CRC Press
Taylor & Francis Group
AN AUERBACH BOOK

HOWTO Secure and Audit Oracle 10g and 11g

Ron Ben Natan

Foreword by Pete Finnigan

HOWTO

Secure and Audit Oracle 10g and 11g

HOWTO
Secure and Audit Oracle 10g and 11g

Ron Ben Natan

Foreword by Pete Finnigan

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
AN AUERBACH BOOK

Auerbach Publications
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-13: 978-1-4200-8412-2 (Hardcover)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Ben-Natan, Ron.
How to secure and audit Oracle 10g and 11g / Ron Ben-Natan.
p. cm.
Includes index.
ISBN 978-1-4200-8412-2 (hardcover : alk. paper)
1. Oracle (Computer file) 2. Computer security. 3. Data protection. 4. Database security. I. Title.

QA76.9.A25B446 2009
005.8--dc22 2009001575

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the Auerbach Web site at
http://www.auerbach-publications.com

Dedication

To my father Danny

Contents

Foreword

In recent years, Oracle security has assumed a whole new meaning for many people in organizations around the world; we have seen the rise of regulatory requirements and worse still a huge rise in the reporting of data loss. While not from an Oracle database, the recent loss of two CDs in the United Kingdom containing all the child benefit details of over 25 million people could not be a more graphic example for people who want their data to remain secure. Securing Oracle databases is more important today than it was many years ago when I started dedicating my business and research life purely to thinking about and providing companies and the community with assistance in securing their data held in Oracle databases.

Companies have also been more widely reporting an increase in internal rather than external attacks to their systems. This is of the highest significance for the security of data held in an Oracle database as in today's world the use of perimeter security is of little value when it comes to securing the data. Unfortunately, most companies have open network architectures and most employees have indirect or direct access to the database (whether intended or not) due to open routing policies and also standard desktop builds. As we have seen, the biggest threat in recent times comes from internal attacks; this could again be intentional or unintentional. Remember, in the world of securing data, the adversary that the Database Administrator (DBA) has to compete with may not be a hacker; he or she may be a fellow employee who has too much access that allows damage and unauthorized changes to data and the database itself to occur, or any other of the myriad of situations that allow people unauthorized access, again intended or not.

Securing an Oracle database and the data held in it should be of utmost importance to all of the people within an organization—from the managers who write the checks to the DBAs, developers, security analysts, users, and owners of the data. Securing data held in an Oracle database is not "rocket science" but it is complex because the Oracle database itself is complex and infinitely configurable and also because the applications, data needs, and use are also infinite and different for each organization. Wow! This sounds like a big problem, doesn't it? If every system, application, configuration, use, people, and so on is different we clearly need best practices to secure Oracle and the data.

I should make a clear distinction here. The task of securing the "data" should be held separately from the task of securing the "Oracle software." Why is this? Well, simply put, following a checklist or "tip sheet" is not good enough. We must ensure that this is done as practitioners of "securing data"; yes, securing data must come first and not securing the software; we must understand the data, understand its use, understand what I call its "flow"; how does the data get into the database, how does it leave the database, and where is it at rest. Only then can we start to secure the data using the tools provided by Oracle.

I am a believer in best practices and ideas—good ideas, not just simply ticking off checklists. I believe that if you understand your data you can secure it. I have helped define best practice and helped many clients secure their data. A methodology should teach the principles and obviously discuss the features and tools that Oracle provides, but it must also teach how to understand the risks to the data. Ron's book is clearly written and focuses on the core technology available from Oracle to secure your data, but, importantly, it discusses why you should secure your data and then provides guidelines as to how you should do it using out-of-the-box tools. This is important as it means the book is useful to any customer of Oracle since Ron uses the techniques and tools provided with the Oracle license (additional cost options are discussed, as well such as Audit vault or Virtual Private Database). This is a practical book that any layman can follow and the core concepts are well explained. Did I mention that Oracle security is complex…? ; well Ron cuts to the quick and covers all the core issues. I like to think an ideal book teaches the reader the "how" and the "why," which they can then apply to all aspects of the security of their data. I think Ron has achieved this. I was particularly impressed with the clear discussions on privileges; this has been my main bugbear with lots of customers; I see lots of sites with excessive privileges, serious privileges, wide-ranging access for lots of people; if we can teach users of data the importance of access to only the data they should access and only at the times they should access it, we would have made massive progress toward secure data. Enjoy this book but most importantly learn from it and use it to secure your data.

