OVAL and Database Vulnerability Assessment

Who are we and and why are we talking?

- Who are we?
  - **Charles McClain,** Advisory Software Engineer, IBM Infosphere Guardium
  - **Louis Lam,** Database Manager, IBM Infosphere Guardium
  - Responsible for IBM Infosphere Guardium database vulnerability assessment content
- Why are we talking?
  - Guardium is an industry leader in database activity monitoring and database vulnerability assessment
  - Guardium application is 10 years old, about to release v9.0
  - IBM acquired Guardium in 2010
  - IBM is a database vendor – DB2, Informix, and Netezza
  - In the coming release of our application, we've translated our proprietary database assessment results into an SCAP-compliant results stream
  - We're working with the OVAL/MITRE people to identify current OVAL issues relating to database vulnerability assessment, and to suggest possible improvements

Why *shouldn't* we be talking?

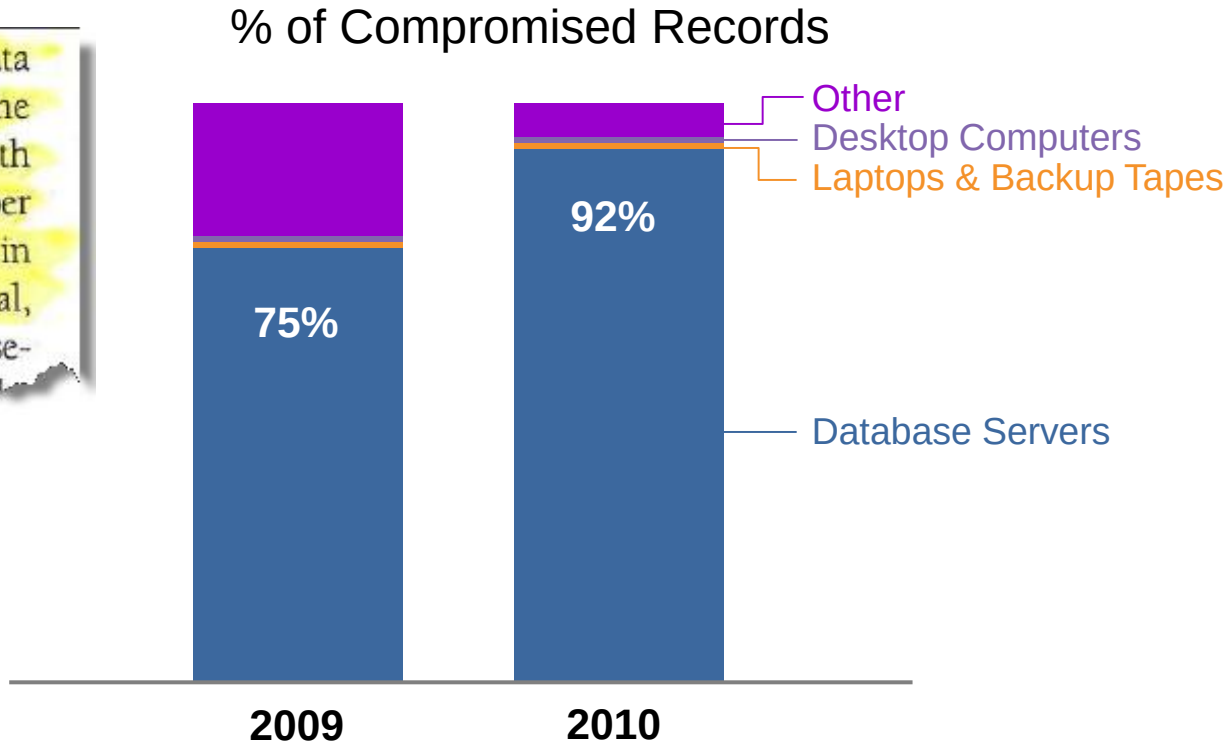
- We are *not* SCAP or OVAL experts
- You probably know SCAP and OVAL much better than we do
- We don't have all the answers

Why should you care about database vulnerability assessment?

Database Servers are the Primary Source of Breached Data

For example, the 2010 Verizon data breach report places databases as the top type of compromised asset by both the number of breaches and the number of records stolen. Yet our investments in protecting database systems is minimal, at best. When it comes to database se-

InformationWeek
"Epic Fail"
10/11/2010



Sources: Verizon Business Data Breach Investigations Report 2009, 2010
www.verizonbusiness.com/resources/reports/rp_2010-data-breach-report_en_xg.pdf

Although much angst and security funding is given to ***offline data, mobile devices, and end-user systems,*** *these assets* ***are simply not a major point of compromise****."*

Why should you care about database vulnerability assessment?

Hackers obtained personal information on 70 million subscribers.

April 2011: *Malicious outsiders stole name, address (city, state, zip), country, email address, birth date, PlayStation Network/Qriocity password and login, and handle/PSN online ID, and possibly credit card numbers from 70 million Sony PlayStation users.*

SQL injection is fast becoming one of the biggest and most high profile web security threats.

April 2011*: A mass SQL injection attack that initially compromised 28,000 websites shows no sign of slowing down. Known as LizaMoon, this malicious code is after anything stored in a database.*

Unprotected test data sent to and used by test/ development teams as well as third-party consultants.

February 2009: *An FAA server used for application development & testing was breached, exposing the personally identifiable information of 45,000+ employees.*

Hundreds of thousands of secret reports regarding US wars in Iraq and Afghanistan published on WikiLeaks.

December 2010: *A private in the US military, downloaded top secret military documents and passed them to journalist for publication. This puts US national security at risk as well as the lives of those named in reports.*

Why should you care about database vulnerability assessment?

- 73% of security professionals say the volume of database attacks will increase
- $7.2M USD is the average cost of a data breach
- 88% of organizations surveyed had at least one data breach

What's different about database vulnerability assessment?

