Introduction to Hedgehog: Realtime Database Security
The complexity of the problem

Outsiders

Corporate Firewall

Authentication & Access Control

Insiders

Skilful Users
(Privileged & Not privileged)

CRM

HR

ERP

DB

DB

DB
Monitoring All database activity

- DB
- CRM
- HR
- ERP
- Insiders
- Privileged Users
- Outsiders
- Corporate Firewall
- Authentication & Access Control
- Stored Proc.
- Trigger
- View
- Data
- Shared Memory
- DBMS
- Listener
- Bequeath
- Local Connection
- Network Connection
Database Vulnerabilities Go Unpatched

CVE (Common Vulnerabilities and Exposures, an independent security website) no. of vulnerabilities per DBMS:

- Oracle: 129
- MySQL: 90
- DB2: 54
- Informix: 35
- Sybase: 5
- MSSQL: 4

No. of vulnerabilities reported Jan 2006-June 2008

Sentrigo Survey of 315 Oracle professionals, Jan 2008
Introducing Hedgehog

All database transactions (externally- or internally-initiated) go via the shared memory.
Hedgehog Enterprise: Real-time monitoring and prevention

- Principle: Guard the data, not the access paths
- See all activity, regardless of where it originates, and be close enough to intervene
- Agnostic to access path, access method, environmental variables, and data complexity
- Can terminate user sessions and quarantine users
- Minimal impact on performance - typically less than 5% of a single CPU
- Fully scalable
- Easy: download, install, use
Hedgehog: Architecture Overview

- Network
  - Sensors
    - DB
  - 3rd party mgmt tools
- Hedgehog JavaEE Server (software)
- Web-based Admin Console
- Alerts
How Hedgehog Works

**Rule**
- **Trigger**: IF App<>’SAPFinance’ AND object = ‘CC_Table’ THEN
- **Action**: Send HIGH Alert
  - Send mail to: security team
  - Terminate User Session
  - Quarantine User 60 minutes

**Policy**
- Rule 1
- Rule 2
- Rule 3
- Rule 4
- Rule 5...

**Layers**
- Virtual Patching
- Compliance Templates
- Community Best Practices
- Customer-Defined

---

The diagram illustrates the process and components of how Hedgehog works, focusing on rule triggering and policy execution within a layered security approach.
IDentifier Diagram

App User 1
App User 2
App User 3

Application Server

Transactions

Hedgehog IDentifier

Pooled Connection

Hedgehog Enterprise

DBMS

App User 1
App User 2
App User 3
IDentifier - End-User Accountability

Alert 100322008 Details (28 Feb 2008 23:53:37)

Sensor: oracle-erp-sensor
Session ID: 114
Serial#: 88
User: WEBSPHERE
OS User: WEBSPHERE
CMD Type: SELECT
Action: POST
Log on time: 

Statement:
SELECT T1.CUSTOMERID, T1.PASSWORD, T1.FI
T1.ADDR1, T1.ADDR2, T1.ADDRCITY, T1.ADDR3,
T1.PHONE FROM CUSTOMER T1 WHERE T1.CUS

Rules: catch all
Accessed Objects: WEBSPHERE CUSTOMER

DBMS: db10203
Application: 
IP: 192.168.150.33
Hostname: joe-desktop
Module: tsByWebSphere/servlet/
Client ID: 192.168.78.21/johnsmith@google.com

Resolution: Unresolved

sentrigo™
Hedgehog IDentifier

- Plug-in to the application that sends client ID, originating IP, and URL used (with web-based applications) to the Hedgehog sensor

- Available for JAVA/J2EE apps and .NET

- Works seamlessly with most standard containers and Oracle JDBC (including Weblogic, Websphere, JBoss, Tomcat)
Thank You!