. 5th ACM Conference on Computer and Communications Security, San Francisco, CA, Nov 1998.

Wenke Lee, Sal Stolfo, and Kui Mok. A data mining framework for building intrusion detection models. In Proc. 1999 IEEE Symposium on Security and Privacy, Oakland, CA, May 1999.

P. Liu, S. Jajodia, and C.D. McCollum. Intrusion confinement by isolation in information systems. Journal of Computer Security, 8(4):243–279, 2000.

P. Luenam and P. Liu. Odam: An on-the-fly damage assessment and repair system for commercial database applications. In Proc. 15th IFIP WFG11.3 Working Conference on Database and Application Security, Ontario, Canada, July 2001.

T. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, C. Jalali, H. S. Javitz, A. Valdes, P. G. Neumann, and T. D. Garvey. A real time intrusion detection expert system (ides). Technical report, SRI International, Menlo Park, California, 1992.

Teresa Lunt and Catherine McCollum. Intrusion detection and response research at DARPA. Technical report, The MITRE Corporation, McLean, VA, 1998.

T.F. Lunt. A Survey of Intrusion Detection Techniques. Computers & Security, 12(4):405–418, June 1993.

J. McDermott and D. Goldschlag. Storage jamming. In D.L. Spooner, S.A. Demurjian, and J.E. Dobson, editors, Database Se-curity IX: Status and Prospects, pages 365–381. Chapman & Hall, London, 1996.

J. McDermott and D. Goldschlag. *Towards a model of storage jamming.* In Proceedings of the IEEE Computer Security Foundations

Workshop, pages 176–185, Kenmare, Ireland, June 1996.

B. Mukherjee, L. T. Heberlein, and K.N. Levitt. *Network intrusion detection*. IEEE Network, pages 26–41, June 1994.

P.A. Porras and R.A. Kemmerer. *Penetration state transition analysis: A rule-based intrusion detection approach*. In Proceedings of the 8th Annual Computer Security Applications Conference, San Antonio, Texas, December 1992.

F. Rabitti, E. Bertino, W. Kim, and D. Woelk. *A model of authorization for next generation database systems*. ACM Transactions on Database Systems, 16(1):88–131, 1994.

P. Liu S. Ingsriswang. Aaid: *An application aware transaction level database intrusion detection* system. Technical report, Department of Information Systems, UMBC, Baltimore, MD, 2001.

D. Samfat and R. Molva. Idamn: *An intrusion detection architecture for mobile networks*. IEEE Journal of Selected Areas in Communications, 15(7):1373–1380, 1997.

R. Sandhu and F. Chen. *The multilevel relational (mlr) data model.* ACM Transactions on Information and Systems Security, 1(1), 1998.

S.-P. Shieh and V.D. Gligor. On a pattern-oriented model for intrusion detection. IEEE Transactions on Knowledge and Data Engi-neering, 9(4):661–667, 1997.

M. Winslett, K. Smith, and X. Qian. Formal query languages for secure relational databases. ACM Transactions on Database Systems, 19(4):626–662, 1994.

P. A. Porras and R. A. Kemmerer. Penetration state transitionanalysis: A rule-based intrusion detection approach. In Proceedings of the Eighth Annual Computer Security Ap-plications Conference, pages 220–229, San Antonio, Texas, Nov. 30–Dec. 4, 1992.

P. Proctor. Audit reduction and misuse detection in heterogeneous environments: Framework and application. In Proceedings of the Tenth Annual Computer Security Applications Conference, pages 117–125, Orlando, Florida, Dec. 5–9, 1994.

M. M. Sebring, E. Shellhouse, M. E. Hanna, and R. A. Whitehurst. Expert systems in intrusion detection: A case study. In Proceedings of the 11th National Computer Security Conference, pages 74–81, Baltimore, Maryland, Oct. 17–20, 1988. National Institute of Standards and Technology/National Computer Security Center.

J. Habra, B. Le Charlier, A. Mounji, and I. Mathieu. ASAX: Software architecture and rule-based language for universal audit trail analysis. In Y. Deswarte et al., editors, Computer Security – Proceedings of ESORICS 92, volume 648 of LNCS, pages 435–450, Toulouse, France, Nov. 23–25, 1992. Springer-Verlag.

L. T. Heberlein et al. A network security monitor. In Proceedings of the 1990 IEEE Symposium on Security and Pri-vacy, pages 296–304, Oakland, California, May 7–9, 1990.

K. Ilgun. USTAT: A real-time intrusion detection system for UNIX. In Proceedings of the 1993 IEEE Symposium on Security and Privacy, pages 16–28, Oakland, California, May 24–26, 1993.

U. Lindqvist and P. A. Porras. Detecting computer and network misuse through the production-based expert system toolset (P-BEST). In Proceedings of the 1999 IEEE Symposium on Security and Privacy, pages 146–161, Oakland, California, May 9–12, 1999.

R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das. Analysis and results of the 1999 DARPA off-line intrusion detection evaluation. In H. Debar, L. M´ e, and S. F. Wu, editors, Recent Advances in Intrusion Detection (RAID 2000), volume 1907 of LNCS, pages 162–182, Toulouse, France, Oct. 2–4, 2000. Springer-Verlag.

A. Mounji. Languages and Tools for Rule-Based Distributed Intrusion Detection. PhD thesis, Institut d'Informatique, University of Namur, Belgium, Sept. 1997.

P. G. Neumann and P. A. Porras. Experience with EMERALD to date. In Proceedings of the 1st Workshop on Intrusion Detection and Network Monitoring, Santa Clara, California, Apr. 9–12, 1999. The USENIX Association.

A. One. Smashing the stack for fun and profit. Phrack Magazine, 7(49), Nov. 8, 1996. http://www.fc.net/phrack/files/ p49/p49-14.

J. Picciotto. The design of an effective auditing subsystem. In Proceedings of the 1987 IEEE Symposium on Security and Privacy, pages 13–22, Oakland, California, Apr. 27–29, 1987.