AN ARCHITECTURAL DESIGN FOR A HYBRID INTRUSION DETECTION SYSTEM FOR DATABASE

MOHAMMAD HOSSEIN HARATIAN

A project report submitted in partial fulfillment of the

requirements for the award of the degree of

Master of Computer Science (Information Security)

Centre for Advanced Software Engineering (CASE)
Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia

APRIL 2009

ABSTRACT

In today's business world, information is the most valuable asset of organizations and thus requires appropriate management and protection. Amongst all types of data repositories, database is said to play the role of the heart in the body of IT infrastructure. On the other hand, nowadays, a growing number of efforts have concentrated on handling the vast variety of security attacks. The characteristic of such handling method depends on when we want it to be occurred and how we intent to deal with attack attempts. Generally there are two ways to handle subversion attempts. One way is to equip our systems by security controls. However in reality this is not feasible due to many reasons. Hence, we are interested in detecting the security attacks. Amongst different types of intrusion detection systems (like network-based, host-based and application-based IDS), database intrusion detection systems which are considered as a type of application-based IDS has become a matter of increasing concern.

In this paper we proposed the architecture for a hybrid database intrusion detection system (DB-IDS). This architecture consists of several component and sub-components. It encompasses Anomaly Detection and Misuse Detection sub-components as Detector component. Anomaly detection component works based on the Profiles constructed by Profiler. Suspicious sequence of events which are considered as potential attacks would be detected by Misuse Detector. Data Collector components is responsible for capturing necessary data for profiling. Moreover, the Transformer component is in place to convert the raw log files into an understandable format for Profiler. Finally, Anomaly Detector and Misuse Detector components send alert to Responder component in case of detection any suspicious activity.

ABSTRAK

Dalam urusan seharian kini, maklumat merupakan harta yang paling penting bagi sesebuah organisasi. Oleh itu ia memerlukan pergurusan dan perlindungan yang cekap. Diantara jenis-jenis penyimpanan data, pengkalan data merupakan bahagian utama bagi sesebuah rangka infrastruktur untuk teknologi maklumat. Oleh yang demikian, berbagai tindakan telah dilaksanakan bagi membendung masalah serangan keselamatan yang berleluasa. Cara pengurusan masalah ini bergantung kepada bagaimana kita mahu ia dilakukan dan bagaimana kita akan berurusan dengan percubaan serangan keselamatan tersebut. Secara umumnya terdapat dua kaedah bagi menguruskan masalah percubaan 'subversion'. Salah satu cara adalah dengan melengkapkan peralatan system komputer yang sediaada dengan system kawalan keselamatan. Walaubagaimanapun, kaedah ini sukar dilaksanakan dewasa ini disebabkan oleh masalah kewangan dan masalah kekurangan sumber manusia. Oleh itu, mengesan masalah keselamatan merupakan tujuan utama kajiselidik ini. Diantara cara mengesan masalah keselamatan yang dikenalpasti adalah Sistem Pengesanan Pencerobohan (IDS) seperti Sistem Pengesanan Pencerobohan bagi dasar-rangkaian, dasar-kerangka utama, dasar-aplikasi, dan Sistem Pengesanan pencerobohan pengkalan data. Perkara ini semakin hangat dibincangkan bagi meningkatkan lagi mutu kawalan keselamatan bagi sesebuah sistem komputer. Dalam kajian ini, kami mencadangkan sistem arkitektur bagi Sistem Pengesanan pencerobohan pengkalan data 'hybrid' (DB-IDS). Arkitetur ini mengandungi beberapa komponen dan cabang komponen. Ia termasuk Pengesanan 'Anomaly' dan Pengesanaan Penyalahgunaan yang dikenali sebagai komponen pengesanan. Pengesanan 'anomaly' melaksanakan tugasnya berdasakan profil yang dijana oleh 'profiler' Aktiviti-aktiviti yang mencurigakan di kenali sebagai kemungkinan serangan akan dikesan oleh Pengesan Penyalahgunaan. Komponen pengumpulan Data berfungsi bagi mengenalpasti data untuk 'profiling'. Komponen 'transformer' pula bersedia untuk menukar fail asal 'log' kepada format yang boleh dibaca oleh

