[INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY AND MANAGEMENT VOL.-I, ISSUE - I] August 1, 2011

ISSN-2249-4510

A Real-Time Intrusion Prevention System for Databases and File Systems

Shalini Goel

Research Scholar, Singhania University, Rajasthan , India

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 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

ISSN-2249-4510

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.

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, to inference control, to multilevel secure databases, and to multi-level secure transaction processing, 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. 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 or on known patterns of attacks, called signatures. 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.

ISSN-2249-4510

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 nonsensitive data. It is further possible to assign different encryption keys of the same algorithm to different data columns. With

ISSN-2249-4510

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

ISSN-2249-4510

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. Thus, it is important to further separate the DBA's and the SA's privileges. For instance, if services are outsourced, the owner of the database contents may trust a vendor to administer the database.

Then the role of the DBA belongs to an external person, while the important SA role is kept within the company, often at a high management level. Thus, there is a need for preventing a DBA to impersonate a user in an attempt to gain access to the contents of the database. 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

ISSN-2249-4510

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 systemwide 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.

ISSN-2249-4510

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. The Intrusion Prevention System may be connected to enforce field-level Intrusion Prevention Policy Rules on a File System, Database System, or other data store with field level organization. The Intrusion Prevention System may be connected to Database System the level of views, triggers, database-proxy, connectivity products (ODBC, JDBC, …), or other points where the field or column level transactions can be controlled.

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

ISSN-2249-4510

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 revented, 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 result will be restricting the queries accepted from a group of users at one time or over a period of time. The user, role and server entities are not exclusive of other entities which might benefit from a security policy. According to an implementation of the method, items subject to item access rates are marked in the database, so that any query concerning the items automatically can trigger the intrusion detection process. This is especially advantageous if only a few items are intrusion sensitive, in which case most queries are not directed to such items.

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

ISSN-2249-4510

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.

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

ISSN-2249-4510

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 [50], 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 [48]. USTAT [39] offered another formulation of intrusion heuristics using state transition diagrams [46], but by design remained a classic forwardchaining expert sys-tem inference engine. ASAX introduced the Rule-based Sequence Evaluation

ISSN-2249-4510

Language (RUSSEL), which is tuned specifically for the analysis of host audit trails. Recent literature form the RAID conferences, as well as IEEE Security and Privacy, he 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 this hybrid solution 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 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

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

[2] P. Ammann, S. Jajodia, and P. Liu. *Recovery from malicious trans-actions*. IEEE Transactions on Knowledge and Data Engineering, 2001. To appear.

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

ISSN-2249-4510

[4] 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.

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

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

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

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

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

ISSN-2249-4510
[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

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

[20] 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.

[21] 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.

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

ISSN-2249-4510

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

[24] 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.

[25] 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.

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

[27] 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.

[28] 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.

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

[30] 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.

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

[32] 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.

ISSN-2249-4510

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

[34] The U.S. Health Information Portability and Accountability Act (HIPAA) - compliance by October 2002 www.hipaacomply.com

[35] The European Union 95/46/EC Directive on Data rivacy - compliance October 1998 - and individual EU member state privacy legislation - various compliance dates http://europa.eu.int/comm/internal_market/en/dataprot/

[36] EU/US Safe Harbor - compliance 11/1/2000 www.export.gov/safeharbor http://europa.eu.int/comm/internal_market/en/dataprot/m odelcontracts/index.htm

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das. Analysis and results of the 1999 DARPA offline intrusion detection evaluation. In H. Debar, L. M´ e, and S. F. Wu,

ISSN-2249-4510

editors, Recent Advances in Intrusion Detection (RAID 2000), volume 1907 of LNCS, pages 162–182, Toulouse, France, Oct. 2–4, 2000. Springer-Verlag.

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

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] P. A. Porras and P. G. Neumann. EMERALD: Event monitoring enabling responses to anomalous live disturbances. In Proceedings of the 20th National Information Systems Security Conference, pages 353–365, Baltimore, Maryland, Oct. 7–10, 1997. National Institute of Standards and Tech-nology/National Computer Security Center.

[48] 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.

ISSN-2249-4510

[49] T. H. Ptacek and T. N. Newsham. Insertion, evasion, and denial of service: Eluding network intrusion detection. Technical report, Secure Networks, Inc., Calgary, Alberta, Canada, Jan. 1998. http://www.clark.net/˜roesch/idspaper.html.

[50] 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.

[51] Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, CA 94303, USA. SunSHIELD Basic Security Module Guide, Solaris 7, Oct. 1998. Part No. 805-2635-10.

[52] U.S. Department of Defense. Trusted Computer System Evaluation Criteria, Dec. 1985. DoD 5200.28-STD.

[53] A. Valdes and K. Skinner. Adaptive, model-based monitoring for cyber attack detection. In H. Debar, L. M´ e,and S. F. Wu, editors, Recent Advances in Intrusion Detection (RAID 2000), volume 1907 of LNCS, pages 80–92,Toulouse, France, Oct. 2–4, 2000. Springer-Verlag.