A real-time intrusion prevention system for commercial enterprise databases and file systems

Ulf T. Mattsson

Protegrity Corp.

Abstract. Modern intrusion detection systems are comprised of three basically different approaches, host based, network based, and a third relatively recent addition called procedural 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 surveillance. 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 services, 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

T. Mattsson U. (2005).
A real-time intrusion prevention system for commercial enterprise databases and file systems.
In *Proceedings of the 3rd International Workshop on Security in Information Systems*, pages 94-101
DOI: 10.5220/0002543200940101

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 (Griffiths, 1976, Rabitti, 1994), to inference control (Adam, 1989), to multilevel secure databases (Winslett, 1994, Sandhu, 1998), and to multi-level secure transaction processing (Atluri, 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 (Lunt, 1992, Jagannathan, 1993, Helman, 1993, Lunt, 1993, Mukherjee, 1994, Lunt, 1998, Lane, 1998, Lee, 1999). Intrusion detection systems monitor system or network activity to 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 (Javitz, 1991, Javitz, 1994, Samfat, 1997) or on known patterns of attacks, called signatures (Ilgun, 1993, Garvey, 1991, Porras, 1992, Ilgun, 1995, Shieh, 1997).

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.

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.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.

3.1 A New Approach

The solution protects the data in storage in a database. The architecture is built on top of a traditional COTS (Commercial-Of-The-Shelf) DBMS. 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. The Intrusion Protector prevents the damage using a rollback. The Intrusion Manager restricts the access to the objects that have been identified by the Intrusion Assessor as ‘under attack’, and unlocks an object after it is cleared by the security officer. The Policy Enforcement Agent (PEA) (a) functions as a filter for normal user transactions that access critical fields in the database, and (b) is responsible for enforcing system-wide intrusion prevention policies. For example, a policy may require the PEA to reject every new transaction submitted by a user as soon as the Intrusion Detector finds that the user submits a malicious transaction. It should be noticed that the system is designed to do all the intrusion prevention work on the fly without the need to periodically halt normal transaction processing.

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. 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. The Security Administrator (SA) monitor and control the Database Intrusion Prevention Policy rules via the Central Intrusion Prevention System. The Central Intrusion Prevention System also performs analysis of long-term audit transactions. The Local Intrusion Detection System performs a real-time analysis of online transactions. The Local Intrusion Prevention System performs real-time blocking of online transactions resulting from the combined analysis by the components described above.

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. This implementation provides a second type of intrusion detection, based on inference patterns, again resulting in a real time prevention of intrusion.

3.5 Real time analysis

Items (Fields and columns) are marked for monitoring in the policy database. The intrusion detection component compares the current query result and the updated intrusion detection record (record) with the item access rate included in the security policy associated with the current user, the role that the user belongs to, or the server the user is connected to. Only item access rates associated with the marked items comprised in the current result need to be compared. If the current query result or accumulated record includes a number of rows exceeding a particular item access rate, such a request will be classified as an intrusion, and the access control system (intrusion prevention system component) will be alerted. Secondly, if no item access rate is exceeded, the intrusion detection process compares the query result and accumulated record with any inference pattern included in the security policy. If the result includes a combination of items that match the defined inference pattern, such a request will also be classified as an intrusion, and the access control system will be alerted. If no intrusion is found, the result set is communicated to the user. Upon an alert, the access control system is arranged to immediately alter the user authorization, thereby making the submitted request unauthorized. For the user, the request, or at least parts of the request directed to items for which the item access rate was exceeded, will thus appear to be unauthorized, even though authority was initially granted by the access control system. In addition to the immediate and dynamic alteration of the access control system, other measures can be taken depending on the seriousness of the intrusion, such as sending an alarm to e.g. the administrator, or shutting down the entire database.

