

Ron Ben Natan

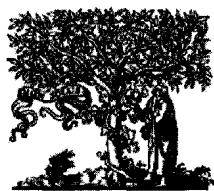
Implementing Database Security and Auditing

Includes
Examples For:

**ORACLE
SQL SERVER
DB2 UDB
SYBASE**

Implementing Database Security and Auditing

**A guide for DBAs, information
security administrators and auditors**



ELSEVIER

**DIGITAL
PRESS**

Amsterdam • Boston • Heidelberg • London • New York • Oxford
Paris • San Diego • San Francisco • Singapore • Sydney • Tokyo

Elsevier Digital Press
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA
Linacre House, Jordan Hill, Oxford OX2 8DP, UK

Copyright © 2005, Elsevier Inc. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

Permissions may be sought directly from Elsevier's Science & Technology Rights Department in Oxford, UK: phone: (+44) 1865 843830, fax: (+44) 1865 853333, e-mail: permissions@elsevier.com.uk. You may also complete your request on-line via the Elsevier homepage (<http://elsevier.com>), by selecting "Customer Support" and then "Obtaining Permissions."

∞ Recognizing the importance of preserving what has been written, Elsevier prints its books on acid-free paper whenever possible.

Library of Congress Cataloging-in-Publication Data
Application submitted.

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

ISBN: 1-55558-334-2

For information on all Elsevier Digital Press publications
visit our Web site at www.books.elsevier.com

Printed in the United States of America
05 06 07 08 09 10 10 9 8 7 6 5 4 3 2 1

Working together to grow libraries in developing countries		
www.elsevier.com www.bookaid.org www.sabre.org		
ELSEVIER	BOOK AID International	Sabre Foundation

Preface

This book is a guide on implementing security and auditing for database environments. It is meant to be used by database administrators, security administrators, system administrators, auditors, and operational owners—anyone who manages or oversees the database environment, data/database security, or the process by which database security and database audits are accomplished.

The book shows you how to secure and audit database environments which include the major relational products: environments, which include the major relational database products: Oracle, Microsoft SQL Server, IBM DB2, Sybase, and even a bit of MySQL. It is useful if you have a single database product and is even more useful if you need to secure and/or audit heterogeneous environments that include more than one database version. The methods you will learn apply to all modern relational database environments.

This book is meant to show you *methods* and *techniques* that will help you elevate the security of your database infrastructure. Each chapter in the book focuses on a certain area of database administration and usage and shows you what you need to do in that domain, as well as how to do it. Because educated administrators are sure to be more effective than those that follow checklists with a limited understanding of what each item does and why, each chapter details anatomies of vulnerabilities in addition to the remedies. By understanding how attackers may try to compromise the database, you will be better able to invest your limited resources where they count most. You may even be able to address issues that are not mentioned in this book and that may not even be known at this point in time.

I mentioned that the aim of this book is to make your database environment more secure and that the focus is often both administration and usage. Many database vulnerabilities and security issues are caused by misconfigurations and inappropriate usage of the database by application serv-

ers and other clients (or even other databases in replicated and other distributed environments). In addressing this topic, many of the chapters take a broader look of database security and show you how to resolve problems by improving the way the database interacts with applications and with other elements in the infrastructure. Without understanding these techniques, you may invest a lot of time in securing “your island,” only to learn that you have a gaping hole—one that you could have easily addressed if you weren’t too busy investing in perfecting your corner of the world. The book is therefore not only meant to be a practical guide, but it also means to be an *effective* guide and address real-world problems.

This book is not a checklist. Detailed instructions are included in almost all chapters, but the book is not a reference text for each of the database products. I will include pointers to relevant checklists and reference texts and instead focus on ensuring that you invest your time wisely. Security is a never-ending battle against would-be attackers, and if you don’t pick your fights wisely, you can lose to attrition. Auditing is another area that can easily overwhelm you in terms of work. Therefore, I will try to highlight the most important areas in which you should invest your time, show you what to do, and how to do it.

I mentioned that each chapter addresses a certain area—or category of techniques. This means that in most cases you can read the book sequentially or skip directly to a particular chapter when you are starting an initiative that has a specific focus. As an example, if you plan to start an initiative focused on database encryption, you should read Chapter 10; if you are concerned with database links, synonyms, nicknames, or replication, skip to Chapter 8; and if you are concerned with Web application access to your database, you can start with Chapter 5. The chapters that discuss auditing (Chapters 11 through 13) are a bit different. Rather than discussing categories of *techniques* as do Chapters 3 through 10, each chapter on the topic of auditing focuses on database auditing from a different *perspective*: Chapter 11 from the perspective of mapping of business requirements or regulations to actionable audit tasks, Chapter 12 from a content perspective, and Chapter 13 from an architectural perspective. Chapters 1 and 2 are introductory chapters. Chapter 1 details some starting points you should always have in place, and Chapter 2 gives you a brief overview of enterprise security and domains from which you can get many implementation ideas.