‘profiler’. Komponen Pengesan ‘Anomaly’ dan Pengesan Penyalahgunaan akan menghantar maklumat pengawasan kepada komponen penerima (Responder) bila terdapat aktiviti-aktiviti yang mencurigakan.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Overview

Nowadays, a growing number of efforts have concentrated on handling the vast variety of security attacks. The characteristic of such handling method depends on when we want it to be occurred and how we intent to deal with attack attempts. According to [1], generally there are two ways to handle subversion attempts. One way is to equip our systems by all security controls such as cryptographic methods, sophisticated access control mechanisms, rigorous authentication protocols and etc. to prevent the subversion itself. However in reality this is not feasible due to many reasons, for example, (a) flaws of cryptographic techniques, (b) trade-off between efficiency and the level of access control and (c) insiders who abuse their privileges. Doubtlessly, it is very important that the security mechanism of a system is designed so as to *prevent* unauthorized access to system resources and data. However, as it was mentioned before completely preventing breaches of security appear, at present, unrealistic. We can, however, try to detect these intrusion attempts so that action may be taken to repair the damage later [1] and this is what the **Intrusion Detection** refers to.

Generally there are two types of intrusion detection techniques[1]. One is named Anomaly Detection technique in which a profile is established for the system and any activity that cause a deviation from the normal activity profile would be flagged as an intrusion. This method may rely on statistical approaches or predictive

pattern generation. Another technique which is called Misuse Detection mostly is based on signature or patterns of attacks.

In both techniques, however, the Artificial Intelligence [2] and Data Mining [3] [4] [5] applications may be employed to reduce the human effort and to increase the accuracy of the detection. In recent years, Data Mining-based intrusion detection systems (IDSs) have demonstrated high accuracy, good generalization to novel types of intrusion, and robust behavior in a changing environment [6].

Although a variety of approaches have been proposed for enhancing the capabilities of intrusion detection as well as the efficiency and accuracy, most of these efforts concentrated on detecting intrusions at the network or operating system level (refers to Network-based and Host-based intrusion detection system respectively). They are not capable of detecting malicious data corruptions, that is, what particular data in the database are manipulated by which specific malicious database transaction(s) [7]. So, this opens the issue of detecting the intrusions at the database level.

1.2 Background

In today's business world, information is the most valuable asset of organizations and thus requires appropriate management and protection [8]. Amongst all types of data repositories, *database* is said that play the role of the heart in the body of IT infrastructure. They not only allow the efficient management and retrieval of huge amounts of data, but also because they provide mechanisms that can be employed to ensure the integrity of the stored data [8].

Thus, obviously databases always have been the interesting target of attacks for hackers. Getting access to a database containing hundreds thousands of credit card numbers is almost every hacker’s dream. This is what indicates the violation of confidentiality; however, an intrusion can be defined as “any set of actions that

attempt to compromise the integrity, confidentiality or availability of a resource" [9]. So, enough and balanced care should be taken to protect the whole of this triad. Attack reports have been released [10] in which the intruder had updated the field of the price in an on-line store website and decreased the values of specific items, and then bought those items for just few dollars (integrity violations).

According to [11], The Privacy Rights Clearinghouse reports that during the period from January 2005 to May 2007, more than 154 million records containing sensitive information, including credit card numbers, Social Security numbers, bank account numbers, and drivers license numbers, were stolen from United States organizations. The actual total could be much higher. This number only represents reported breaches and in many cases, the total records compromised remain undetermined. Approximately one third of the reported breaches were the result of a direct attack on the database.

Hence, today, the critical need for securing the databases has become much more inevitable than any time before. *Database Security* is as old as the emergence of the own database concept and encompasses a broad range of procedures that protect a database from unintended activity. One of the most important techniques for securing the database is applying the Intrusion Detection System which is used to detect potential violations in database security.