- 90% of database breaches perpetrated by insiders, not hackers
- Most of these “insider” breaches involve misuse of database privileges – system privileges, role privileges, and/or object privileges
- Privileges maintained in the database catalog
- Catalog format unique to each DBMS
- Catalog may be centralized or decentralized
- Pass/fail for a given test may depend on complex SQL query involving 100's or 1000's of databases, users, objects, and privileges
- A simple pass/fail isn't enough; failure requires reporting of every combination of database, user, object, and privilege that caused failure

Why not just use <sql57_test>?

- Connection String Issues
  - <connection_string> exposes credentials
  - <connection_string> exposes database identification
  - <connection_string> associated with each test
  - No link between <connection_string> and asset identification
  - Need for <datasource> entity
  - Execution location
- <engine> enumeration updates
  - Database types
  - Distinction between DB2 products
- Multi-database queries (Decentralized catalog)
- Support for test categories, identification of executable
- Support for failure detail
- Exceptions from tests
- Lack of tool support for SQL tests

<connection_string> exposes credentials

```
<objects>
  <ind-def:sql57_object xmlns="http://oval.MITRE.org/XMLSchema/oval-definitions-5#independent"
    id="oval:com.ibm.guardium.va:obj:2134" version="1">
    <ind-def:engine>oracle</ind-def:engine>
    <ind-def:version>11.1</ind-def:version>
    <ind-def:connection_string>
      jdbc:oracle:thin:@server1.guard.swg.usma.ibm.com:1521:ORCL,SYSTEM,MANAGER
    </ind-def:connection_string>
    <ind-def:sql>select ENAME from EMP where EMPNO=7369</ind-def:sql>
  </ind-def:sql57_object>
</objects>
```

<connection_string> exposes database identification

```
<objects>
  <ind-def:sql57_object xmlns="http://oval.MITRE.org/XMLSchema/oval-definitions-5#independent"
    id="oval:com.ibm.guardium.va:obj:2134" version="1">
    <ind-def:engine>oracle</ind-def:engine>
    <ind-def:version>11.1</ind-def:version>
    <ind-def:connection_string>
      jdbc:oracle:thin:@server1.guard.swg.usma.ibm.com:1521:ORCL,SYSTEM,MANAGER
    </ind-def:connection_string>
    <ind-def:sql>select ENAME from EMP where EMPNO=7369</ind-def:sql>
  </ind-def:sql57_object>
</objects>
```

No link between <connection_string> and asset identification

```
 <core:relationship subject="database_1" type="arfvocab:servedBy">
  <core:ref>service_1</core:ref>
 </core:relationship>
.
.
<assets>
 <asset id="database_1">
  <ai:database>
   <ai:instance-name>ORCL</ai:instance-name>
  </ai:database>
 </asset>
 <asset id="service_1">
  <ai:service>
   <ai:hostname>server1.guard.swg.usma.ibm.com</ai:hostname>
   <ai:port>1521</ai:port>
   <ai:protocol>TCP</ai:protocol>
  </ai:service>
 </asset>
</assets>
.
 <connection_string>
  jdbc:oracle:thin:@server2.guard.swg.usma.ibm.com:1522:INST1,SYSTEM,MANAGER
 </connection_string>
```

Connection string issues – Possible solution

- Remove connection string from <sql57_object> entirely, put connection responsibility on tool
- Still need some way to tell tool what database to connect to

Encapsulated datasource identification at asset level

```
<assets>
  <asset id="datasource_1">
    <ai:datasource>
      <ai:datasource-id>173689</ai:datasource-id>
      <ai:datasource-name>Charlie's Oracle 11g datasource</datasource-name>
    </ai:datasource>
  </asset>
```

<engine> enumeration updates

- Netezza, Teradata missing
- DB2 LUW, DB2 z/OS different products
  - Different codebases
  - Tests that work on DB2 LUW don't work on DB2 z/OS, and vice versa

Proposed solution

```
<xsd:complexType name="EntityObjectEngineType">
   <xsd:simpleContent>
     <xsd:restriction base="oval-def:EntityStateStringType">
       .
       <xsd:enumeration value="db2"/>
       <xsd:enumeration value="db2-luw"/>
       <xsd:enumeration value="db2-zos"/>
       .
       <xsd:enumeration value="netezza"/>
       <xsd:enumeration value="teradata"/>
       .
       <xsd:enumeration value=""/>
     </xsd:restriction>
   </xsd:simpleContent>
 </xsd:complexType>
```

Multi-database queries

- DBMS catalog types
  - Centralized (Oracle)
  - Decentralized (Sybase, SQL Server)
- Privilege tests must examine all privileges in entire catalog
- For decentralized catalog DBMS's, that involves:
  - Executing a query to discover databases
  - Executing the test query against each discovered database
- Pass/fail is based on results of all queries

Possible solutions

- OVAL-centric
  - Add a behavior to <sql57_object> (<behavior decentralized=”true/false”/>)
  - Add a <catalog_sql> entity to the <sql57_object>
  - If <behavior decentralized=”true”>, tools must first execute <catalog-sql> to discover databases, then iteratively execute <sql> against each discovered database
- Tool-centric
  - Put the burden on tools to know whether a DBMS is centralized/decentralized, based on DBMS identification in the <engine> entity
  - Put the burden on tools to know the form of the database discovery query for each decentralized DBMS
  - Put the burden on tools to know that, for decentralized DBMS's, the query in <sql> must be executed iteratively for each discovered database
- In either case, test must fail if <sql> query returns any rows for any discovered database

Support for test categories, identification of executable

- SQL tests require a supporting code mechanism in the tool
  - Custom code mechanism for that test only
  - Code mechanism that supports a class of tests (e.g., CVE tests)
  - User-defined tests
- <engine> and <sql> entities don't always provide enough information to determine which code mechanism to choose for a test
- Need a way to tell the tool which category a test belongs to – i.e., which code mechanism to use

Possible solutions

- OVAL-centric
  - Add a <category> enumeration to <sql57_object>, limiting the categorization of tests to enumerated categories
- Tool-centric
  - Add an integer <category> entity to <sql57_object>
  - Leave the use of <category> to the tool