Pete Finnigan
Managing Director,
PeteFinnigan.com Limited

Acknowledgments

I would like to thank my wife and kids for tolerating my vanishing acts during nights, weekends, and any other time that should have been spent with them and went instead into writing this book.

I would like to thank the whole crew at Guardium for helping me deal with the complex topics of security and Governance, Risk and Compliance (GRC) in large database environments every single day.

Finally, I would like to thank the many people I have worked with over the years on Oracle security—those that I have met as Guardium customers, and others who I have interacted with in Oracle forums. The many hundreds of such interactions helped me understand what is important from a very real and pragmatic point of view, and not merely from a "tooling" perspective. The list of people is too long to mention here but if you have ever taken the time to sit down with me and explain what you need, I thank you for that.

Ron Ben-Natan

Author

Ron Ben-Natan has more than 20 years of experience developing enterprise applications and security technology for blue-chip companies. Ron is currently the chief technical officer at Guardium, the leader in database security and auditing. Prior to this he has worked for companies such as Merrill Lynch, J.P. Morgan, Intel, and AT&T Bell Laboratories. He is an IBM GOLD consultant with a PhD in computer science. Ron is an expert in distributed application environments, application security, and database security. He has authored 11 technical books including *Implementing Database Security and Auditing*. He speaks frequently at database and security seminars including sessions run by Oracle University.

Chapter 1

Introduction: How This Book Will Help You Be Secure and Compliant

The word Oracle means (from Wikipedia):

> An **oracle** is a person or agency considered to be a source of wise counsel or prophetic opinion; an infallible authority, usually spiritual in nature. It may also be a revealed prediction or precognition of the future, from deities, that is spoken through another object (e.g.: runemal) or life-form (e.g.: augury and auspice). In the ancient world many sites gained a reputation for the dispensing of oracular wisdom: they too became known as "oracles", and the oracular utterances, called khrēsmoi in Greek, were often referred to under the same name — a name derived from the Latin verb ōrāre, to speak.

A pretty good name for a database management system (DBMS). But there is another important bit of history behind the use of the word Oracle to name the database engine. Much before Oracle was a company as we know it today, Larry Ellison and Bob Miner, two of the founders of Software Development Labs (which later became Oracle), were working on a consulting project for the Central Intelligence Agency (the CIA in the United States). The CIA wanted to use a new Structured Query Language (SQL) that IBM had written a white paper about. The code name for the project was Oracle (the CIA saw this as the system to give all answers to all kind of questions intelligence analysts had). Larry Ellison and Bob Miner saw the opportunity to take what they had started as part of this project and market it. So they used that project's code name of Oracle to name their new database engine and later the company.

Today, Oracle is the number one database engine based on any metric and it dominates many geographies and many verticals/industries. Perhaps the vertical that it dominates most is that of military organizations, agencies such as the CIA, National Security Agency (NSA), Federal Bureau of Investigation (FBI), and other organizations where security is of utmost importance. This is part of the company's legacy and it is evident in the product. Maybe this origin is the reason that Oracle has more security-related functions, products, and tools (both built by Oracle as well as available as third-party products) than any other comparable DBMS in the world.

Unfortunately, the fact that these capabilities exist does not mean that they are always well-known and used correctly. In fact most users of the Oracle database, even those who have been working with Oracle for many years, are often familiar with less than 20 percent of the security mechanisms within Oracle. This leads sometimes to insecure Oracle environments and other times to implementation which use the wrong tools—meaning a lot of unnecessary work and solutions which require too much effort to sustain over time. One of the main goals of this book is to review the methods and tools available for securing Oracle so that when you have to implement any security-related requirement (be it access control, audit trails, configuration assessment, encryption, etc.), you know how to navigate the options, which tool to use for which scenario, and how to avoid common pitfalls.