Finally, I’d like to thank the many people who have helped me understand, prioritize, implement, and navigate the complex topic of database security and audit, including George Baklarz, Moshe Barr, Roy Barr, Rodrigo Bisbal, Heather Brightman, Nir Carmel, Mike Castricone,

Stephen Chaung, Curt Cotner, Peggy Fieland, Gilad Finkelstein, Bobbi Fox, Guss Frasier, Guy Galil, Jerrilyn Glanville, Richard Gornitsky, Yaffi Gruzman, Evan Hochstein, Memy Ish-Shalom, Nate Kalowski, Dario Kramer, Kai Lee, Mike Lee-Lun, Michael MacDonald, Art Manwelyan, Jack Martin, Charles McClain, Ram Metser, Ola Meyer, Bruce Moulton, Gary Narayanan, Alex Narinski, Fred Palmer, Themis Papageorge, Jason Patti, Jennifer Peng, Daniel Perlov, Bob Picciano, Harold Piskiel, Jonathan Prial, James Ransome, Leonid Rodniansky, Elliott Rosenblatt, Mojgan Sanayei, Ury Segal, Pat Selinger, Nati Shapira, Mark Shay, Izar Tarandach, David Valovcin, Holly Van Der Linden, and John Young. I would also like to thank Tim Donar, Alan Rose, Theron Shreve, and Stan Wakefield for making this book fun to write.

Contents

Preface	xv
I Getting Started	I
Getting Started	I
1.1 Harden your database environment	6
1.1.1 Hardening an Oracle environment	7
1.1.2 Hardening a SQL Server environment	10
1.1.3 Hardening a DB2 UDB (LUW) environment	13
1.1.4 Hardening a Sybase environment	14
1.1.5 Hardening a MySQL environment	16
1.1.6 Use configuration scanners or audit checklists	17
1.2 Patch your database	20
1.2.1 Track security bulletins	21
1.2.2 Example of a class of vulnerabilities: Buffer overflows	24
1.2.3 Anatomy of buffer overflow vulnerabilities	25
1.3 Audit the database	29
1.4 Define an access policy as the center of your database security and auditing initiative	30
1.5 Resources and Further Reading	31
1.6 Summary	33
1.A C2 Security and C2 Auditing	33
2 Database Security within the General Security Landscape and a Defense-in-Depth Strategy	35
2.1 Defense-in-depth	36
2.2 The security software landscape	38
2.2.1 Authentication, authorization, and administration	38

2.2.2	Firewalls	39
2.2.3	Virtual private networks (VPNs)	39
2.2.4	Intrusion detection and prevention	39
2.2.5	Vulnerability assessment and patch management	40
2.2.6	Security management	40
2.2.7	Antivirus	40
2.2.8	Cutting across categories	41
2.3	Perimeter security, firewalls, intrusion detection, and intrusion prevention	42
2.3.1	Firewalls	42
2.3.2	Intrusion detection systems (IDS)	43
2.3.3	Intrusion prevention systems (IPS)	46
2.4	Securing the core	48
2.5	Application security	49
2.6	Public key infrastructure (PKI)	51
2.7	Vulnerability management	52
2.7.1	Why are there so many vulnerabilities?	53
2.7.2	Vulnerability scanners	54
2.7.3	Monitoring and baselining	55
2.8	Patch management	55
2.9	Incident management	57
2.10	Summary	59
3	The Database as a Networked Server	61
3.1	Leave your database in the core	62
3.2	Understand the network access map for your database environment	63
3.3	Track tools and applications	66
3.4	Remove unnecessary network libraries	71
3.4.1	SQL Server (and Sybase) networking layers	72
3.4.2	DB2 networking layers	75
3.4.3	Oracle networking layers	76
3.4.4	Implementation options: Use TCP/IP only	79
3.5	Use port scanners—so will the hackers	81
3.6	Secure services from known network attacks	84
3.6.1	Anatomy of a vulnerability: SQL Slammer	84
3.6.2	Implementation options: Watch vulnerabilities that can be exploited over the network	86
3.7	Use firewalls	86
3.8	Summary	87
3.A	What is a VPN?	88

