May 6–10, 2007

San Jose Convention Center

San Jose, California, USA

Session: I04

Most Recent Advanced in DB2 Attacks

IDUG® 2007 North America

Aaron Newman
Application Security, Inc.

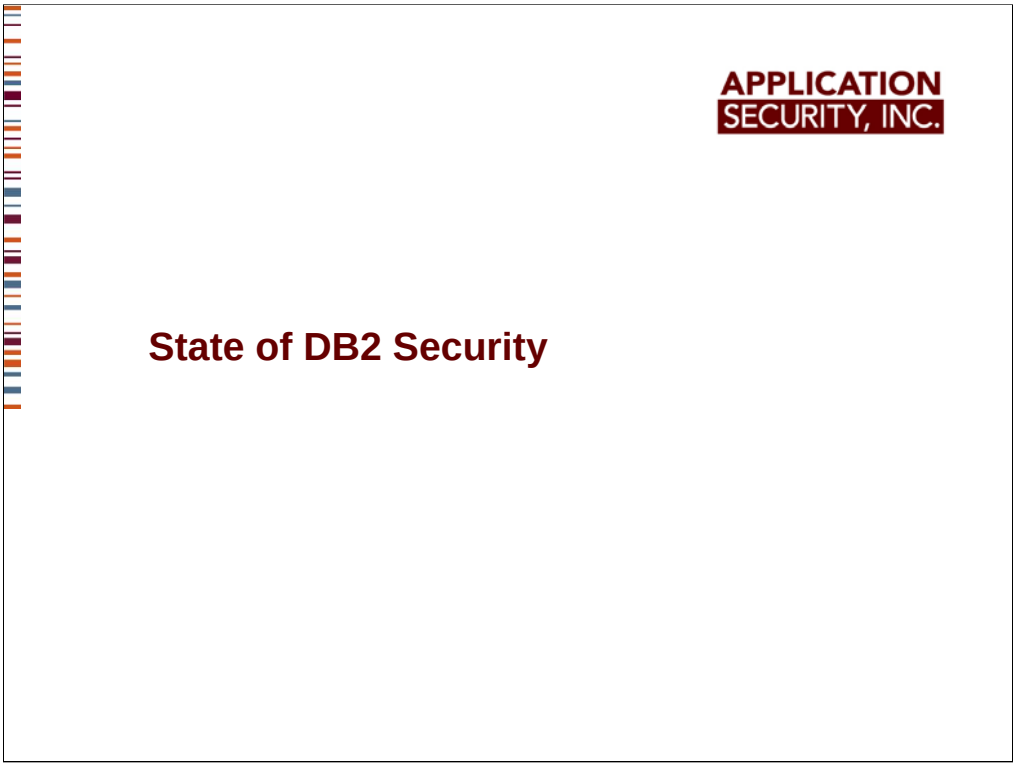
May 7, 2007 4:20 a.m. – 5:20 a.m.

Platform: DB2 Universal Database for LUW

GoFurther

Main Points

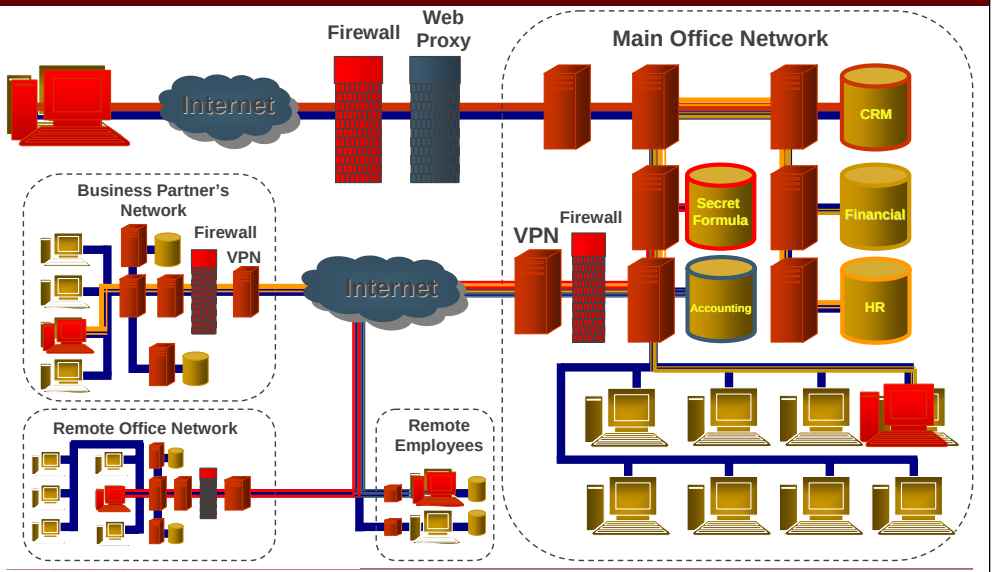
- State of DB2 Security
- Securely Configuring DB2
  - Demo – Password Attack
- Securing DB2 in a Web Application
  - Demo – SQL Injection
- Database Vulnerabilities
  - Demo - Remote Command Execution
- Resources, Conclusions, & Questions