1.1 Why Secure the Data?

Let's start with understanding why you must invest in Oracle security. This sounds like a silly question—but you'll have to answer this question in one form or another once you start asking for budgets. It's quite obvious why you need to secure the data—right? Everything you care about sits in a database (and if you're reading this book most likely a lot of it is in an Oracle database). Whether it is financial company data, data about customers, data about employees, credit card information—it is usually stored in a database. There are many other forms that this data can take—data in documents, data in e-mails, data in IM messages, etc. These do not usually live inside the database and there are other tools and techniques (not covered in this book) for securing these endpoints. These data elements are often permutations of data that was sourced from a database. Almost all information that you and your organization consider to be valuable resides in a database in some form or at some point in time. Obviously, you need to secure these "crown jewels."

So far so good. But now let's start asking slightly different questions. Suppose that you did your analysis and go to your management with a proposal to secure the database that will cost $50,000 (or Euros, or Pounds, or whatever your currency may be) per database. What if your proposal required you to add 10 to your headcount? Will it still be so obvious that you need to secure the data (to that level)? Management will most likely not want to have the data "so secure" at that point. If you map your requirements and request $1000 per database; will it then be approved? That depends on what value this investment provides at a business level and what business problem this investment solves. Security, like any other IT investment needs to be justified and being able to answer the question of "why secure the data" (or the less simplistic variants of this question) is important.

At a very high level, the reasons for securing your data fall into two broad categories—you need to avoid a data breach and you need to remain in compliance with some internal or external set of requirements. Both data breaches and noncompliance can cost an organization dearly. A data breach can damage a company's brand, can lead to loss of customers, and always costs a lot of money—money spent on remediation, compensation, investigations, audits, notification, etc. Noncompliance too can be very costly. Noncompliance leads to fines, can lead to loss of a license to operate the business, can lead to continual expensive audits for many years to come, etc.

The justification for investing in securing Oracle is simple—it is a far lower cost than the cost involved with a data breach or noncompliance.

When you build your business case for elevating the security of your data you may have to justify it with a return on investment (ROI). When you build your business justification you should first list the essential elements that must be performed to achieve an acceptable level of security and compliance. You usually do not need to justify this part with an ROI. Security is in many ways like buying insurance. There is no ROI on buying home insurance. If nothing happens during that year (and normally nothing does) then you get no return on a sizable investment. But if your house burns down—well, then you'll be very happy you bought insurance. Security is similar—if there is an attempt to hack your database and it is foiled, your investment has paid off. What is usually more important than ROI is the total cost of ownership (TCO) of the proposal.

Once you have defined the essential elements that have to be implemented comes the second part—your implementation plan. This is where ROI comes in. Given a set of requirements, there are usually many ways to achieve them. Some may involve tools and others may involve manual work done by staff. At this point, you will have to justify your decisions to implement one method over another—and this is done by showing that one method has a much better ROI than another.

Data Breaches

There are many resources and Web sites that track data breaches. For example, the Privacy Rights Clearinghouse keeps a chronology of data breaches that have been reported that involve data that can be useful to identify thieves—including Social Security numbers, account numbers, and driver's license numbers. This list enumerates more than just database-related breaches—it lists all kinds of data-related breaches (which include also things like stolen laptops, etc.). This list only covers reported incidents—and not all incidents are reported by any stretch. The list covers only incidents reported in the United States. This includes a running total of the number of records that have been compromised. Here too, this number is understated because in many incidents the number of affected records is unknown and therefore not counted. Just to give you an idea of the magnitude of the problem—between 2005 and June 2008 there have been at least 225 million records compromised as part of the reported data breaches in the United States. Another good list is the Data Breach Blog that is maintained by SC magazine.

