Database Security and Auditing: Leading Practices

Rob Barnes
Director, Enterprise Auditing Solutions
Application Security, Inc.
Getting to Know Database Threats and Vulnerabilities
Key Objectives

- Understand threats and vulnerabilities in common database environments.
  - Oracle
  - DB2/UDB
  - Microsoft SQL Server
  - MySQL
  - Sybase
What Do The Numbers Tell Us?

- 84% of companies feel database security is adequate.
- 56% of the same companies experienced a breach in the last 12 months.
- 73% of companies predict database attacks will increase.

Enterprises are Plagued by a False Sense of Security

Don’t wonder if your data is protected, Know it’s protected.

Source: Enterprise Strategy Group
Survey of 200 Enterprise Organizations
The Threats to Enterprise Data Continue to Rise

- The database security landscape has changed:
  - Attacks are targeting the database where records can be harvested in bulk on a global scale
  - Perimeter security measures are necessary but not sufficient
Databases Account For 92% of Stolen Records!

- 428 Million records compromised 2008-2009
- Hundreds of incidents, Dozens of industries

Source of Records Lost in 2009
- 92% Database
- 4% Laptop
- 3% Mail Server
- 1% FTP Server

Source: Verizon

Cost Per Exposed Record
- 2005: $138
- 2006: $181
- 2007: $197
- 2008: $202
- 2009: $204

Source: Ponemon Research

www.appsecinc.com
More Numbers - Overview

Who is behind data breaches?
- 62% external sources
- 46% insiders
- 10% business partners
- 18% multiple parties

What’s involved in a data breach?
- 40% hacking and intrusion
- 38% incorporated malicious code
- 48% abuse of privileges
- 15% physical threats
- 2% significant error
- 43% multiple vectors

Source: Verizon 2010 Data Breach Investigations Report
Organizations Aren’t Protecting Themselves

- 96% of breaches in 2009 were avoidable through simple controls
- 79% of organizations with credit card data breaches in 2009 failed their last PCI audit
- 41% of successful attacks in 2009 involved script kiddie skills or less.
  - 85% “not considered highly difficult”
- 48% of attacks were insiders abusing privileges
  - 70% were executed by non-technical employees

<table>
<thead>
<tr>
<th>In 2009 – The % of Records Stolen by Industry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Financial Services</td>
</tr>
<tr>
<td>Others</td>
</tr>
</tbody>
</table>

Source: Verizon 2010 Data Breach Investigation Report
Database Breach Settlements

- Heartland settles with Visa - $60M
- Heartland settles with AMEX - $3.6M
  - Heartland CEO only set aside $12.6M
  - 5 Issuing banks seeking class action lawsuits against Heartland Bank and Key Bank
- Countrywide settles 35 lawsuits – Could cost $17M to $51M (credit monitoring alone)
Databases and Data Breaches – More Stats

- Databases are the central repositories for the most confidential data
  - Statistics show more sensitive data is stored in databases than file servers, web and email servers, and endpoints such as PCs and laptops.
  - 43% of enterprise databases contain sensitive data
- The threat landscape has changed:
  - Attackers are well-funded, organized, and extremely sophisticated
  - Attackers have been successfully harvesting data en masse
  - Organizations increasingly grant access to data to: employees, contractors, suppliers, partners, customers and 3rd party (outsourcing) vendors

Source: ESG 2009 Database Security Controls Survey
But with the Threat Landscape Changed…

- We still fail to include the database in our Audit and Security Programs
  - Still mainly focus at the application level
  - Separation of duties at the application level – reviewing toxic combinations

- If we do include it, we run into challenges:
  - DBA’s are reluctant to give access
  - Only audit a small sample set of in-scope at times
  - Don’t always understand what audit controls to review at the database
Emerging Database Threats

- Sophisticated attacks that exploit un-patched vulnerabilities
- Double or triple encrypted SQL-injection attacks that render web-application firewalls virtually useless
- Insider attacks
- Insider mistakes
- Advanced identity theft via database rootkits
- Increasingly sophisticated social engineering leading to full-blown database disclosures
- Weak or non-existent audit controls
- Powerful self-propagating attacks distributed via “infection kits” on legitimate websites and other creative means

The Insider Threat
Common Database Threats

**Database Vulnerabilities**
- Default accounts and passwords
- Easily guessed passwords
- Missing Patches
- Misconfigurations
- Excessive Privileges

**External Threats**
- Web application attacks (SQL injection)
- Insider mistakes
- Weak or non-existent audit controls
- Social engineering
Database Vulnerabilities