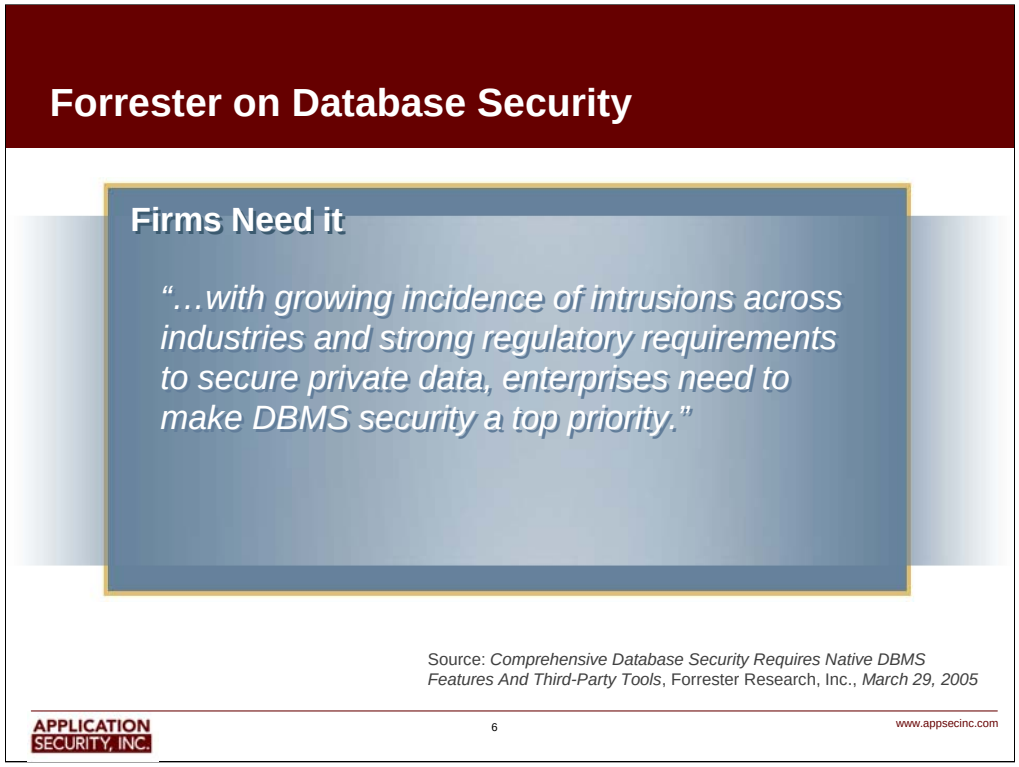

APPLICATION
SECURITY, INC.
State of DB2 Security

Evolving DB Threat Environment

- A decade ago, databases were:
  - Physically secure
  - Housed in central data centers – not distributed
  - External access mediated.
  - Security issues rarely reported
- Now increasingly DB's externally accessible:
  - Suppliers/Customers directly connected
  - Customers & partners directly sharing data
- Data is most valuable resource in application stack
  - Value increases with greater integration & aggregation
  - Opportunities for data theft, modification, or destruction
- DB security a growing problem:
  - Need to design for & test for high threat public internet scenario
  - Recommend defense in depth

What about perimeter security?
Barrier Defense Is No Longer Enough





Here is another report from the Forester Research group, just release a report in March 2005. Growing inc

Attackers focusing on databases

http://news.softpedia.com/news/Toying-with-Microsoft-s-breaches-is-no-longer-fun-for-hackers-1603.shtml

.. **until now the hackers have mostly targeted Microsoft's software products**, starting with this year, it seems their attention has been draw to other products and services ... According to a recently published study, ...it's clear they are exploring new territories besides the old Windows .. the **software products developed by database vendors have started to be more and more targeted by hackers**.