Complex queries

- Some database vulnerability assessment tests require very complex queries, including, but not limited to:
  - Multi-table queries involving complex JOIN and filter conditions
  - UNION SELECT queries involving a dozen or more tables
  - Subqueries, both simple and correlated
  - Anonymous block execution
- These queries can run to 100's of lines
- These queries can and must use DBMS-specific features

Solution

- Put the burden on tools to support any and all queries supported by the DBMS
- Include clear documentation to that effect in the OVAL specification

Support for failure detail

- For complex privilege tests, the <sql> query that determines pass/fail is typically something like “select count(*) where [condition] is true”
- Test should fail if count > 0
- Upon failure, tests must provide failure detail – each combination of user, object, and privilege that caused the test to fail
- This detail can run into 100's or 1,000's of lines

Possible solutions

- Add a <detail_sql> to <sql57_object>, to be conditionally executed if and when the test fails
- Stipulate that the original <sql> query is the detail query, and that pass/fail is determined by <sql57_state>
- PERFORMANCE IMPLICATIONS?

Exceptions from tests

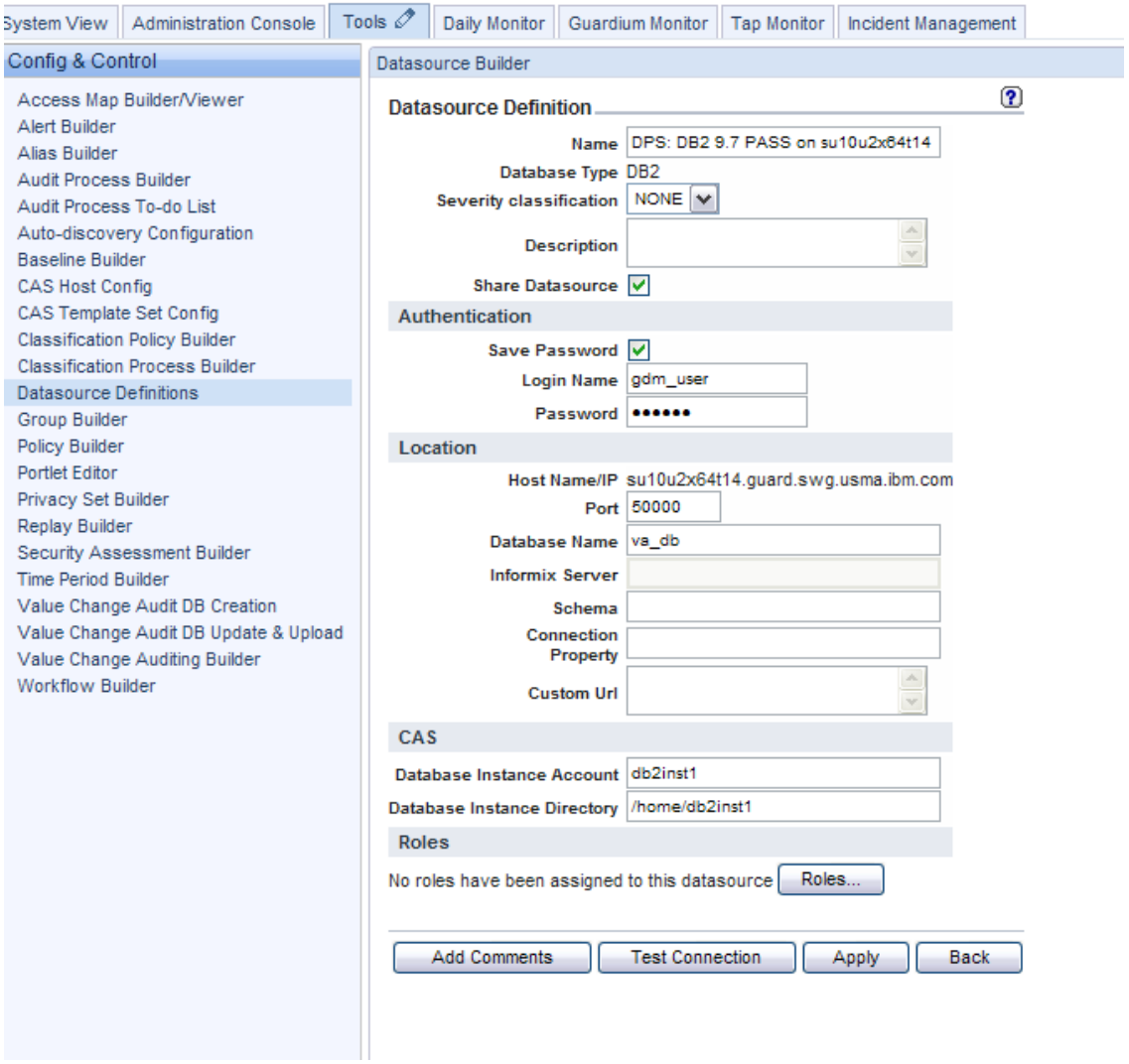
- In the real world – particularly for tool vendors whose products are used by many customers – tests must be able to observe *exceptions*
- EXAMPLE:
  - CIS v2.01, Item 9.07: “Only DBA access to SYS.USER$”
  - Certain Oracle components (e.g., APEX and XDB) install administrative users with such privileges, and **will not function** if you revoke those privileges
  - Impossible to know in advance whether a customer will be using APEX and/or XDB
- In other cases, customers have simply configured their environment in a particular way and don't want to change it – and don't want to exclude the test

Possible solutions

- OVAL-centric
  - Add an <exception> entity to <sql57_object> (PROBLEM: Exposes user information in clear text)
- Tool-centric
  - Put the burden of observing exceptions on the tool (PROBLEM: Test passes when, by all visible evidence, it should fail)