Here is a small sampling of some known incidents involving database breaches (not necessarily Oracle databases):

- In February 2003 the BBC reported an attack on a database that held credit card accounts where the attacker gained access to more than five million Visa and Mastercard accounts.
- In October 2004 a hacker compromised a database containing sensitive information on more than 1.4 million California residents. The breach occurred on August 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 although 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 August 2005 an Air Force spokesman reported that a hacker tapped into a U.S. military database containing Social Security numbers and other personal information for 33,000 Air Force officers and some enlisted personnel.
- In April 2006 Computerworld reported on a case in which an employee at Progressive Casualty Insurance wrongfully accessed information on foreclosure properties she was interested in buying. There was no hacking involved—just a misuse of insider privileges. When the incident was discovered the company sent out letters informing people that confidential information had been accessed by this employee who was fired. The incident was discovered because a local woman complained about receiving calls from a Progressive agent inquiring about her house being under foreclosure—not because there was any database monitoring or auditing in place.
- In September 2006, the virtual reality game SecondLife reported that one of its databases containing unencrypted user information was breached.
- In October 2006, an official with the Illinois Ballot Integrity Project, a not-for-profit organization dedicated to the correction of election system deficiencies, reported that the organization hacked into the voter database for the 1.35 million voters in the city of Chicago and could have not only stolen private information but also create election problems by modifying status and data.
- In June 2007, eWeek reported that Fidelity National Information Services, an electronic payment processor, fired a database administrator (DBA) after they found that the DBA stole and sold customer data exposing as many as 2.3 million bank and credit card records. The DBA, who worked at the company's Certegy Check Services unit, sold the information to a data broker who in turn sold some of it to direct marketers. These activities led to customers receiving marketing solicitations—which was how the incident was exposed.
- In July 2007, credit card information, names, addresses, and phone numbers were hacked from a Western Union database.
- In August 2007, Monster.com reported that details of over 1.6 million job seekers and information on 146,000 subscribers to usajobs.gov residing in a resume database managed by monster.com has been stolen.
- In September 2007, TD Ameritrade reported that information on 6.3 million customers was stolen from one of its database. The stolen information included names, addresses, and e-mail addresses plus a variety of account activity information. The company reported that it discovered and eliminated unauthorized code. Although it did not provide further details, it is likely that this was done by a privileged user. TD Ameritrade said it discovered the breach after customers said they had received spam offering unsolicited investment advice.
- In January 2008, a hacker broke into a database of the Davidson Companies, a financial services firm. The hacker obtained the names and Social Security numbers of practically all of the company's clients as well as information relating to account numbers and balances.
- In March 2008, Harvard University reported that the database containing summaries of GSAS applicant data had been compromised and that about 10,000 of 2007 applicants' personal information may have been compromised. At least 6000 Social Security numbers were exposed and a compressed 125 MB file containing the stolen data is available through BitTorrent, a peer-to-peer network. The file included server backups of three databases and a note was attached which reads "maybe you don't like it but this is to demonstrate that persons like … (admin of the server) in that they don't know how to secure …".

- In May 2008 Computerworld reported that a professional penetration tester (an ethical hacker) managed to hack his way through to a major FBI crime database within a mere six hours.

Data breaches have, unfortunately, become part of our daily lives. In addition to looking at long lists of incidents, it is instrumental to look at some commonalities. One of the best sources for learning about these is the 2008 Data Breach Investigations Report—a study published by the Verizon Business RISK Team. This report draws from over 500 forensic engagements handled by the Verizon Business Investigative Response team over a period of four years. The report provides statistics on how breaches occur, where they occur most (in terms of verticals), what methods were used, and more. For example, the report lists that most breaches resulted from a combination of events rather than a single event but in almost all cases some form of error contributed to a compromise, whereas less than a quarter of attacks exploited vulnerabilities. This shows how important it is to know how to use the security options correctly. Of the incidents due to vulnerabilities, 90 percent of the vulnerabilities exploited by these attacks had patches available for at least six months prior to the breach (hence, you'll learn how to read Critical Patch Updates in Chapter 2).