External threats

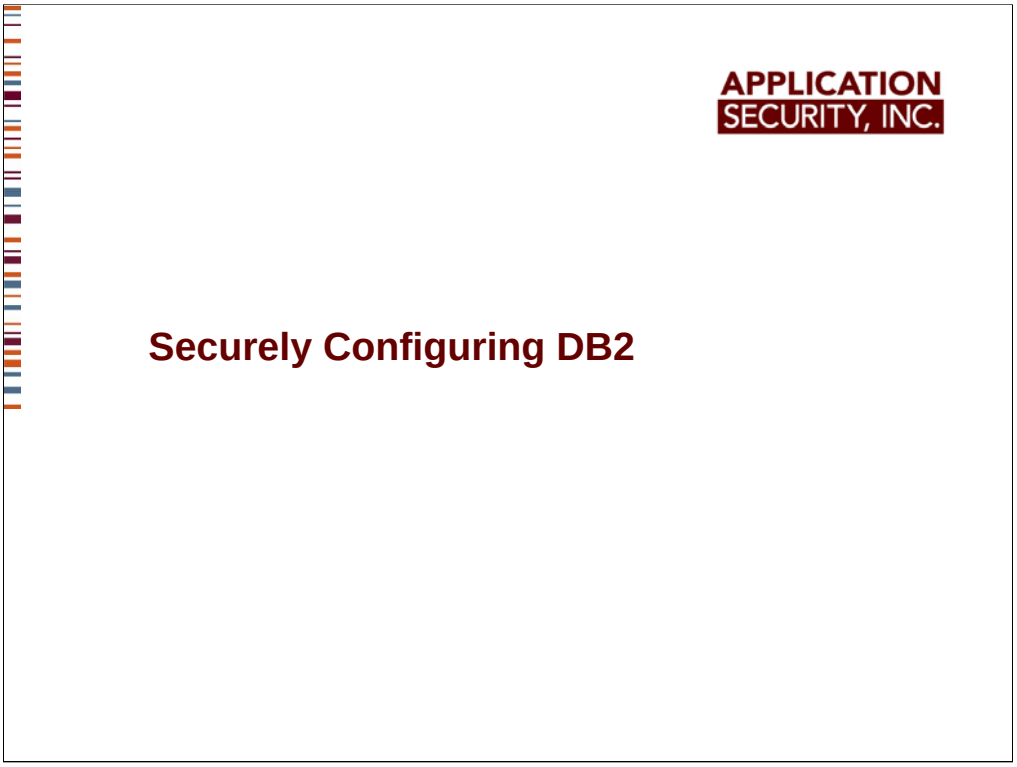
- Exposing a database to the Internet
  - DO NOT DO THIS!
  - Protocol stacks are riddled with overflows
  - ALL the database vendors!
- Behind a firewall
  - Recommend data be served through web app
  - Still vulnerable to application level attacks
  - Still vulnerable to internal attacks

Internal threats

- Need to focus less on people getting through perimeter
- Need to focus more on securing data at the source
- Attacks from DBAs, sysadmins, and legimate users
- Need to focus not only on vulnerabilities
  - Be able to see who is doing what in your applications
  - Providing auditing and accountability for users and database administrators
  - Being able to identify malicious activity

http://www.thesmokinggun.com/archive/0623042aol1.html

"JUNE 23--An AOL software engineer was arrested today for stealing the company's entire subscriber list--totaling 92 million screen names--and selling it to a 21-year-old Las Vegas spammer. "


APPLICATION
SECURITY, INC.
Securely Configuring DB2

Configuring Authentication

- Controlling where authentication occurs
- NEVER allow server to accept CLIENT authentication
- Allows client to authenticate the user (*trust_allclnts*)
  - inherently insecure
- Recommended settings
  - DCE_ENCRYPT
  - SERVER_ENCRYPT
  - KRB_SERVER_ENCRYPT
- Discouraged setting
  - CLIENT
  - SERVER

Encryption During Authentication

- Controlling passwords traversing the network
  - ALWAYS require *_ENCRYPT protocols
  - Passwords encrypted by DB2 at client
  - Before being sent to the server
- WARNING
  - You must set clients to use *_ENCRYPT protocols

DB2 CONNECT Packet

- Below is an example of a connection:
  - account is “User”, password is “3”

```
000000D0  04 00 00 00 04 00 00 00 34 F7 12 00 00 00 00 00   ♦...♦...4÷↕.....
000000E0  01 00 00 00 00 00 00 00 08 00 00 00 01 00 00 00   ☺.......◘...☺...
000000F0  44 F7 12 00 00 00 00 00 00 00 00 00 00 00 00 00   D÷↕.............
00000100  88 00 00 00 0C 00 00 40 A0 F6 12 00 00 00 00 00   ê...↓..@á÷↕.....
00000110  01 00 00 80 00 00 00 00 00 00 00 00 00 00 00 00   ☺..Ç............
00000120  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00000130  04 00 00 00 02 00 00 40 8C 79 85 00 00 00 00 00   ♦...☻..@íyà.....
00000140  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00000150  00 00 00 00 02 00 00 00 03 00 00 00 78 F6 12 00   ....☻...♥...x÷↕.
00000160  6A 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00   j☻..............
00000170  00 00 00 00 01 53 00 00 01 00 03 00 01 00 00 75   ....☺S..☺.♥.☺..u
00000180  73 65 72 33 53 41 54 43 54 4C 44 42 05 01 01 23   ser3SATCTLDB♣☺☺#
00000190  E4 04 00 00 30 33 37 34 30 35 34 38 01 00 00 00   Σ♦..037405480☺...
000001A0  43 30 41 38 30 31 39 42 2E 34 31 30 37 2E 30 32   C0A8019B.4107.02
000001B0  31 31 31 33 31 36 32 33 35 36 00 00 00 00 00 00   1113162356......
000001C0  00 00 00 00 00 00 00 00 00 00 53 51 4C 30 36 30   ..........SQL060
```