Anderson [12] has classified intruders into two types, the *external* intruders who are unauthorized users of the machines they attack, and *internal* intruders, who have permission to access the system, but not some portions of it. He further divided internal intruders into intruders who *masquerade* as another user, those with *legitimate* access to sensitive data, and the most dangerous type, the *clandestine* intruders who have the power to turn off audit control for them. Despite the necessity of protecting information stored in database systems (DBS), existing security models are insufficient to prevent misuse, especially insider abuse by legitimate users [8]. The external intrusions are supposed to be detected and handled by network-based IDSs. However, when it comes to the database level, an intruder

cannot do anything otherwise he gets access to a valid credential and login to the system. Means, the transactions must be issued by a valid database user, who has logged in using a valid database login, no matter the login information is provided by a legitimate user or not. Here, when we use the term of the "valid database user", it doesn't indicate that this database user is necessarily associated with a legitimate actual user. For example by using "social engineering" techniques, the intruder can get access to some valid database user information.

Forrester Research estimates that nearly 80 percent of all database attacks are internal and Gartner estimates that more than 95 percent of intrusions that result in significant financial loss are perpetrated by insiders [11]. If one intruder gets access to account information of a legitimate database user, (s)he may cause damage to the database by executing transaction(s) that illegitimately manipulate the sensitive data. In such case, the external intruder becomes an internal one who masquerades as another user. In this scenario, identifying whether the data corruption indeed has been done by legitimate user or by one who has got access to a legitimate user's account information is tough.

Risk from insiders comes in many forms and as attackers recognize the value and importance of the information in the database, attacks are becoming more focused. Attackers have also changed. In the past people hacked into networks to "prove they could." While those attacks were malicious, recently the motivation has become financial. Attackers are seeking data to sell and that information resides in the database [11].

Another common type of intrusions forensically analyzed in [10] is that one the intruder logins to the database system as a high-privileged user (by brute force, for example) and then creates an account for himself and starts to manipulate the database by logging in as new-created account.

On the other hand, the illegal transactions executed by legitimate database users who are not authorized to perform certain activities—and for any reason, the

logical permissions of these activities have been not denied for those users—seems to be more difficult to be detected. Carter and Katz [13] have revealed that in computer systems the primary security threat comes from insider abuse rather than from intrusion. This observation results in the fact that much more emphasis has to be placed on internal control mechanisms of systems. Furthermore, policies usually do not sufficiently guard data stored in a database system against privileged users [8] like sa and members of sysadmin fixed server role in MS SQL Server 2005 for example.

1.3 Problem Statement

As it was mentioned before, a great number of database attacks come from inside the organization, either by privileged users, authorized users or unauthorized users who hack into the system by gaining access to legitimate accounts. In either case, we intend to be able to detect the attacks conducted by each of these intruder categories. This question may be raised:

How can we enable database management systems to monitor, detect, mitigate or/and prevent the attacks using some tools like DB-IDS and/or its built-in capabilities?

Moreover, security policies often fail to prevent the database attacks. There have been many scenarios in which authorized users are inadvertently granted access to run certain operations. Initial database security configuration often fails to comply with security policies of organizations. Users usually hold privileges which are not supposed to be granted to them. We can assume that in a realistic environment none of those users ever exceed their rights and use those privileges. But what if once a hacker steals those accounts and enters to the system? In that case we cannot guarantee that hacker adhere to the ethics. Now the additional question is:

How the database intrusion detection system may aid to revise the security policies?

1.4 Project Aim

The aim of this project is to design an architecture for a hybrid intrusion detection system for database. This architecture is containing different components and sub-components which interact to each other. The system is called hybrid DB-IDS since it encompasses anomaly detector and misuse detector modules. The proposed architecture could be adapted for any DBMS with consideration its features and capabilities. Thought, we develop a model based on this architecture for MS SQL Server 2005 to show how it works. Moreover, leveraging the information provided by our DB-IDS, database security policies could be revised to strengthen the system.