- Default & Weak Passwords
- Patchable Vulnerabilities
- Misconfigurations & Excessive Privileges
Database Vulnerabilities: Default Accounts

Databases have their own user accounts and passwords

User: system / Password: manager
User: sys / Password: change_on_install
User: dbsnmp / Password: dbsnmp

User: SA / Password: null

User: db2admin / Password: db2admin
User: db2as / Password: ibmdb2

User: root / Password: null
User: admin / Password: admin
User: myusername / Password: mypassword

Proper Safeguards are Needed because:
Not all databases have Account Lockout
Database Login activity is seldom monitored
Scripts and Tools for exploiting weak passwords are widely available
Database Vulnerabilities: Missing Patches

Databases have their own Privilege Escalation, DoS’s & Buffer Overflows

**Privilege Escalation**
Become a DBA or equivalent privileged user

**Denial of Service Attacks**
Result in the database crashing or failing to respond to connect requests or SQL Queries

**Buffer Overflow Attacks**
Result in an unauthorized user causing the application to perform an action the application was not intended to perform
Database Vulnerabilities: Misconfigurations

Misconfigurations can make databases vulnerable

- External Procedure Service
  - Privilege to grant Java permissions
  - Default HTTP Applications
  - Privilege to Execute UTL_FILE

- Permissions granted on xp_cmdshell

- CREATE_NOT_FENCED privilege granted (allows logins to create SPs)

- Permissions on User Table (mysql.user)

Simple changes can make a big difference:
Remember? 96% of breaches were avoidable through simple controls
Remember? 85% of breaches were “not considered highly difficult”
Database “Insider Threat”: Excessive Privileges

The CISO of one of the largest banks in the world says…

“I define insiders in three categories

1. Authorized and Intelligent
   - use IT resources appropriately

2. Authorized and “stupid”
   - make mistakes that may appear as malicious or fraudulent.

3. Unauthorized and Malicious
   - mask either their identity or their behavior or both!

The first two categories I can identify and track with identity management systems – the later, I can not!!”
Recap: Emerging Database Threats

- Sophisticated attacks that exploit un-patched vulnerabilities
- Double or triple encrypted SQL-injection attacks that render web-application firewalls virtually useless
- Insider attacks
- Insider mistakes
- Advanced identity theft via database rootkits
- Increasingly sophisticated social engineering leading to full-blown database disclosures
- Weak or non-existent audit controls
- Powerful self-propagating attacks distributed via “infection kits” on legitimate websites and other creative means
Database Security
Leading Practice
Key Objectives

- Understand the database security lifecycle.
- Understand how to secure common database environments.
Leading Practices: Stopping the Attack

- **Key Findings**
  - Half of the insiders had authorized access to the systems/networks at the time of the incidents.
  - Over half of the insiders used relatively sophisticated tools or methods for their illicit activities.
  - Over half of the insiders exploited systemic vulnerabilities in applications, processes, and/or procedures.
Leading Practices: Stopping the Attack

**Implications**

- Apply the principle of “least privilege” giving users only the access they need to do their jobs – separation of duties
- Eliminate weak or default passwords on systems
- Maintain role-based access controls and disable access after an employee changes positions within a company
- Formal policies and procedures for disabling access upon an employee’s termination or resignation should be established and followed
- Procedural and technical controls should be established for system administrator functions
- Periodic account audits should be conducted to check for unneeded or unauthorized accounts, including: Remote access accounts, login accounts, DBA accounts, application, customer, and company accounts
How to Protect Against Attacks

- Start with a Secure Configuration
- Stay Patched
  - Stay on top of all the security alerts and bulletins
- Implement the Principal of Least Privilege
  - Review User Rights to ensure all access is appropriate
- Defense in Depth / Multiple Levels of Security
  - Regularly scan your databases for vulnerabilities
    - Fix the problems reported!
  - Implement database activity monitoring…
  - …and database intrusion detection
    - Especially if you can’t stay patched!
  - Encryption of data-in-motion / data-at-rest
How to Protect Against Attacks

- Set a good password policy:
  - Use strong passwords or passphrases.

- Keep up to date with security patches:
  - Try to install patches as fast as you can. Database vulnerabilities are serious and sometimes a database server can be easily compromised with just a simple query.
  - Always test patches for some time on non-production databases.
How to Protect Against Attacks

- Protect access to the database server:
  - Allow connections only from trusted hosts and block non used ports and outbound connections. Establish exceptions for special instances like replication, linked databases, etc.