Default Username/Passwords

- Installed with the database
  - db2admin/db2admin, db2as/ibmdb2, dlfm/ibmdb2
  - db2inst1/ibmdb2, db2inst2/ibmdb2, etc…
  - db2fenc1/ibmdb2, db2fenc2/ibmdb2, etc…
- Third-party default passwords
- Passwords can be “guessed”
  - Attacking a single account with 100k passwords
  - Attacking many accounts with a few very common passwords
  - People leave test/test or password same as username
- Password dictionaries
  - http://www.openwall.com/passwords/wordlists/
  - The wordlists are intended primarily for use with password crackers …

APPLICATION SECURITY, INC.

Demo –
Password attacks

Locking down OS Privileges

- Unix/Linux
  - Set all DB2 file permissions to –rwxrwx--- or more restrictive
  - Do not run daemon as root
  - Rename OS accounts and select strong password
- Windows
  - Set file permissions to Owner only
  - Do not run service as LocalSystem
  - Run service as local non-privileged user
  - Lock down registry permissions on DB2 keys

Locking down database privileges

- Remove permissions granted to public
- Review users granted SYSADM group
- Revoke privileges on system catalogs
  - SYSIBM.DBAuth
  - SYSIBM.TabAuth
  - SYSIBM.INDEXAuth
  - SYSIBM.COLAuth
  - SYSIBM.SCHEMAAuth
  - SYSIBM.PASSTHRUAuth
- Create UDFs as fenced


APPLICATION
SECURITY, INC.
Securing DB2 in a Web Application

Can attacks go through a firewall?

- YES!!!
- Firewall configuration
  - Block access through port 523, 50000-50010, 446
  - Only allow traffic to port 80
  - Block UDP as well as TCP
- SQL Injection
  - Not specific to DB2
  - a web programming problem

How Does It Work?

- Try to modify the query
- Change:

```
Select * from my_table
      where column_x = '1'
```

- To:

```
Select * from my_table
      where column_x = '1'
UNION select credit_card_number
      from orders where 'q'='q'
```

Example JSP Page

```
Package myseverlets;
<….>

String sql = new String(“SELECT * FROM
WebUsers WHERE Username=’” +
request.getParameter(“username”) + “’ AND
Password=’” +
request.getParameter(“password”) + “’”

stmt = Conn.prepareStatement(sql)
Rs = stmt.executeQuery()
```

Valid Input

- If I set the username and password to:
  - Username: Bob
  - Password: Hardtoguesspassword
- The SQL statement is:

```
SELECT * FROM WebUsers WHERE
  Username='Bob' AND
  Password='Hardtoguess'
```

Hacker Input

- Instead enter the password:

```
Aa' OR 'A'='A
```

- The SQL statement now becomes:

```
SELECT * FROM WebUsers WHERE
  Username='Bob' AND
  Password='Aa' OR 'A'='A'
```

- The attacker is now in the database!

SELECT from other tables

- SELECT arbitrary data
  - Name the tables to SELECT against
- UNION statement
  - Adds/executes second SQL statement
  - Column types and number must match

Sample ASP Page

```
Dim sql

Sql = “SELECT ProductName FROM
Products WHERE ProductCategory=’” &
request.form[“product_category”] & “’”

Set rs = Conn.OpenRecordset(sql)
‘ return the rows to the browser
```

Valid Input

- Set the product_category to :
  - `DVD Player`
- The SQL Statement is now:
  - `SELECT ProductName FROM Products WHERE ProductCategory='DVD Player'`

Hacker Input

- Set the product_category to :
  - `test' UNION select credit_card_number from CUSTOMERS where 'a' = 'a`
- The SQL Statement is now:
  - `SELECT ProductName FROM Products WHERE ProductCategory='test' UNION select credit_card_number from CUSTOMERS where 'a'='a'`

Reverse Engineering Database Schema

- Hacker doesn't know the database schema
  - Can they figure it out?
- Yes!!!
- UNION SQL statement with system catalog
- Tables with information
  - SYSIBM.SYSTABLES
  - SYSIBM.SYSCOLUMNS
  - SYSIBM.SYSSERVERS
- Retrieving the list of tables
  - UNION select name from SYSIBM.SYSTABLES;