1.5 Project Objectives

As mentioned before, our hybrid DB-IDS consists of anomaly and misuse detectors. In order to detect anomalies, a normal activity profile is created for specific database objects. These objects may include but not limited to *principals* and *securables*[1]. Any considerable deviation from captured normal behavior of the system may be thought as an intrusion. Our model is designed in a way that enables us to apply different methods range from statistical measures to artificial intelligence techniques to build system profiles.

---

[1] - The SQL Server 2005 security model relies on two fairly straightforward concepts: principals and securables. Principals are those objects that may be granted permission to access particular database objects, while securables are those objects to which access can be controlled [14].

To capture the behavior of database objects, we need to monitor and audit the system operation. This auditing system helps us to collect necessary data for building database profiles. To be more accurate, whatever technique the profiler utilizes to build the profiles, data gathered by auditing system provides necessary input for it.

A security alert is raised in case of any anomaly and misuse detection. Depending on the suspicious level or sensitivity of intrusion, detection mechanism can contribute to Access Control system to deny access and prevent the intruder from causing further corruption. However, although such capability is in place, the system is not supposed to entirely function as an intrusion prevention system.

Another objective of this project is to help to revise database security policies and configurations by providing daily bases reports. Based on the facts these reports provide, database use policies can be changed, modified or even removed. Furthermore, the reports may help us to create new database security policies and/or re-configure the database security schema.

In the following we accordingly list the project objectives with the aim of designing an architecture for DB-IDS:

i. Proposing a database anomaly detection model

ii. Constructing sample profiles for a database system

iii. Developing an database audit system

iv. Proposing a brute-force detection model for the database systems

v. Proposing a model for database security policy revision

1.6 Project Scope

As we said before, intrusion detection systems do not usually take preventive measures when an attack is detected; it is a reactive rather than pro-active agent. It plays the role of an informant rather than a police officer [1]. So, the proposed architecture only takes into account the detection of an intrusion. Though, the model is capable to contribute with database access control system to stop the intruder from making further damages to the system.

The architecture design of DB-IDS would be developed implemented and tested for Microsoft SQL Server 2005 using Structured Query Language (SQL). All components of the proposed architecture for DB-IDS are built using built-in means including stored procedures, views and triggers and so on.

The range of the intrusions detectable by this model is limited to those which have been conducted by either the external intruders who gets access to legitimate database account information, or those insiders who abuse their privileges. External intrusions and attacks like SQL injection will not be covered in this project.

In the last chapter of this project report, we go through the recommendations and future works to enhance the accuracy, efficiency and scalability of proposed DB-IDS. As the matter of fact, what is already beyond the scope of this project could be considered as future works.

1.7 Summary

Nowadays, intrusion detection plays a vital role in security mechanisms. Organizations need to be able to detect the intrusions into their database systems as soon as they can to prevent further damages to their sensitive data which may cause financial loss. The critical necessity of having intrusion detection system in place is highly arisen when we hear that in many real life scenarios the intrusion remains

undetected for hours and even days. These concealed attacks are thought to be the most horrible DBA's nightmare which makes the recovery procedure too difficult and time-consuming and even in some cases infeasible. On the other hand, from forensic point of view, intactness of evidences is the key point in database intrusion investigation, while the delay in detection of intrusion may lead to corruption of digital evidences on which the guilt of intruder is based.

In this project we intend to come up with an architecture model for database intrusion detection which tries to detect the suspicious transactions by comparing the normal activity of the system to the current transaction. As we saw in the introduction section, such systems are named Anomaly Detection [1]. However, there are challenges in the design of these systems such as selection of threshold levels so that minimize the false negatives and false positives and selection of features to monitor. Furthermore, Anomaly detection systems are also computationally expensive because of the overhead of keeping track of, and possibly updating several system profile metrics [1]. The proposed architecture is intended to be designed in such a way that address these challenges and balance the efficiency and overhead.