The report also finds that nine of ten breaches involved something that is unknown to the owner—the most common one being data that was not known to exist on the compromised system. This is shown in Figure 1.1. The report calls these recurring situations as the "Achilles heel in the data protection efforts of every organization." This is perhaps the most important theme in data breaches—you cannot protect that which you do not know about and you cannot secure what you cannot see. The importance of visibility—visibility into where sensitive data resides, visibility into who or what is accessing sensitive data, visibility into controls—is perhaps the most important element that must exist for a database to be secure. Two other very disturbing facts that the report finds is that most breaches go undetected for a long time and are discovered more often by a third party than the breached organization and that most

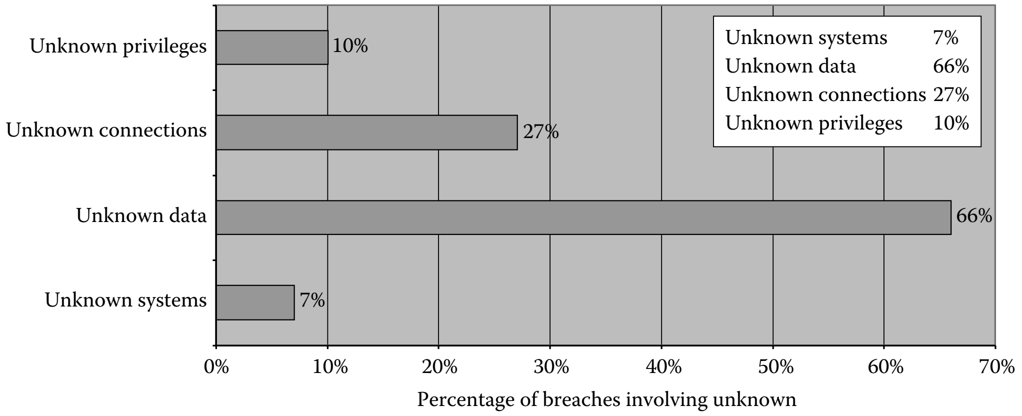

Figure 1.1 Breaches involving unknown factors.

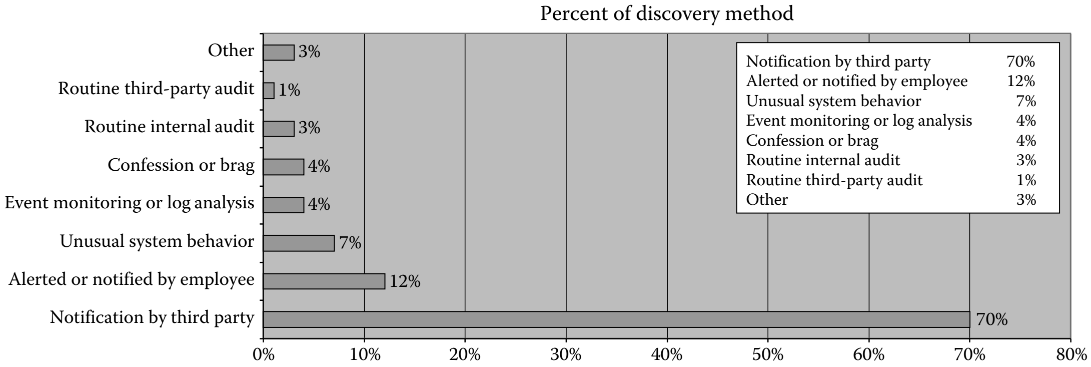

Figure 1.2 Discovery of breaches.

attacks are low to moderate in difficulty—i.e., the attacker does not have to work too hard. Specifically, the report finds that

- 66 percent of attacks involved data the victim did not know was on the system.
- 75 percent of the breaches were discovered by a third party and not the breached organization—as shown in Figure 1.2. The report goes on to present the data shown in Figure 1.3 regarding the time until discovery. This is a very serious finding—it shows that there is a great deficiency in monitoring and auditing. There is a huge difference between a breach that lasts for a day versus a breach that goes on for months, and between a breach that is discovered and handled versus one where the victims (the people who's data is stolen) cannot defend themselves because they don't even know they are victims.
- 83 percent of the attacks were not difficult to perform.
- 87 percent of the attacks could have been avoided through reasonable controls.