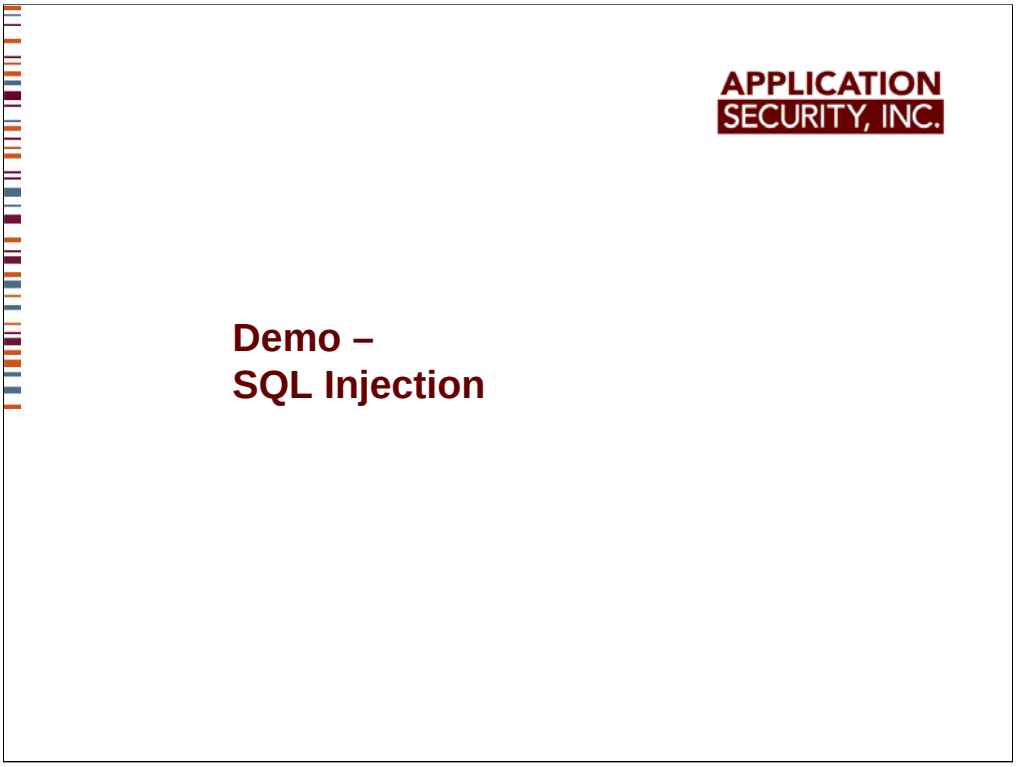
Preventing SQL Injection

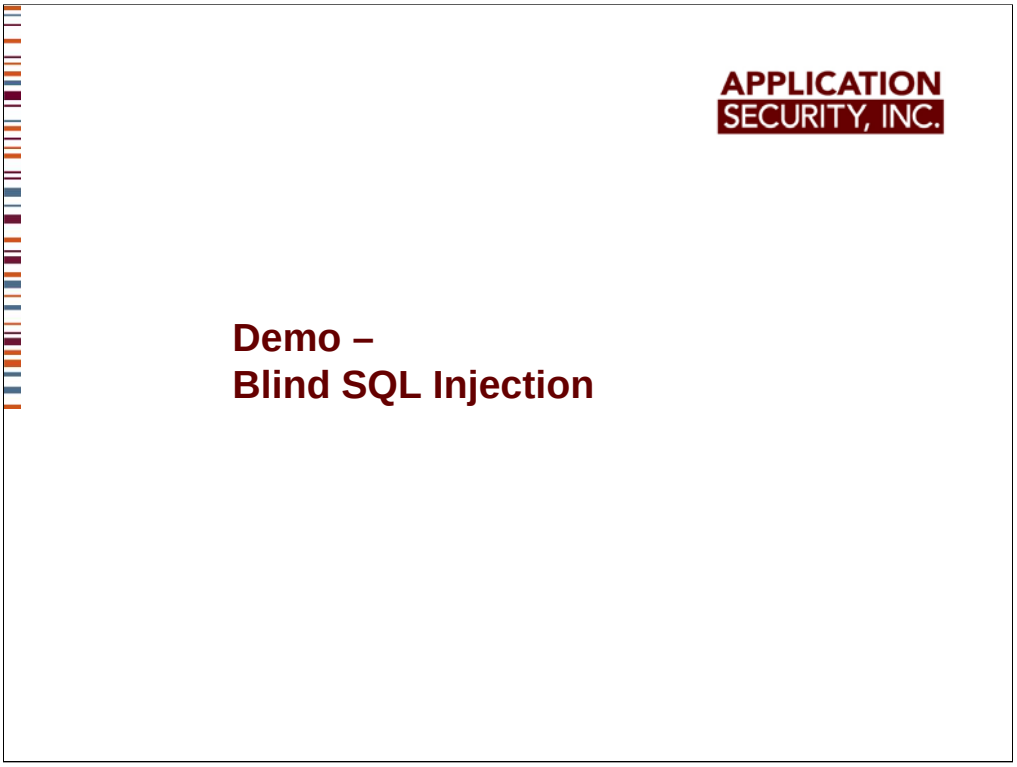
- Bind variables – don't concatenate SQL strings
- Right way