The other mechanism of detecting database intrusions in this project relays on the Misuse Detection concept in which we intend to identify the meaningful sequence of events that turns to be a kind of database misuse. Misuse detection works based on database attack patterns. The model enables us to monitor all apparently irrelevant small events that if occur in a specific order, we may believe the database is the target of an intrusion.

REFERENCES

1. Sundaram, A., *An Introduction to Intrusion Detection.* 1996.
2. Frank, J., *Artificial Intelligence and Intrusion Detection: Current and Future Directions.* In Proceedings of the 17th National Computer Security Conference, 1994.
3. Lee, W., et al., *A Data Mining and CIDF Based Approach for Detecting Novel and Distributed Intrusions.* In Proceedings of 3rd International Workshop on the Recent Advances in Intrusion Detection, 2000.
4. Lee, W. And S.J. Stolfo, *Data Mining Approaches for Intrusion Detection.* in the Proceedings of the 7th USENIX Security Symposium San Antonio, Texas, 1998.
5. Lee, W., *Applying Data Mining to Intrusion Detection: the Quest for Automation, Efficiency, and Credibility.*
6. Campos, M.M. And B.L. Milenova, *Creation and Deployment of Data Mining-Based Intrusion Detection Systems in Oracle Database 10g.*
7. Hu, Y. And B. Panda, *A Data Mining Approach for Database Intrusion Detection. A*CM Symposium on Applied Computing, 2004.
8. Chung, C.Y., M. Gertz, and K. Levitt, *DEMIDS: A Misuse Detection System for Database Systems.* In Third Annual IFIP TC-11 WG 11.5 Working Conference on Integrity and Internal Control in Information Systems, 1999.
9. Heady, R., et al., *The architecture of a network level intrusion detection system.* 1990.
10. Fowler, K., *Forensic Analysis of a SQL Server 2005 Database Server.* 2007.
11. (2009) *Addressing the Insider Threat, Improving Database Security to Manage Risk within the Federal Government*.
12. *Computer Security Threat Monitoring and Surveillance.* 1980.
13. Carter, D.L. And A.J. Katz, *Computer Crime: An Emerging Challenge for Law Enforcement.* FBI Law Enforcement Bulletin, 1997: p. 1-8.
14. Labib, K., *Computer Security and Intrusion Detection*, in *The ACM Student Magazine*.
15. Bace, R. And P. Mell, *NIST Special Publication on Intrusion Detection Systems.*
16. Alessandri, D., *Towards a Taxonomy of Intrusion Detection Systems and Attacks.* MAFTIA deliverable D3, 2001.
17. Axelsson, S., *Intrusion Detection Systems : A Survey and Taxonomy.* 2000.

18. Fuchsberger, A., *Intrusion Detection Systems and Intrusion Prevention Systems.* Information Security Technical Report, 2005. **10**: p. 134-139.
19. Kemmerer, R.A. And G. Vigna, *Intrusion Detection : A Brief History and Overview.* 2002.
20. Allen, J., et al., *State of the Practice of Intrusion Detection Technologies.* TECHNICAL REPORT CMU/SEI-99-TR-028, 2000.
21. Debar, H., M. Dacier, and A. Wespi, *Towards a Taxonomy of Intrusion Detection Systems.* Computer Networks, 1999. **31**.
22. Halme, L.R. And R.K. Bauer, *AINT Misbehaving: A Taxonomy of Anti-Intrusion Techniques.*
23. Srivastava, A., S. Sural, and A.K. Majumdar, *Database Intrusion Detection using Weighted Sequence Mining.* JOURNAL OF COMPUTERS, 2006. **1**(4).
24. Lunt, T., et al., *A real-time intrusion detection expert system (IDES).* 1992.
25. Kumar, S. And E.H. Spafford, *A software architecture to support misuse intrusion detection.* Proceedings of the 18th National Information Security Conference, 1995: p. 194–204.
26. Ilgun, K., R.A. Kemmerer, and P.A. Porras, *State transition analysis: A rule-based intrusion detection approach.* IEEE Transactions on Software Engineering, 1995.
27. Marc, G.W. And H. Andrew, *Interfacing Trusted Applications with Intrusion Detection Systems*, in *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*. 2001, Springer-Verlag.
28. *Access Control from an Intrusion Detection Perspective.*
29. Fayyad, U., G.P. Shapiro, and P. Smyth, *The KDD Process for Extracting Useful Knowledge from Volumes of Data.* Communications of the ACM, 1996.
30. Lee, W., S.J. Stolfo, and K.W. Mok, *A Data Mining Framework for Building Intrusion Detection Models.*
31. Pietraszek, T. And A. Tanner, *Data mining and machine learning - Towards reducing false positives in intrusion detection.* Information Security Technical Report, 2005.
32. Stolfo, S., et al., *JAM: Java agents for Meta-Learning over Distributed Databases.* 1997.
33. Lee, W., *A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems*, in *Graduate School of Arts and Sciences*. 1999, COLUMBIA UNIVERSITY.
34. Helmer, G., J. Wong, and V.H.L. Miller, *Automated Discovery of Concise Predictive Rules for Intrusion Detection.* 1999.
35. Daniel, B., et al., *ADAM: a testbed for exploring the use of data mining in intrusion detection.* SIGMOD Rec., 2001. **30**(4): p. 15-24.