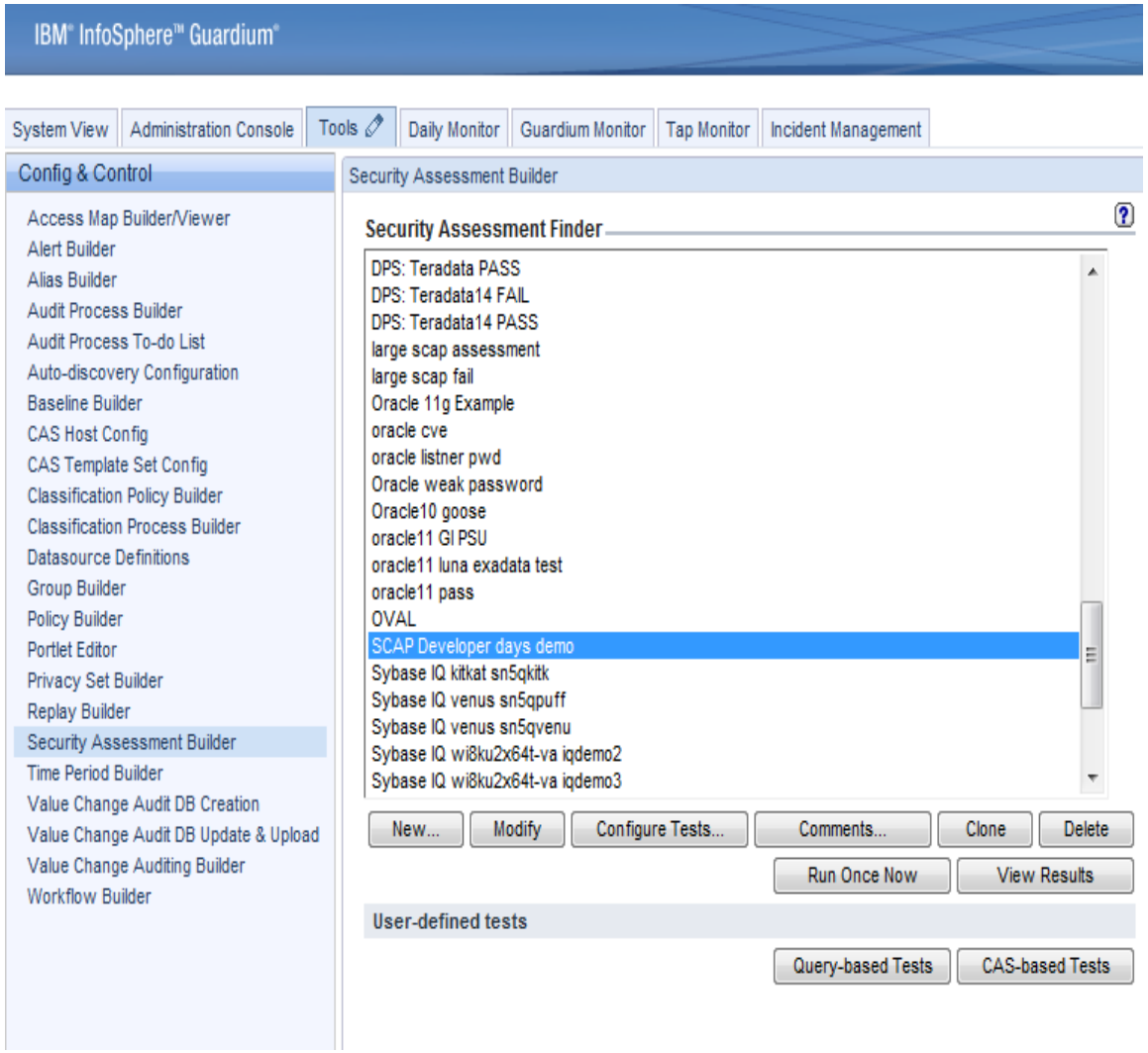
Where We Want to Be

Datasource definitions



Datasource Contain

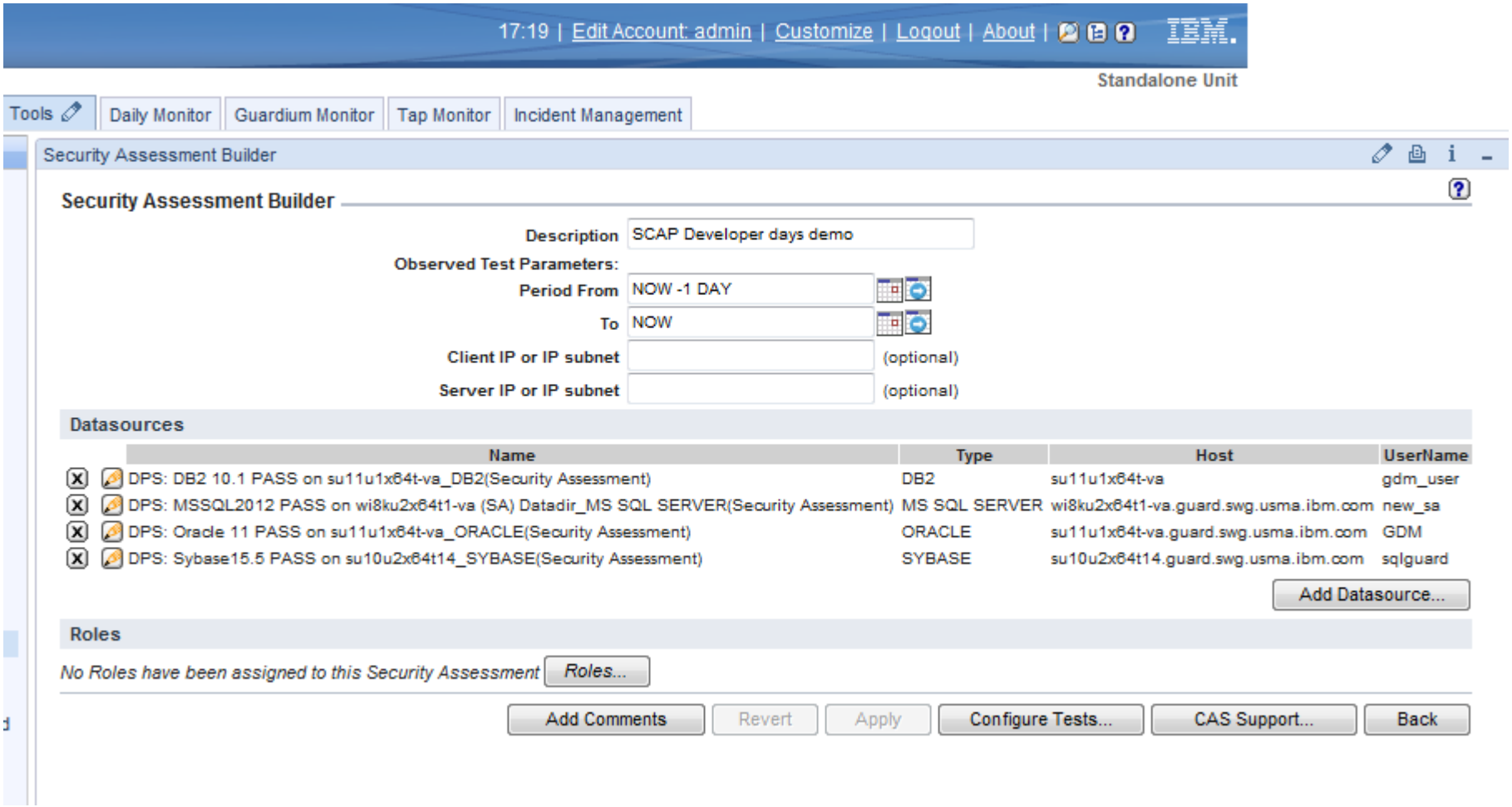
- Datasource name
- Database type
- Description
- User credential
- Encrypted Password
- Host or IP
- Port,
- Database or instance information
- Relevant information for CAS agent.