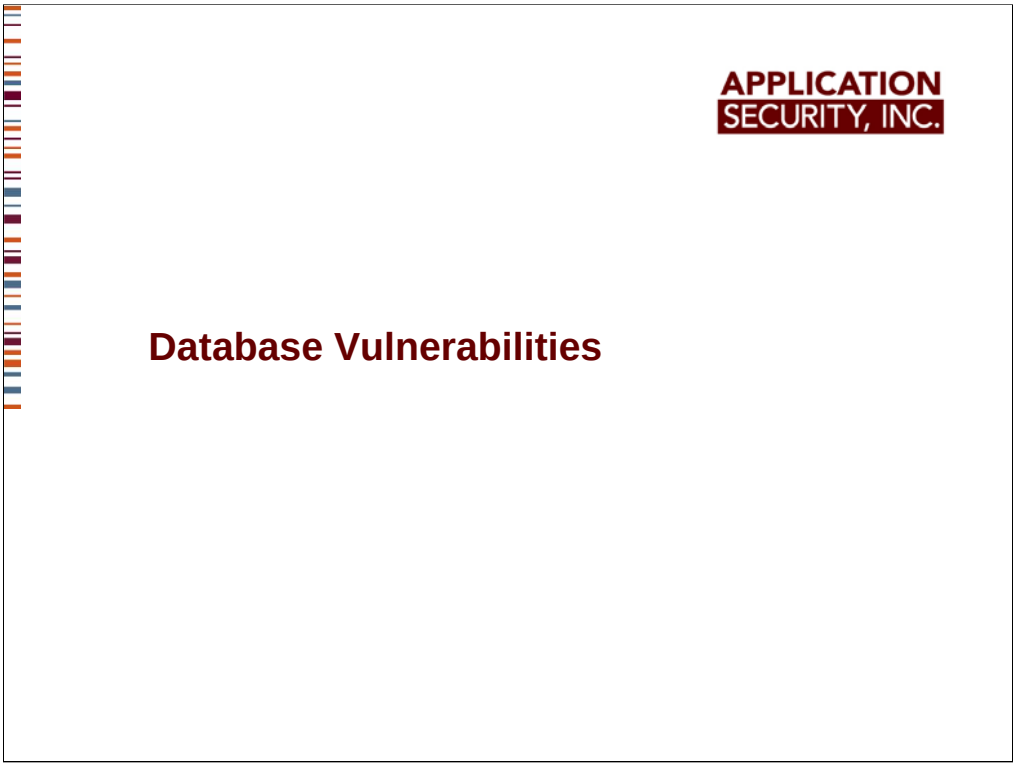
```
String sql = new string(
 "UPDATE EMPLOYEE SET BONUS=?")
```

- Wrong way

```
String sql = new string(
 "UPDATE EMPLOYEE SET BONUS=" +
 request.getParameter("bonus"))
```


APPLICATION
SECURITY, INC.
Demo –
SQL Injection


APPLICATION
SECURITY, INC.
Demo –
Blind SQL Injection

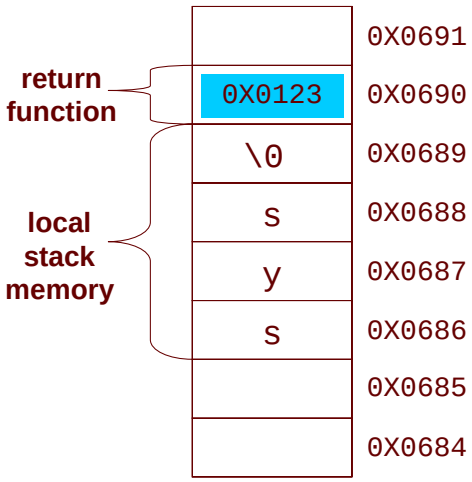

APPLICATION
SECURITY, INC.
Database Vulnerabilities

What is a buffer overflow

- When a program attempts to write more data into buffer than that buffer can hold…
- …Starts overwriting area of stack memory
  - That can be used maliciously to cause a program to execute code of attackers choose
  - Overwrites stack point