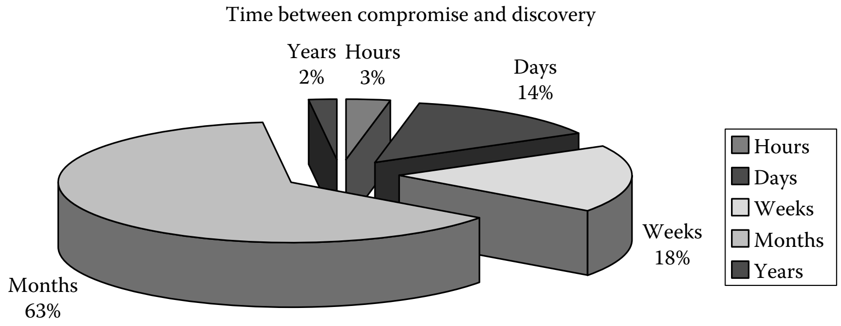

Figure 1.3 Time until discovery.

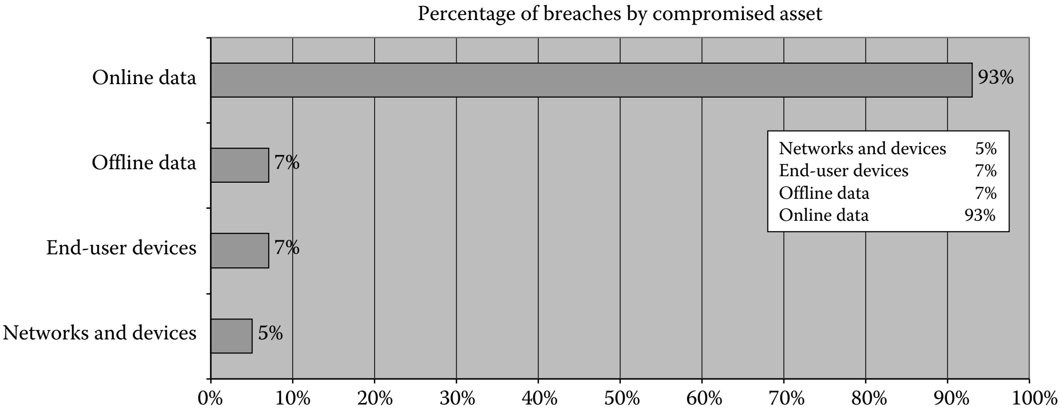

Figure 1.4 Which data is most often targeted?

Finally, if you have any doubt about the importance of database security in the battle against data breaches, the report brings some statistics on the assets that were compromised. Data sits in many places and takes many forms. Data can be in a database but can also reside on USB keys. Data that resides in a database also exists in backups and taken offline. As Figure 1.4 shows, compromises to online data repositories occurred more than five times more often than all other asset classes combined!

Compliance

It would be wonderful if we invested in security because we were so disturbed by the many data breaches we've seen and because we always want to do the right thing—but the truth is that we usually invest in security because we're told to do so. Most of the investment in Oracle security is made because of the need to comply with a regulation or a requirement.

There are two kinds of compliance requirements—internal and external. Internal requirements are policies that are defined within the company. They include policies set by the information security or internal audit groups and they are usually derived from industry best practices, from a regulation, or from preparation for an external audit. External requirements derive from a regulator or from external auditors. There are numerous examples of regulations that affect database security—including Sarbanes Oxley (and its derivatives), the Payment Card Industry Data Security Standard (PCI DSS), various data privacy laws, and many others. Some of these regulations are industry-specific, some are national (i.e., affect only companies operating in a certain country), and others relevant to companies of a certain size. Most of these regulations do not directly set requirements related to database security—they need to be interpreted and mapped to IT terms and these interpretations determine what is required to be in compliance. Luckily, most of these regulations have been around long enough to have a standard interpretation and one that is consistent with requirements set by industry best practices.

Compliance is a very important driver—especially when it comes from an external source. If you need to comply with a certain regulation, it is very hard to shut down a project for lack of sample content of HOWTO Secure and Audit Oracle 10g and 11g