3.6 Long term analysis

The query result can also be stored in the central log file by the intrusion detection module, as described above. The central log file, which thus contains accumulated query results from a defined time period, can also be compared to the inference patterns in the security profiles of users, roles or servers, this time in a “after the event” type analysis. Even though such an analysis cannot prevent the intrusion from taking place, it may serve as intelligence gathering, improving the possibilities of handling intrusion problems. While the real time protection is most efficient when it comes to preventing security breaches, the long term analysis can be more in depth, and more complex, as time is no longer a critical factor. The real time protection system also receives feedback from the “after the event” type analysis to enable immediate alter of the user authorization, thereby making the submitted request unauthorized if the longer term item access rate was exceeded, or if longer term inference rules are violated. The policy may include additional policy rules in a practical implementation, including verification of the validity of time-of-day, authentication method, source of request (process, ID, port number, and the integrity of software components and metadata.

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 (Sebring, 1988), 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 (Proctor, 1994). USTAT (Ilgun, 1993) offered another formulation of intrusion heuristics using state transition diagrams (Porras, 1992), but by design remained a classic forward-chaining expert sys-tem inference engine. ASAX (Habra, 1992) introduced the Rule-based Sequence Evaluation Language (RUSSEL) (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

While the existing paradigms of computer security are still very useful and serve perfectly well in their capacities, there has existed a gap in the computer security space. 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. No one needs to be sacrificed to the new virus or worm anymore. In essence, we have learned to solve the right problem. 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 behavioural control is the missing piece in the security puzzle. The complete puzzle has three pieces; data control (encryption), access control, and behavioural control. This solution includes a method for detecting intrusion in a database, managed by an access control system, comprising defining at least one intrusion detection profile, each comprising at least one item access rate and associating each user with one of the profiles. Further, the method determines whether a result of a query exceeds any one of the item access rates defined in the profile associated with the user, and, in that case, notifies 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. 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.

References

1. Adam, 1989. M. R. Adam. Security-Control Methods for Statistical Database: A Comparative Study. ACM Computing Surveys, 21(4), 1989.
2. Atluri, 1999. V. Atluri, S. Jajodia, and B. George. Multilevel Secure Transaction Processing. Kluwer Academic Publishers, 1999.
3. Garvey, 1991) 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.
4. Griffiths, 1976. 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.
5. Helman, 1993. 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.
6. Ilgun, 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.
7. Ilgun, 1995. 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.
8. Jagannathan, 1993. R. Jagannathan and T. Lunt. System design document: Next generation intrusion detection expert system (nides). Technical report, SRI International, Menlo Park, California, 1993.
9. Javitz, 1991. 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.
10. Javitz, 1994. H. S. Javitz and A. Valdes. The nides statistical component description and justification. Technical Report A010, SRI International, March 1994.
11. Lane, 1998. 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.
12. Lee, 1999. 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.
13. Lunt, 1992. 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.
14. Lunt, 1998. Teresa Lunt and Catherine McCollum. Intrusion detection and response research at DARPA. Technical report, The MITRE Corporation, McLean, VA, 1998.
15. Lunt, 1993. T.F. Lunt. A Survey of Intrusion Detection Techniques. Computers & Security, 12(4):405–418, June 1993.
16. Mukherjee, 1994. B. Mukherjee, L. T. Heberlein, and K.N. Levitt. Network intrusion detection. IEEE Network, pages 26–41, June 1994.
17. Porras, 1992. 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.

18. Rabitti, 1994. 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.
19. Samfat, 1997. 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.
20. Sandhu, 1998. R. Sandhu and F. Chen. The multilevel relational (mlr) data model. ACM Transactions on Information and Systems Security, 1(1), 1998.
21. Shieh, 1997. 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.
22. Winslett, 1994. M. Winslett, K. Smith, and X. Qian. Formal query languages for secure relational databases. ACM Transactions on Database Systems, 19(4):626–662, 1994.
23. Habra, 1992. 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.
24. Ilgun, 1993. 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.
25. Mounji, 1997. A. Mounji. Languages and Tools for Rule-Based Distributed Intrusion Detection. PhD thesis, Institut d'Informatique, University of Namur, Belgium, Sept. 1997.
26. Porras, 1992. 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.
27. Proctor, 1994. 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.
28. Sebring, 1988. 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.