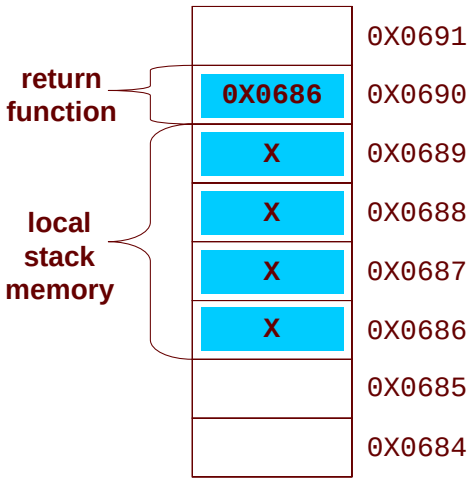
Mechanics of stack-based buffer overflow

- Stack is like a pile of plates
- When a function is called, the return address is pushed on the stack
- In a function, local variables are written on the stack
- Memory is written on stack
  - char username[4] reserved 4 bytes of space on stack



Mechanics of stack-based buffer overflow

- When function copies too much on the stack
- The return pointer is overwritten
- Execution path of function changed when function ends
- Local stack memory has malicious code



Installing the latest FixPak

- Usually addresses latest buffer overflows
- Most current versions (will be outdated by the time you read this!!!)
  - V6.1 – Upgrade to V7.2 or V8.1
  - V7.1 – Upgrade to V7.2
  - V7.2 – Fix Pak 14
  - V8.1 – Fix Pak 14
  - V9 – Fix Pak 2
- Download from
  - http://www-4.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/download.d2w/

Denial of Service attacks

- Denial of service allows an attacker to bring down the database
  - Affects availability of the database
- Two functions used to crash the database engine
  - `select to_char('test', '') from sysibm.sysdummy1`
  - `select to_date('test', '') from sysibm.sysdummy1`
- Requires no special privileges in database
- Fixed in: DB2 version 8.1 FixPak 8

Local Privilege Escalation Attacks

- Some attacks used to become privileged users
  - Based on a non-privileged user on the UNIX operating system
- Find files that are setUID
  - Runs under file owners privileges
  - Most dangerous when setUID file is owned by root
- File used to run fenced libraries - db2fmp
  - Installed setUID root
  - Buffer overflow occurs when passing in overly long parameter
- Fixed in
  - DB2 version 7.2 FixPak 13, version 8.1 FixPak 8

Buffer overflow in SQL functions

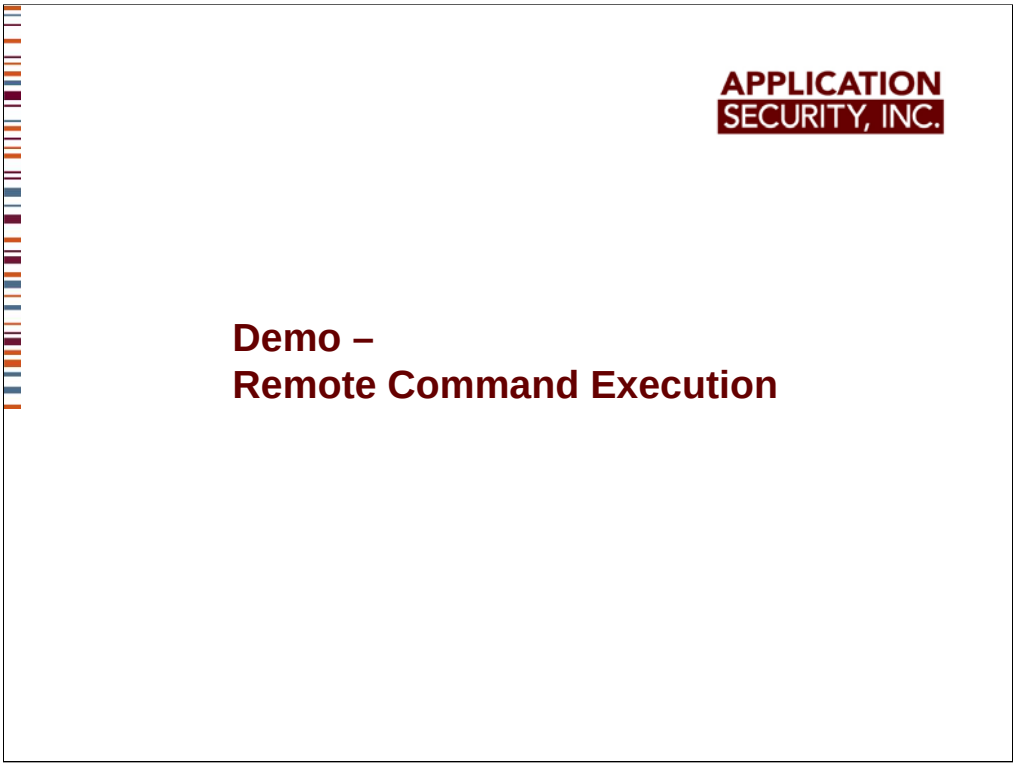
- Must have a user ID in the database
- May or may not require special privileges
- Example: generate_distfile procedure
  - Overly long string as an argument
  - Cause a stack-based buffer overflow
  - Function exported by db2dbappext.dll
  - Third parameter of the function is a file name
- Affects both Windows and UNIX