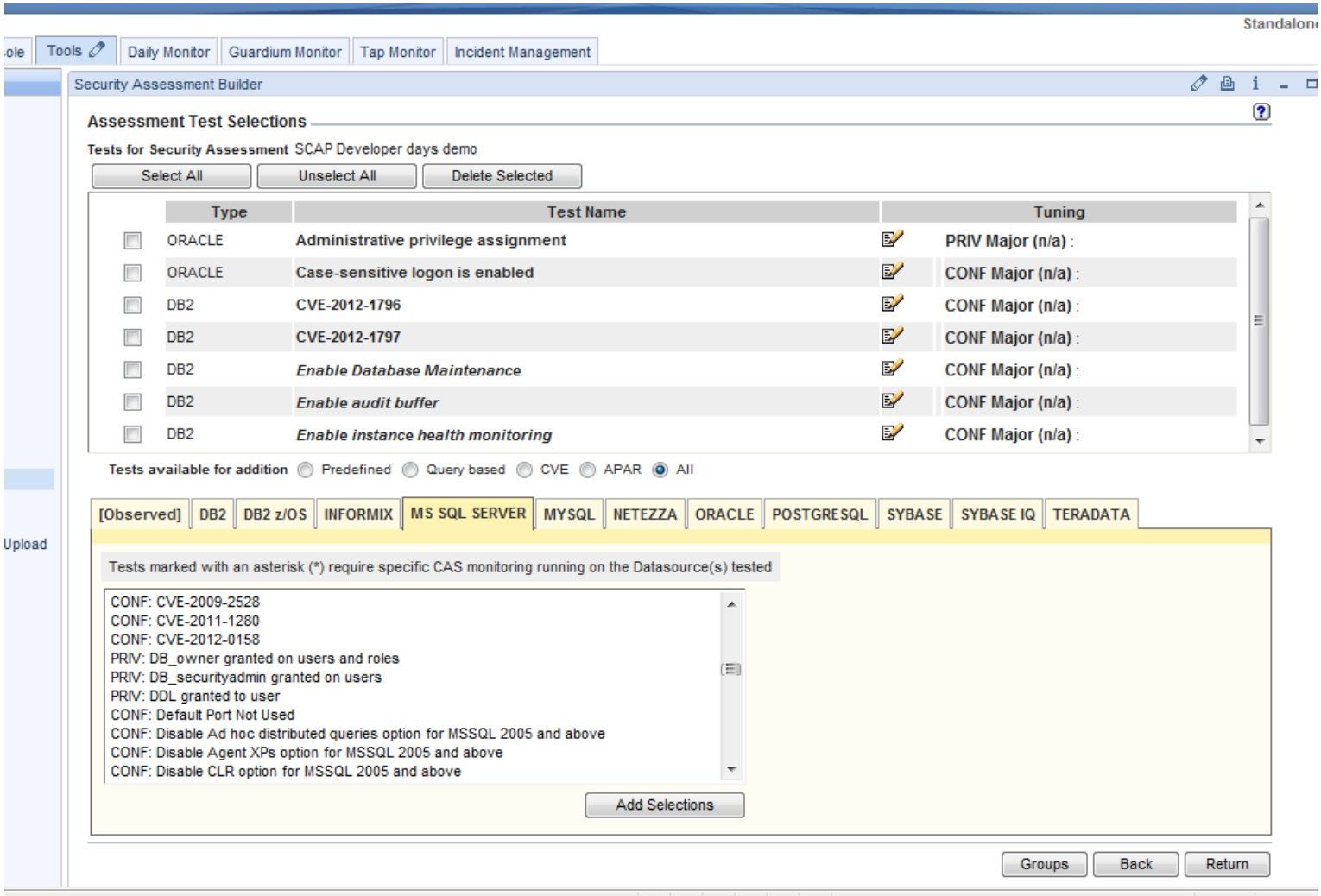
Security Assessment

- List of assessments
- Create or Modify assessment.
- Create custom tests
- Execute assessment