- Disable all non used functionality:
  - Excess functionality can lead to vulnerabilities

- Use selective encryption:
  - At network level: use SSL, database proprietary protocols.
  - At file level for backups, laptops, etc.
# Data Security Life Cycle

<table>
<thead>
<tr>
<th>Lifecycle Component</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discover</td>
<td>Produce a database or asset inventory</td>
</tr>
<tr>
<td>Classify</td>
<td>Finds sensitive data to determine business value of systems and associated regulatory requirements</td>
</tr>
<tr>
<td>Assess</td>
<td>Scan databases for vulnerabilities, misconfigurations / configuration changes, and user entitlements</td>
</tr>
<tr>
<td>Prioritize</td>
<td>Combine info from classify and assess phases to determine what to fix, what to mitigate through compensating controls (monitoring), and in what order to do the work</td>
</tr>
<tr>
<td>Fix</td>
<td>Create and run fix scripts, apply patches, create monitoring policies to implement compensating controls</td>
</tr>
<tr>
<td>Monitor</td>
<td>Audit privileged access and access to sensitive data. Monitor for exploits and suspicious or unusual behavior</td>
</tr>
</tbody>
</table>
Bringing it All Together

Entities | Relationships | Data

Discover
Who, What, Where, and How They are Involved

Classify
Values are based on interrelationships and level of criticality

Monitor
Alert for any deviation from the established baseline

Assess
Pinpoint weakest link in the chain

Fix
Viable remediation and compensating controls are applied along with new baselines

Prioritize
Remediation efforts are ranked based upon their value and criticality
Database Auditing
Leading Practice
Key Objectives

- Preparing for the Audit
- Understand how to audit *Users and User Entitlements*.
- Understand how to audit for baseline configuration, password and patch-level settings.
- Understand *Continuous Compliance/Continuous Assurance* (Activity Monitoring and Compensating Controls)
Relational Database Environments

Most business applications use database management systems including:

- Oracle
- DB2 LUW
- MS SQL Server
- Sybase
- MySQL

Audit and control for each type are similar but require an understanding of the architecture and technology.

Source: John Tannahill, J. Tannahill & Associates
Security as Part of the Audit

- A database environment is a data repository or data store for:
  - Operational Data (Financial, Personal, etc.)
  - Data Warehouse Data
  - Security Data

- You need to understand security requirements for data in terms of:
  - Confidentiality
  - Integrity
  - Availability

- Need to understand the compliance and regulatory requirements based on the business environment.

Source: John Tannahill, J. Tannahill & Associates
Security as Part of the Audit

- Database security mechanisms include:
  - Identification and authentication mechanisms
  - Access controls
  - Audit trail mechanisms

- Network security and host operating system security are required in addition to database security

- Database systems are “TCP/IP services” and can be compromised even when the operating system is “hardened”
  - Database compromise can result in an operating system compromise

Source: John Tannahill, J. Tannahill & Associates
Security Architecture and Design

- Need to understand network and application system architecture and design
- Need to identify and understand database connections in relation to the following access paths to the database environment:
  - On-line transaction processing
  - Batch processing
  - Business user ad-hoc access
  - Database administration
  - Developer and application support access

Source: John Tannahill, J. Tannahill & Associates
Database Audit – Initial Planning

- Understand application system and network infrastructure
- Identify database administrators
- Identify database environments and versions
  - Operating system hosts
  - Database configuration files/documentation
  - Database schemas
  - Security design
- SQL queries to obtain database security information

Source: John Tannahill, J. Tannahill & Associates
OS Considerations for Database

- OS accounts and related password controls
- Privileged OS accounts
- Group memberships
  - Unix groups
  - Windows 2000/2003 Administrators Group
- Owner/Service accounts for Database Management System software
- Program and file protection
  - OS directory and file permissions

Source: John Tannahill, J. Tannahill & Associates
Database Vulnerability Testing

- Vulnerability tests
  - OS probes for known vulnerabilities
  - Identify vulnerable TCP/IP connections
  - Database probes for known weaknesses and vulnerabilities
  - Specific tests for default accounts and weak passwords

- Manual Scripts versus Automation
  - AppDetectivePro

Source: John Tannahill, J. Tannahill & Associates
Database Security Standards

- Secure configuration (hardening)
- Secure patch management
- Example standards include:
  - Center for Internet Security (cisecurity.org) Benchmarks
    - Oracle 8i/9i/10g/11g
    - SQL Server 2000/2005
    - DB2 (Windows/Unix Hosts)
    - Sybase ASE
  - Database Security (STIG)
    - AppDetectivePro – Only automated solution with complete database STIG controls, standards and test work plan.