36. Abraham, T., *IDDM: Intrusion Detection using Data Mining Techniques.* 2000.
37. Lazarevic, A., et al., *A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection.* in Proc. Third SIAM International Conference on Data Mining, San Francisco, 2003.
38. Valdes, A. And K. Skinner, *Adaptive, Model-based Monitoring for Cyber Attack Detection.* SRI International.
39. Daniel, B., *Applications of Data Mining in Computer Security*, ed. J. Sushil. 2002: Kluwer Academic Publishers. 272.
40. Lee, W., S.J. Stolfo, and K.W. Mok, *Mining Audit Data to Build Intrusion Detection Models.* in Proc. Fourth International Conference on Knowledge Discovery and Data Mining, NewYork, 1998.
41. Barbara, D., et al., *ADAM: Detecting Intrusions by Data Mining.* Proceedings of the 2001 IEEE Workshop on Information Assurance and Security, 2001.
42. Barbar´a, D., N. Wu, and S. Jajodia, *Detecting novel network intrusions using bayes estimators.* in Proc. First SIAM Conference on Data Mining, Chicago, 2001.
43. *SNORT, SNORT Intrusion Detection System*. [cited; Available from: http://www.snort.org.
44. Anoop, S. And J. Sushil, *Data warehousing and data mining techniques for intrusion detection systems.* Distrib. Parallel Databases, 2006. **20**(2): p. 149-166.
45. Burbeck, K. And S. Nadjm-Tehrani, *Adaptive real-time anomaly detection with incremental clustering.* information security technical report 12, 2007.
46. Portnoy, L., *Intrusion detection with unlabeled data using clustering.*
47. Leung, K. And C. Leckie, *Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters.* 28th Australasian Computer Science Conference, The University of Newcastle, Australia, 2005.
48. Shah, H., J. Undercoffer, and A. Joshi, *Fuzzy Clustering for Intrusion Detection.* The IEEE International Conference on Fuzzy Systems, 2003.
49. Fonseca, J., M. Vieira, and H. Madeira, *Monitoring Database Application Behavior for Intrusion Detection.* 12th Pacific Rim International Symposium on Dependable Computing, 2006.
50. MATTSSON, U.T., *A Practical Implementation of a Real-time Intrusion Prevention System for Commercial Enterprise Databases.*
51. Jin, X. And S.L. Osborn, *Architecture for Data Collection in Database Intrusion Detection Systems.* 2007.
52. Hu, Y. And B. Panda, *Identification of Malicious Transactions in Database Systems.* Proceedings of the Seventh International Database Engineering and Applications Symposium (IDEAS'03), 2003.
53. Rietta, F.S., *Application Layer Intrusion Detection for SQL Injection. A*CM Symposium on Applied Computing, 2006.