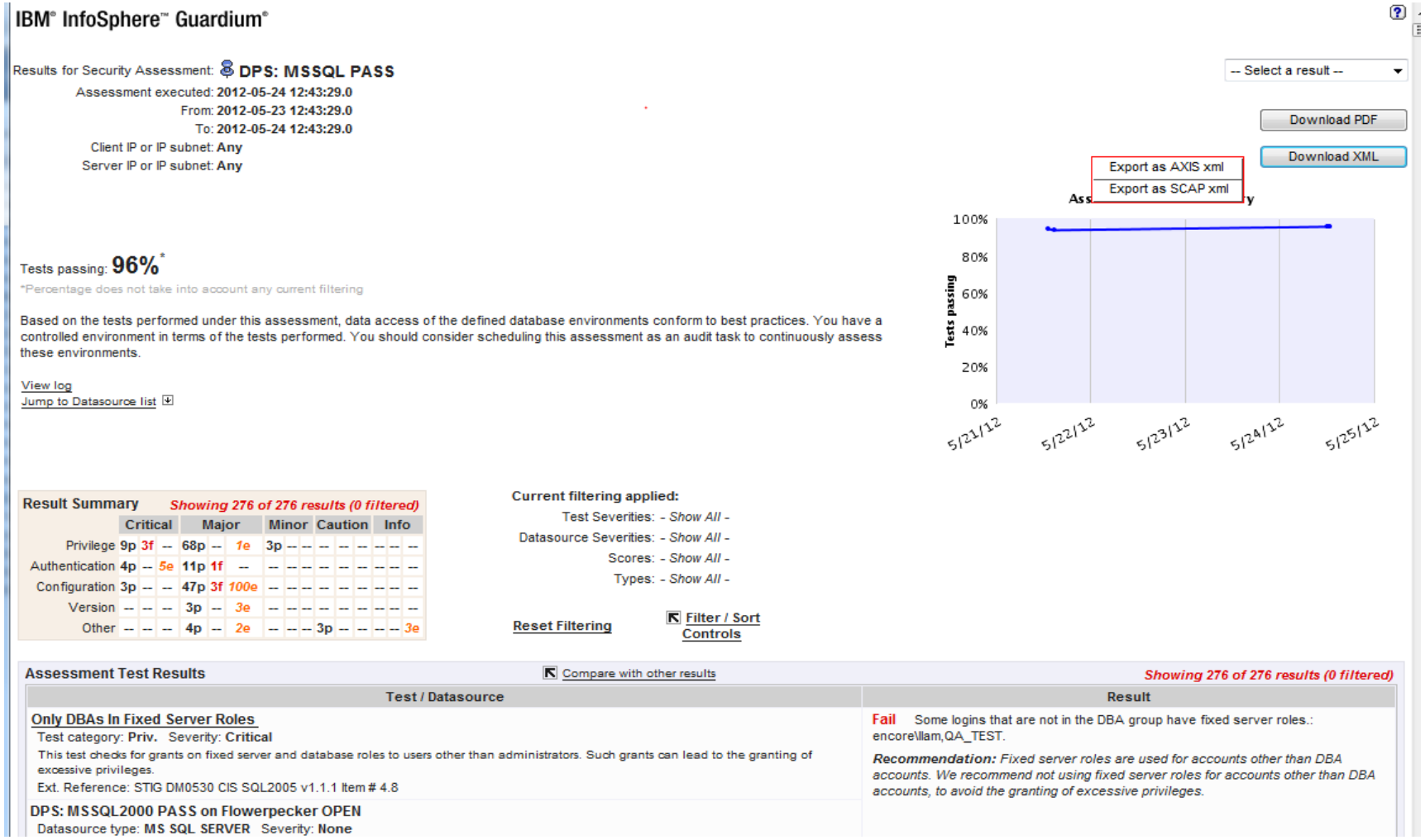
Adding datasources to a given security assessment.



Add tests to an assessment for each DBMS type. Guardium support eleven DBMS types in over 1200 tests.



Security assessment result summary



IBM® InfoSphere™ Guardium®

Results for Security Assessment: DPS: MSSQL PASS
Assessment executed: 2012-05-24 12:43:29.0
From: 2012-05-23 12:43:29.0
To: 2012-05-24 12:43:29.0
Client IP or IP subnet: Any
Server IP or IP subnet: Any

-- Select a result --
Download PDF
Download XML
Export as AXIS xml
Export as SCAP xml

Tests passing: 96%*
*Percentage does not take into account any current filtering

Based on the tests performed under this assessment, data access of the defined database environments conform to best practices. You have a controlled environment in terms of the tests performed. You should consider scheduling this assessment as an audit task to continuously assess these environments.

View log
Jump to Datasource list

Result Summary — Showing 276 of 276 results (0 filtered)

	Critical			Major			Minor			Caution			Info		
Privilege	9p	3f	--	68p	--	1e	3p	--	--	--	--	--	--	--	--
Authentication	4p	--	5e	11p	1f	--	--	--	--	--	--	--	--	--	--
Configuration	3p	--	--	47p	3f	100e	--	--	--	--	--	--	--	--	--
Version	--	--	--	3p	--	3e	--	--	--	--	--	--	--	--	--
Other	--	--	--	4p	--	2e	--	--	--	3p	--	--	--	--	3e

Current filtering applied:
Test Severities: - Show All -
Datasource Severities: - Show All -
Scores: - Show All -
Types: - Show All -

Reset Filtering
Filter / Sort Controls

Assessment Test Results — Compare with other results — Showing 276 of 276 results (0 filtered)

Test / Datasource	Result
Only DBAs In Fixed Server Roles Test category: Priv. Severity: Critical This test checks for grants on fixed server and database roles to users other than administrators. Such grants can lead to the granting of excessive privileges. Ext. Reference: STIG DM0530 CIS SQL2005 v1.1.1 Item # 4.8	Fail Some logins that are not in the DBA group have fixed server roles.: encore\llam,QA_TEST. *Recommendation: Fixed server roles are used for accounts other than DBA accounts. We recommend not using fixed server roles for accounts other than DBA accounts, to avoid the granting of excessive privileges.*
DPS: MSSQL2000 PASS on Flowerpecker OPEN Datasource type: MS SQL SERVER Severity: None	