3.B	Named Pipes and SMB/CIFS	90
4	Authentication and Password Security	95
4.1	Choose an appropriate authentication option	96
4.1.1	Anatomy of the vulnerability: Weak authentication options	97
4.1.2	Implementation options: Understand what authentication types are available and choose strong authentication	98
4.2	Understand who gets system administration privileges	108
4.3	Choose strong passwords	109
4.3.1	Anatomy of the vulnerability: Guessing and cracking passwords	109
4.3.2	Implementation options: Promote and verify the use of strong passwords	111
4.4	Implement account lockout after failed login attempts	117
4.4.1	Anatomy of a related vulnerability: Possible denial-of-service attack	118
4.4.2	Implementation options for DoS vulnerability: Denying a connection instead of account lockout	119
4.5	Create and enforce password profiles	119
4.6	Use passwords for all database components	120
4.6.1	Anatomy of the vulnerability: Hijacking the Oracle listener	120
4.6.2	Implementation options: Set the listener password	122
4.7	Understand and secure authentication back doors	122
4.8	Summary	123
4.A	A brief account of Kerberos	124
5	Application Security	127
5.1	Reviewing where and how database users and passwords are maintained	128
5.1.1	Anatomy of the vulnerability: Database passwords in application configuration files	129
5.1.2	Implementation options: Knowing and controlling how database logins are used	134
5.2	Obfuscate application code	139
5.2.1	Anatomy of the vulnerability: Source code and psuedo-code	140
5.2.2	Implementation options: Precompilation and obfuscation	146
5.3	Secure the database from SQL injection attacks	148

5.3.1	Anatomy of the vulnerability: Understanding SQL injection	149
5.3.2	Implementation options: Preempt, monitor/alert, and block	157
5.4	Beware of double whammies: Combination of SQL injection and buffer overflow vulnerability	168
5.4.1	Anatomy of the vulnerability: Injecting long strings into procedures with buffer overflow vulnerabilities	168
5.4.2	Implementation options: Patches and best practices	170
5.5	Don't consider eliminating the application server layer	170
5.6	Address packaged application suites	171
5.6.1	Anatomy of the vulnerability: All applications have bugs	172
5.6.2	Implementation options: Patch and monitor	174
5.7	Work toward alignment between the application user model and the database user model	175
5.8	Summary	175
6	Using Granular Access Control	177
6.1	Align user models by communicating application user information	179
6.2	Use row-level security (fine-grained privileges/access control)	185
6.3	Use label security	189
6.4	Integrate with enterprise user repositories for multitiered authentication	193
6.5	Integrate with existing identity management and provisioning solutions	198
6.6	Summary	200
7	Using the Database To Do Too Much	203
7.1	Don't use external procedures	203
7.1.1	Disable Windows extended stored procedures	204
7.1.2	Disable external procedures in Oracle	210
7.1.3	Prefer SQL/PL in DB2 UDB over external runtime environments	213
7.2	Don't make the database a Web server and don't promote stored procedure gateways	214
7.2.1	Mod_plsql	215
7.2.2	Mod_ose	218
7.2.3	Implementation options: Remove modules and/or remove the HTTP server	218

7.3	Don't generate HTML from within your stored procedures	219
7.4	Understand Web services security before exposing Web services endpoints	220
7.4.1	XML Web services for SQL Server 2005	221
7.4.2	DB2 Web services	223
7.4.3	Web services callouts from Oracle	224
7.4.4	Web services security	226
7.5	Summary	227
7.A	Cross-site scripting and cookie poisoning	228
7.B	Web services	230
8	Securing database-to-database communications	233
8.1	Monitor and limit outbound communications	233
8.2	Secure database links and watch for link-based elevated privileges	237
8.3	Protect link usernames and passwords	242
8.4	Monitor usage of database links	243
8.5	Secure replication mechanisms	246
8.5.1	Replication options	247
8.5.2	Secure replication files and folders	249
8.5.3	Secure and monitor replication users and connections	252
8.5.4	Monitor commands that affect replication	254
8.5.5	Monitor other potential leakage of replication information	259
8.6	Map and secure all data sources and sinks	259
8.6.1	Secure and monitor log shipping schemes	262
8.6.2	Secure and monitor mobile databases	262
8.7	Summary	266
9	Trojans	267
9.1	The four types of database Trojans	268
9.2	Baseline calls to stored procedures and take action on divergence	269
9.3	Control creation of and changes to procedures and triggers	270
9.4	Watch for changes to run-as privileges	274
9.4.1	Anatomy of the vulnerability: Oracle's PARSE_AS_USER	274
9.4.2	Implementation options: Monitor all changes to the run-as privileges	274