Source: John Tannahill, J. Tannahill & Associates
Major Components of Database Auditing

1. **Access & Authentication Auditing**
   Who accessed which systems, when, and how

2. **User & Administrator Auditing**
   What activities were performed in the database by both users and administrators

3. **Security Activity Monitoring**
   Identify and flag any suspicious, unusual or abnormal access to sensitive data or critical systems

4. **Vulnerability & Threat Auditing**
   Detect vulnerabilities in the database, then monitor for users attempting to exploit them

5. **Change Auditing**
   Establish a baseline policy for database; configuration, schema, users, privileges and structure, then track deviations from that baseline
Database Users and Passwords – Common Issues

- Use of generic and shared user accounts
- Use of OS authentication
  - Problem if OS password is compromised
- Default or week passwords
- Lack of password controls
  - No requirement to force password changes
  - Minimum password length not used
  - Application connections to the database

Source: John Tannahill, J. Tannahill & Associates
Database Objects – Common Issues

- Ownership of database schemas and objects
- Control over Administrative Users
  - DBAs and Developers
- System privileges and authorities
  - Segregation of Duties (SOD)
- Object privileges required for production environment
- Public Access
  - Should be limited to SELECT
- Default access provided to PUBLIC

Source: John Tannahill, J. Tannahill & Associates
Database Objects – Common Issues

Check for object and system permissions:
- Check views, stored procedures, tables, etc. permissions. Check file, folder, registry, etc. permissions. Changes on permissions could mean a compromise or mis-configuration.

Look for new database installations:
- Third party products can install database servers and new installed servers could be installed with blank or weak passwords, un-patched, mis-configured, etc. Detect new database installations and secure or remove them.

Search for users with DBA privileges:
- This helps to detect intrusions, elevation of privileges, etc.
Database Users – Segregation of Duties Audit

- Segregation of Duties Audit
  - Who are the users?
  - What are their *effective* privileges?
  - Who has access to sensitive data?
  - How did they get that access?
  - How many databases do I have???

*Segregation of Duties is the core* to most compliance and regulatory mandates!
Database Change Control

- New Database Instances
- Audit database configuration and settings
  - If security configurations or settings are changed for instance by a system upgrade, patch, etc. your databases could be open to attack. If they change and there wasn't a system upgrade then it could mean a compromise.

- Check database system objects against changes
  - If you detect a change in a system object and you haven't applied a fix or upgrade to your database server it could mean that a rootkit is present.
Database Audit Trail Issues

- Application versus database audit trail issues
- Audit trail configuration
- Audit trail requirements
  - System Access
    - Logins – Success / Fail
    - Account / Role / Permissions Changes
  - Data Access
    - SELECT – Success / Fail
  - Data Change
    - INSERT, UPDATE, DELETE
  - Schema / Object Changes
    - CREATE / ALTER / DROP
  - Privileged User Activity
    - All
- Monitoring, analysis, and follow-up process
- Database Activity Monitoring

Source: John Tannahill, J. Tannahill & Associates
Advantages of Off-database Auditing

- Native database auditing has its disadvantages
  - Must be enabled and configured on each system individually
  - Separation of controls?
    - Can be solved with audit management tools (aka Audit Vault)
- Native auditing
  - Can be disabled or deleted by attacker in the database
  - Most databases have NO auditing configured
Compensating Controls - Monitoring

- Complete these steps first. Then monitor!
  - “Outside in” and Inside out” scan of all database applications to assess
    - Security strength
    - Database vulnerabilities
    - Application discovery and inventory
  - Fix security holes and misconfigurations
  - Develop policies based on results from scan to identify:
    - Database vulnerability
    - Roles and responsibilities functionality to segregate users
    - Compliance risk factors
  - Auditing
    - Comprehensive reporting

- Real-Time Monitoring
  - Defend against misuse, fraud, and abuse from internal and external users
  - Monitor all user activity and system changes (DDL, DML, DCL)
  - Tune detection parameters to capture events while bypassing false positives
Wrap Up

- Databases are under attack.
- Security and Audit work together.
- The security lifecycle is key to developing a programmatic approach to security and audit.
- Focusing on a few key areas will help you simplify the process and reduce the workload.
Questions?

- Vulnerabilities?
- Locking down the database?

Email our security experts at: asktheexpert@appsecinc.com