Security assessment result in detail

IBM® InfoSphere™ Guardium®

Results for Security Assessment: **DPS: MSSQL PASS**
Assessment executed: **2012-05-24 12:43:29.0**
From: **2012-05-23 12:43:29.0**
To: **2012-05-24 12:43:29.0**
Client IP or IP subnet: **Any**
Server IP or IP subnet: **Any**

DDL granted to user
Test category: **Priv.** Test severity: **Major**

DPS: MSSQL2005 PASS on Flowerpecker OPEN
Datasource type: **MS SQL SERVER** Datasource severity: **None**

Pass

DDL privileges is not granted to user in databases as recommend.

Short Description: This test check for DDL granted to user in each MSSQL databases. DDL should be kept limited to only DBA or select admin users. DDL privileges should be granted to role instead of directly to users. Role can then be gratned to a user or group of users.

External Reference: Guardium test 321

Recommendation: *DDL privileges is not granted to user in databases as recommend.*

Details: N/A

Close this window

IBM® InfoSphere™ Guardium®

Results for Security Assessment: **DPS: MSSQL FAIL**
Assessment executed: **2012-05-24 12:19:28.0**
From: **2012-05-23 12:19:28.0**
To: **2012-05-24 12:19:28.0**
Client IP or IP subnet: **Any**
Server IP or IP subnet: **Any**

DDL granted to user
Test category: **Priv.** Test severity: **Major**

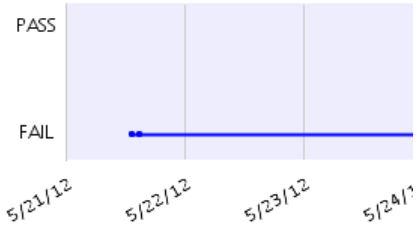
DPS: MSSQL2005 FAIL on Wi3kd2
Datasource type: **MS SQL SERVER** Datasource severity: **None**

Fail

DDL privileges was granted to user in one or more database.

Short Description: This test check for DDL granted to user in each MSSQL databases. DDL should be kept limited to only DBA or select admin users. DDL privileges should be granted to role instead of directly to users. Role can then be gratned to a user or group of users.

Test Result History



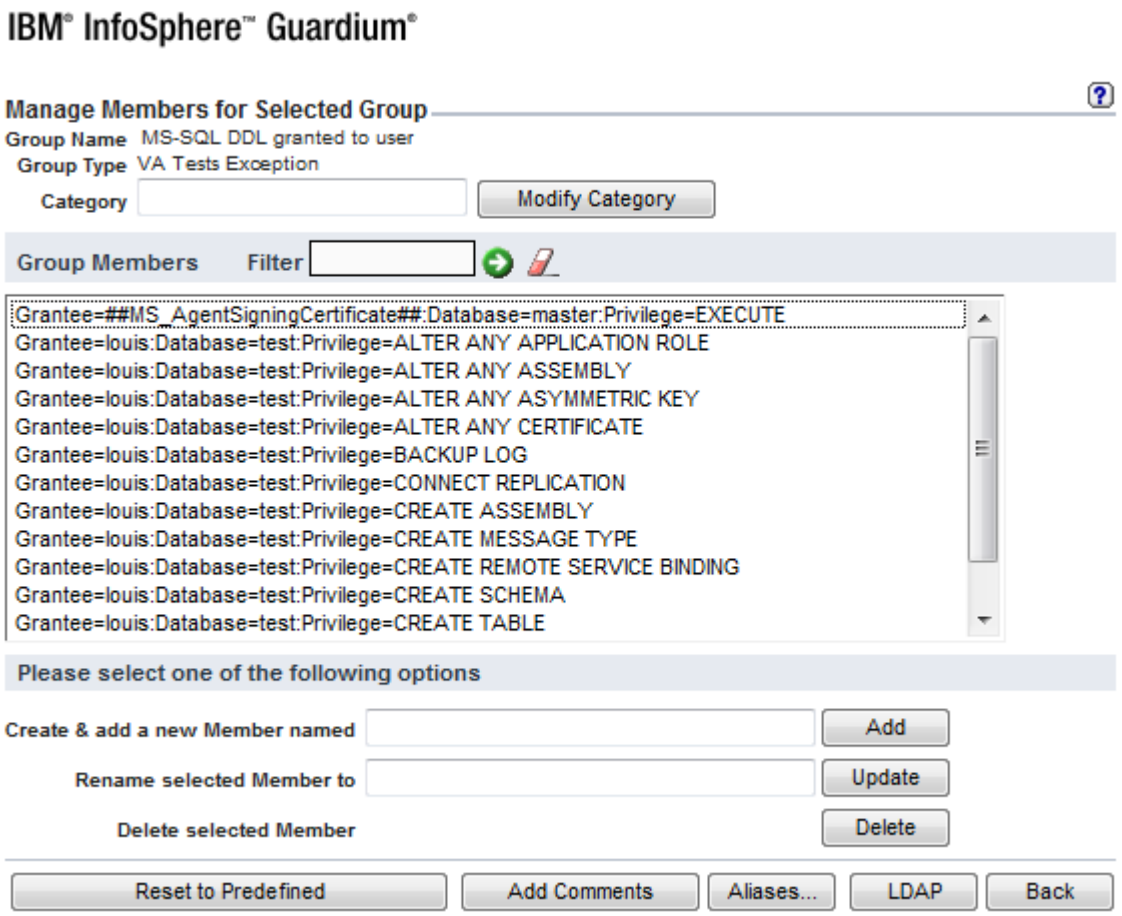
External Reference: Guardium test 321

Recommendation: *We recommend that you grant privileges to role and then grant role to user for security best practice. Please review output to make sure DDL privileges granted directly to users are revoked. REVOKE DDL_PRIVILEGE_NAME from USER_NAME GO*