9.5	Closely monitor developer activity on production environments	274
9.6	Monitor creation of traces and event monitors	278
9.6.1	Anatomy of the vulnerability: Setting up an event monitor or a trace	278
9.6.2	Implementation options: Monitor event/trace creation and/or audit all event monitors and traces	289
9.7	Monitor and audit job creation and scheduling	290
9.8	Be wary of SQL attachments in e-mails	293
9.9	Summary	294
9.A	Windows Trojans	294

10 Encryption 297

10.1	Encrypting data-in-transit	299
10.1.1	Anatomy of the vulnerability: Sniffing data	300
10.1.2	Implementation options for encrypting data-in-transit	306
10.2	Encrypt data-at-rest	316
10.2.1	Anatomy of the vulnerability: Prying SELECTs and file theft	317
10.2.2	Implementation options for encrypting data-at-rest	318
10.2.3	What to consider when selecting an implementation option	321
10.3	Summary	324
10.A	Tapping into a TCP/IP session	324

11 Regulations and Compliance 327

11.1	The alphabet soup of regulations: What does each one mean to you?	328
11.1.1	Health Insurance Portability and Accountability Act of 1996 (HIPAA)	329
11.1.2	Gramm-Leach-Bliley Act of 1999 (GLBA)	332
11.1.3	Sarbanes-Oxley Act (SOX or SarBox)	333
11.1.4	California Senate Bill 1386	334
11.2	Understand business needs and map to technical requirements	335
11.2.1	Use “reverse mappings”	336
11.2.2	Timetable, data, and process mappings	337
11.2.3	Example: SOX and Excel	339
11.3	The role of auditing	340
11.4	The importance of segregation of duties	344

11.5	Implement a sustainable solution	347
11.6	Summary	348
12	Auditing Categories	349
12.1	Audit logon/logoff into the database	349
12.2	Audit sources of database usage	354
12.3	Audit database usage outside normal operating hours	356
12.4	Audit DDL activity	357
12.5	Audit database errors	359
12.6	Audit changes to sources of stored procedures and triggers	362
12.7	Audit changes to privileges, user/login definitions, and other security attributes	364
12.8	Audit creations, changes, and usage of database links and of replication	369
12.9	Audit changes to sensitive data	370
12.10	Audit SELECT statements for privacy sets	372
12.11	Audit any changes made to the definition of what to audit	373
12.12	Summary	374
13	Auditing Architectures	375
13.1	Don't create a false sense of security	375
13.2	Opt for an independent/backup audit trail	376
13.3	Architectures for external audit systems	377
13.4	Archive auditing information	380
13.5	Secure auditing information	382
13.6	Audit the audit system	384
13.7	Sustainable automation and oversight for audit activities	385
13.8	Thinks in terms of a data warehouse	386
13.9	Implement good mining tools and security applications	387
13.10	Support changing audit requirements	388
13.11	Prefer an auditing architecture that is also able to support remediation	390
13.12	Summary	391
13.A	PGP and GPG	391
Index		397



Getting Started

This book is about database security and auditing. By reading it you will learn many methods and techniques that will be helpful in securing, monitoring, and auditing database environments. The book covers diverse topics that include all aspects of database security and auditing, including network security for databases, authentication and authorization issues, links and replication, database Trojans, and more. You will also learn of vulnerabilities and attacks that exist within various database environments or that have been used to attack databases (and that have since been fixed). These will often be explained to an “internals” level. Many sections outline the “anatomy of an attack” before delving into the details of how to combat such an attack. Equally important, you will learn about the database auditing landscape—both from a business and regulatory requirements perspective as well as from a technical implementation perspective.

This book is written in a way that will be useful to you—the database administrator and/or security administrator—regardless of the precise database vendor (or vendors) that you are using within your organization. This is not to say that the book is theoretical. It is a practical handbook that describes issues you should address when implementing database security and auditing. As such, it has many examples that pertain to Oracle, SQL Server, DB2, Sybase, and sometimes even MySQL. However, because detailing every single example for every database platform would have meant a 2,000-page book, many of the examples are given for a single database or a couple of them. The good news is that all techniques (or almost all of them) are relevant to all database platforms, and I urge you to read through all sections even if the example code snippets are taken from a database environment that you are not running. In all of these cases, it will be easy for you to identify the equivalent setting or procedure within your own environment.

More important, many of the techniques you will see in this book will never be described in a manual or a book that is devoted to a certain database product. As you'll learn throughout this book, good database security cannot always be implemented solely within the database, and many of the most serious security issues that you may face as the database owner (or the server owner) have to do with the way applications use a database and the way various interacting systems are configured. Addressing these complex issues must take into account more than just the database, and focusing on capabilities that are provided only by the database vendor is not always enough.