JDBC Applet Server buffer overflow

- JDBC Applet Server and Control Center
  - Runs on port 6789 and 6790
- Capability to remotely administer DB2
- Stack-based buffer overflow in the authentication to the JDBC Applet Server
  - Requires no valid account on the database
- Fixed in
  - DB2 version 7.2 FixPak 13
  - DB2 version 8.1 FixPak 8

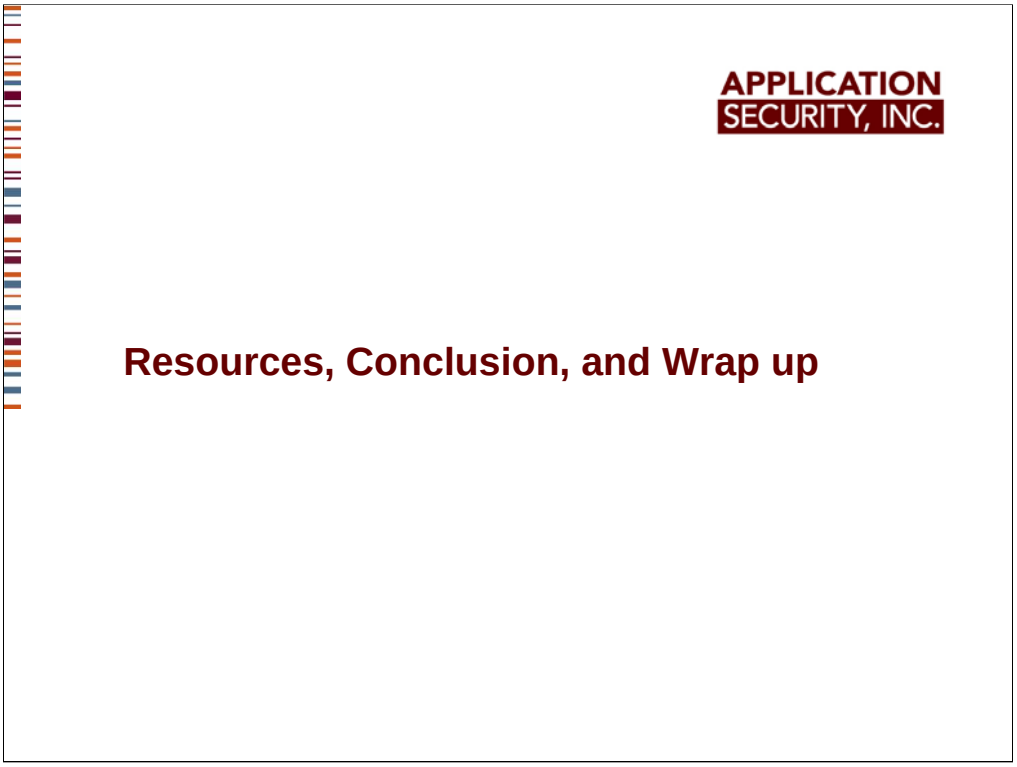
Remote Command Execution

- DB2 provides ability to run remote operating system commands
  - Used for distributed databases/clustering
- Supported in DB2 v8.1
  - Listens on a Named Pipe call \\server\DB2REMOTECMD
  - Requires a non-privileged account on the server
  - Runs commands using full privileges of the DB2 server
- Fixed in v8.1 FixPak 5
- Exploit code from the “ShellCoder’s Handbook”


APPLICATION
SECURITY, INC.
Demo –
Remote Command Execution

Most Recent Vulnerabilities

- Remote DoS during Connect
  - http://www.appsecinc.com/resources/alerts/db2/2006-09-05.shtml
  - Fixed in UDB V8.1 FP14 (AKA 8.2 FP7)
- Remote DoS in sqle_db2ra_as_recvrequest
  - http://www.appsecinc.com/resources/alerts/db2/2006-11-30.shtml
  - Fixed in UDB V8.1 FP14 (AKA 8.2 FP7)
- Authorization Bypass Vulnerability
  - http://www.secunia.com/advisories/24283
  - Fixed in UDB V9.1 FP2
- More Denial of Service attacks coming


APPLICATION
SECURITY, INC.
Resources, Conclusion, and Wrap up

How to Combat Hackers

- Defense in depth
- Multiple levels of security
  - Perform audits and pen tests on your database on a regular basis
  - Encryption of data-in-motion
  - Encryption of data-at-rest
  - Monitor your log files
  - Implement intrusion detection

Session: I04

Most Recent Advances In DB2 Attacks

Aaron Newman

Application Security, Inc.

anewman@appsecinc.com

GoFurther