Details:
Grantee=louis:Database=test:Privilege=ALTER ANY APPLICATION ROLE
Grantee=louis:Database=test:Privilege=ALTER ANY ASSEMBLY
Grantee=louis:Database=test:Privilege=ALTER ANY ASYMMETRIC KEY
Grantee=louis:Database=test:Privilege=ALTER ANY CERTIFICATE
Grantee=louis:Database=test:Privilege=BACKUP LOG
Grantee=louis:Database=test:Privilege=CONNECT REPLICATION
Grantee=louis:Database=test:Privilege=CREATE ASSEMBLY
Grantee=louis:Database=test:Privilege=CREATE MESSAGE TYPE
Grantee=louis:Database=test:Privilege=CREATE REMOTE SERVICE BINDING
Grantee=louis:Database=test:Privilege=CREATE SCHEMA
Grantee=louis:Database=test:Privilege=CREATE TABLE
Grantee=louis:Database=test:Privilege=TAKE OWNERSHIP
Grantee=louis:Database=test:Privilege=VIEW DEFINITION
Grantee=##MS_AgentSigningCertificate##:Database=master:Privilege=EXECUTE
Grantee=bill:Database=master:Privilege=CREATE SCHEMA
Grantee=bill:Database=master:Privilege=CREATE TABLE
Grantee=sqlguard-user:Database=Northwind:Privilege=CONTROL
Grantee=sqlguard-user:Database=Northwind:Privilege=CREATE ASSEMBLY
Grantee=sqlguard-user:Database=Northwind:Privilege=CREATE ASYMMETRIC KEY

Close this window

Exception test facility



CVE tests in detail

IBM® InfoSphere™ Guardium®

Results for Security Assessment: **oracle cve**

Assessment executed: **2012-06-21 15:45:36.0**
From: **2012-06-20 15:45:36.0**
To: **2012-06-21 15:45:36.0**
Client IP or IP subnet: **Any**
Server IP or IP subnet: **Any**

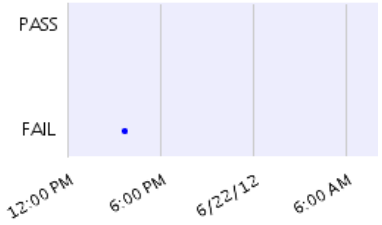
CVE-2012-0082
Test category: **Conf.** Test severity: **Major**

racvm
Datasource type: **ORACLE** Datasource severity: **None**

Fail

System may be vulnerable to CVE-2012-0082

Test Result History



Short Description: Unspecified vulnerability in the Core RDBMS component in Oracle Database Server 10.1.0.5, 10.2.0.3, 10.2.0.4, 10.2.0.5, 11.1.0.7, 11.2.0.2, and 11.2.0.3 allows remote authenticated users to affect integrity and availability via unknown vectors.

External Reference: CVE-2012-0082

Recommendation: *To fix CVE-2012-0082 upgrade to following version/patch level: Version-11.2.0.2, Patch-BP14 Version-11.2.0.2, Patch-CPUJan2012 Version-11.2.0.2, Patch-PATCH15 Version-11.2.0.2, Patch-PSU 11.2.0.2.5.*

Details: N/A

CVSS Information
CVSS Score: 5.5
Access Complexity: LOW
Availability Impact: PARTIAL
Confidentiality Impact: NONE
Integrity Impact: PARTIAL

CVE References

Source	Type	External Link
BID	UNKNOWN	http://www.securityfocus.com/bid/51453
CONFIRM	VENDOR_ADVISORY	http://www.oracle.com/technetwork/topics/security/cpujan2012-366304.html
SECTRACK	UNKNOWN	http://www.securitytracker.com/id?1026527
XF	UNKNOWN	http://xforce.iss.net/xforce/xfdb/72468

IBM® InfoSphere™ Guardium®

Results for Security Assessment: **oracle cve**

Assessment executed: **2012-06-21 15:45:36.0**
From: **2012-06-20 15:45:36.0**
To: **2012-06-21 15:45:36.0**
Client IP or IP subnet: **Any**
Server IP or IP subnet: **Any**

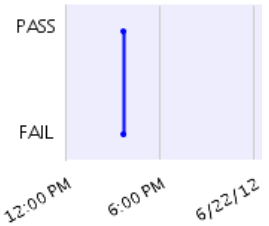
CVE-2011-2257
Test category: **Conf.** Test severity: **Major**

racvm
Datasource type: **ORACLE** Datasource severity: **None**

Pass

System is not vulnerable to CVE-2011-2257

Test Res



Short Description: Unspecified vulnerability in the Database Target Type Menus component in Oracle Database Server 10.1.0.5, 10.2.0.3, 10.2.0.4, 10.2.0.5, 11.1.0.7, 11.2.0.1, and 11.2.0.2; and Oracle Enterprise Manager Grid Control 10.1.0.6, 10.2.0.5, and 11.1.0.1; allows remote attackers to affect confidentiality, integrity, and availability via unknown vectors.

External Reference: CVE-2011-2257

Recommendation: *CVE-2011-2257 is fixed for the system's version and patch level.*

Details: N/A

CVSS Information
CVSS Score: 6.8
Access Complexity: MEDIUM
Availability Impact: PARTIAL
Confidentiality Impact: PARTIAL
Integrity Impact: PARTIAL

CVE References

Source	Type	External Link
CERT	UNKNOWN	http://www.us-cert.gov/cas/techalerts/TA11-201A.html
CONFIRM	VENDOR_ADVISORY	http://www.oracle.com/technetwork/topics/security/cpujuly2011-313328.html

How We Get There

- Phased approach
  - Address <sql57_test> problems first
  - Address non-OVAL problems later
- Involve tool vendors
- Add support for <sql57_test> to *ovaldi* as a proof-of-concept