At this point you may be asking yourself a few questions:

- Doesn't the database have many security and auditing features? Isn't a database merely a file system with a set of value-added services such as transaction management and *security*? Isn't my database secure?
- Why now? The database has been part of the IT environment for many years (relational databases for at least 20 years); why should we suddenly be overly concerned with security and auditing?

The answer to the first set of questions is that while such features exist, they are not always used and are not always used correctly. Security issues are often a matter of misconfiguration, and the fact that the database implements a rich security model does not mean that it is being used or that it is being used correctly. If you are like 90% of database administrators or security administrators, you are probably aware that your database has big gaping holes—disasters waiting to happen. In fact, here are some examples that made the headlines (and rest assured that for every incident that makes headlines there are 100 that are kept quiet):

- In early 2000, the online music retailer CD Universe was compromised by a hacker known as "Maxus." The hacker stole credit card numbers from the retailer's database and tried to extort money from the retailer. When his demands were refused, he posted thousands of customers' credit card details to the Internet. (Go to <http://databases.about.com/gi/dynamic/offsite.htm?site=http://www.pc%2Dradio.com/maxus.htm> to see what Maxus' Web site looked like.)
-

- In December 2000, the online retailer Egghead.com announced that its customer database may have been compromised and warned that more than 3.5 million credit card numbers may have been stolen. Egghead.com later announced that the credit cards were not compromised but the investigation cost millions and few customers were willing to continue to do business with the retailer. The company went out of business shortly thereafter.
- In 2001, Bibliofind, a division of Amazon.com that specialized in rare and out-of-print books, was attacked and details for almost 100,000 credit cards were stolen. Even worse, the attackers maintained free access to the database for four months before being discovered! As a result, Bibliofind stopped offering buy/sell services and ended up as a matching service only (i.e., had to forgo a large portion of its revenues).
- In March 2001, the FBI reported that almost 50 bank and retail Web sites were attacked and compromised by Russian and Ukrainian hackers.
- In November 2001, Playboy.com was attacked and credit card information was stolen. In fact, the hackers sent e-mails to customers that displayed the credit card information.
- In the course of 2001, Indiana University was successfully attacked twice and private information, such as social security numbers and addresses, was stolen.
- A study conducted by Evans Data (a market research firm) in 2002 sampled 750 companies and reported that 10% of databases had a security incident in 2001! More than 40% of banking and financial services companies reported “incidents of unauthorized access and data corruption” and 18% of medical/healthcare firms reported similar types of incidents.
- In Oct. 2004 a hacker compromised a database containing sensitive information on more than 1.4 million California residents. The breach occurred on Aug 1 but was not detected until the end of the month. The database in question contained the names, addresses, Social Security numbers, and dates of birth of caregivers and care recipients participating in California’s In-Home Supportive Services (IHSS) program since 2001. The data was being used in a UC Berkeley study of the effect of wages on in-home care and was obtained with authorization from the California Department of Social Services. The hacker had reportedly taken advantage of an unpatched system and

while officials declined to state which vendor's database was the subject of the attack they did report that it was a "commercially available product with a known vulnerability that was exploited."

- In Jan 2005 the following was reported by Security Focus (<http://www.securityfocus.com/news/10271>):

A sophisticated computer hacker had access to servers at wireless giant T-Mobile for at least a year, which he used to monitor U.S. Secret Service e-mail, obtain customers' passwords and Social Security numbers, and download candid photos taken by Sidekick users, including Hollywood celebrities, SecurityFocus has learned... by late July [of 2004] the company had confirmed that the offer was genuine: a hacker had indeed breached their customer database

The answer to the second set of questions—why now?—is a convergence of several factors—almost a "perfect storm." True, the database has been around for a long time, but the following trends are dominating the last few years:

- E-commerce and e-business
- New and wonderful ways to use databases
- Increased awareness among the hacker community
- Widespread regulations that pertain to IT and to security

E-commerce and e-business have changed the way we live. We buy from online retailers, we pay our utility bills using online banking sites, and more. Businesses have optimized their supply chains and use Customer Relationship Management (CRM) software to manage relationships with their clients. In doing so, systems have become much "closer" to each other and much "closer" to the end users. Sure, we use firewalls to secure our networks and we don't connect databases directly to the Internet, but you'll see in Chapter 5 that there is more than one way to skin a cat and that databases are far more exposed than they used to be. Ten years ago the database was accessed by applications that were only available to internal employees. Now it is (indirectly through the application) accessed by anyone who has access to the Web site (i.e., everyone in the world).