54. Dai, J. And H. Miao, *D_DIPS: An Intrusion Prevention System for Database Security*. 2005.
55. LOW, W.L., J. LEE, and P. TEOH, *DIDAFIT: DETECTING INTRUSIONS IN DATABASES THROUGH FINGERPRINTING TRANSACTIONS.*
56. Wenhui, S. And D. Tan T H, *A Novel Intrusion Detection System Model for Securing Web-based Database Systems.* 25th Annual International Computer Software and Applications Conference (COMPSAC'01), 2001: p. 249.
57. Lee, V.C.S., J.A. Stankovic, and S.H. Son, *Intrusion Detection in Real-time Database Systems Via Time Signatures.* In Proceedings of the Sixth IEEE Real Time Technology and Applications Symposium, 2000.
58. Elisa, B., et al., *Intrusion Detection in RBAC-administered Databases*, in *Proceedings of the 21st Annual Computer Security Applications Conference*. 2005, IEEE Computer Society.
59. Fonseca, J., M. Vieira, and H. Madeira, *Integrated Intrusion Detection in Databases.* 2007.
60. Ramasubramanian, P. And A. Kannan, *A genetic-algorithm based neural network short-term forecasting framework for database intrusion prediction system* 2005.
61. Asmawi, A. And Z.M. Sidek, *A Survey on Artificial Immune System-Based Intrusion Detection System for DBMS.* Postgraduate Annual Research Seminar, 2007.
62. Chen, K., G. Chen, and J. Dong, *An Immunity-Based Intrusion Detection Solution for Database Systems.*
63. Valeur, F., D. Mutz, and G. Vigna, *A Learning-Based Approach to the Detection of SQL Attacks.*
64. Lee, S.Y., W.L. Low, and P.Y. Wong, *Learning Fingerprints for a Database Intrusion Detection System.* 2002.
65. Srivastava, A., S. Sural, and A.K. Majumdar, *Weighted Intra-transactional Rule Mining for Database Intrusion Detection.* 2006.
66. Barbara, D., R. Goel, and S. Jajodia, *Mining malicious data corruption with hidden markov models.* in Research Directions in Data and Applications Security, 2002.
67. ZHONG, Y. And X.-L. QIN, *RESEARCH ON ALGORITHM OF USER QUERY FREQUENT ITEMSETS MINING.* Proceedings of the Third International Conference on Machine Learning and Cybemetics, Shanghai, 2004.
68. Zhong, Y. And X.-l. Qin, *Database Intrusion Detection Based on User Query Frequent Itemsets Mining with Item Constraints.* Conference InfoSecu04, 2004.
69. Spalka, A. And J. Lehnhardt, *A Comprehensive Approach to Anomaly Detection in Relational Databases.* 2005.

70. Mattsson, U.T., *A REAL-TIME INTRUSION PREVENTION SYSTEM FOR COMMERCIAL ENTERPRISE DATABASES AND FILE SYSTEMS.*
71. Ramasubramanian, P. And A. Kannan, *Intelligent Multi-agent Based Database Hybrid Intrusion Prevention System.* 2004.
72. Mauch, J. And N. Park, *Guide To The Successful Thesis And Dissertation - A Handbook For Students And Faculty*. 2003: Routledge, USA.
73. Kothari, C.R., *Research Methodology: Methods & Techniques*. 2005: New Age Publishers.
74. Chung, C.Y., M. Gertz, and K. Levitt, *Misuse Detection in Database Systems Through User Profiling.*
75. ZhaoHui Tang and, J.M., *Data Mining with SQL Server 2005*. 2005: Wiley Publishing, Inc.
76. Utley, C. (2005) *Introduction to SQL Server 2005 Data Mining*.
77. Asanka, D. (2004) *Basics of C2 Auditing*.
78. Rizzo, T., et al., *Pro SQL Server 2005*. 2006: Apress.
79. Higgins, K.J. (2008) *Hacker's Choice: Top Six Database Attacks*.