Application Security, Inc.

Hack-proofing DB2

Aaron C. Newman, CTO/Founder

APPLICATION SECURITY, INC.

www.AppSecInc.com

Main Points

- State of DB2 Security
- Securely Configuring DB2
- Securing DB2 in a Web Application
- Database Vulnerabilities
- Resources, Conclusions, & Questions

Application Security, Inc.

State of DB2 Security

Hack-proofing DB2

In the media

“Look what they've done to my database, Ma” By John Leyden, The Register

- 1 out of 10 corporate databases connected to the Internet had a breach of security last year.
- Survey of 750 US database developers
  - Two-thirds of the respondents … say that their most important development task … is the ability to provide dynamic Web access to their databases
  - Survey conducted by market research firm Evans Data Corporation

http://www.theregister.co.uk/content/55/23800.html

Evolution of DB2

- Historically DB2 has lived on a mainframe
  - Resided in a fairly secure network
- More and more we see DB2 exposed to the large world
  - DB2 on Linux/Windows/Unix
  - Used as backend for web applications
- With these changes in DB2 comes increased risk
  - Of hackers
  - Of malicious internal users

Underground Hacking Conventions

Presentations on hacking databases - 2001

- Blackhat – 1
- Defcon – 1

Presentations on hacking databases – 2002

- Blackhat – 5
- Defcon - 4

Blackhat –2003

- Track dedicated to hacking databases

A Secure Mindset

- In order to protect yourself, your should establish a paranoid mindset
  - Should not trust anything passed to the app
  - Should not trust that single layer will be secure
  - Should not trust developers, vendors, DBA, etc…
- Security is only as strong as the weakest link
  - Attack those aspects that have been neglected
  - Lock down development/test/backup databases

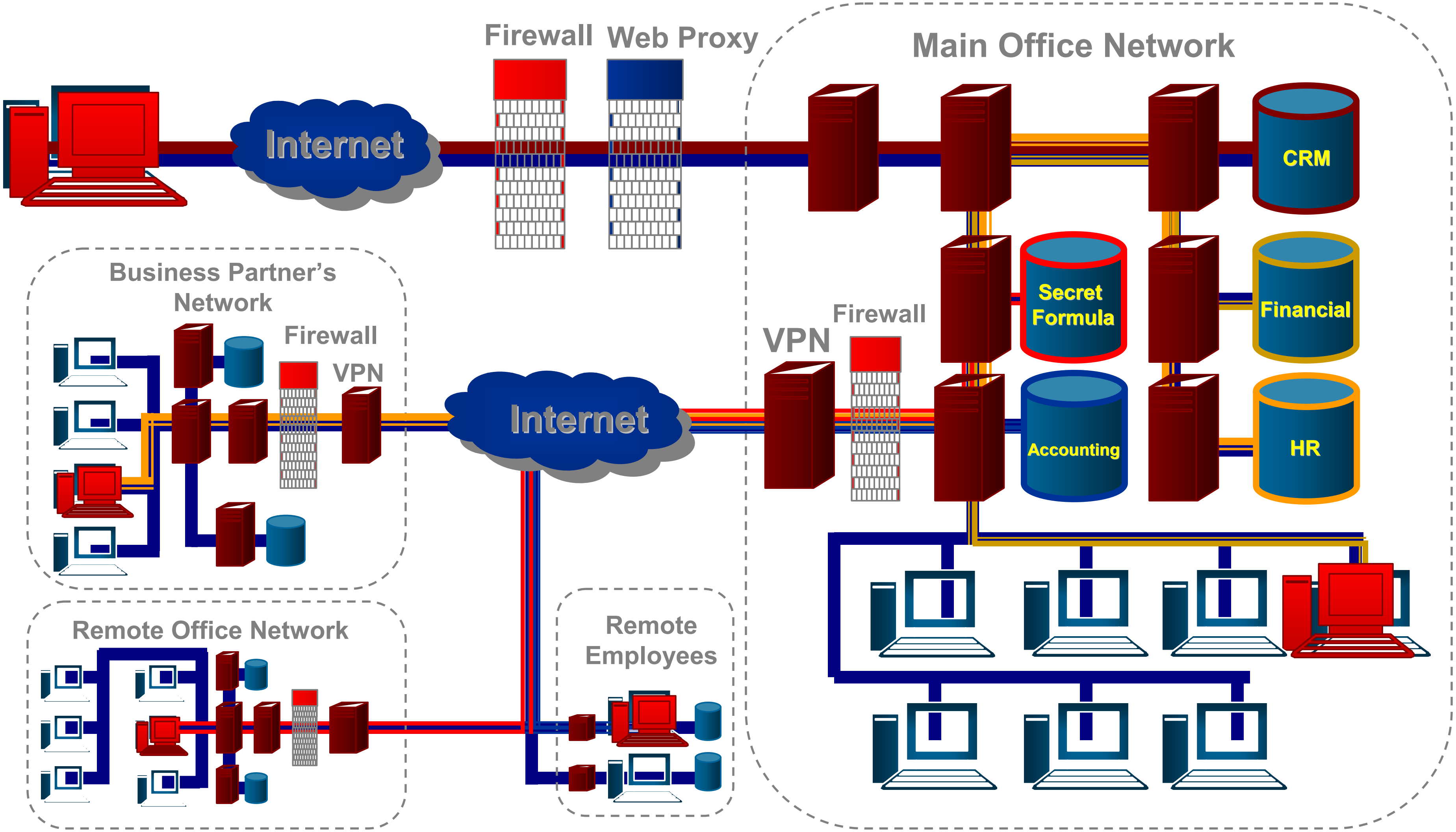
Security is a process

- Security of application depend on administrator more than on software itself
- Security is a process, not a product
- Even the most secure software and hardware is not safe if not properly locked down.

Secure Behind a Firewall

- Is your database secure because it is behind a firewall?
  - NO!!!
- Most security compromises are a result of inside jobs
- Internal threats are the most dangerous
  - Non-privileged users in the database
- Perimeter security has crumbled
  - Information needs to be more and more accessible
  - Business transactions cross corporate boundaries

Perimeter is no longer viable



Application Security, Inc.

Securely Configuring DB2

Hack-proofing DB2

APPLICATION SECURITY, INC.

www.AppSecInc.com

Configuring Authentication

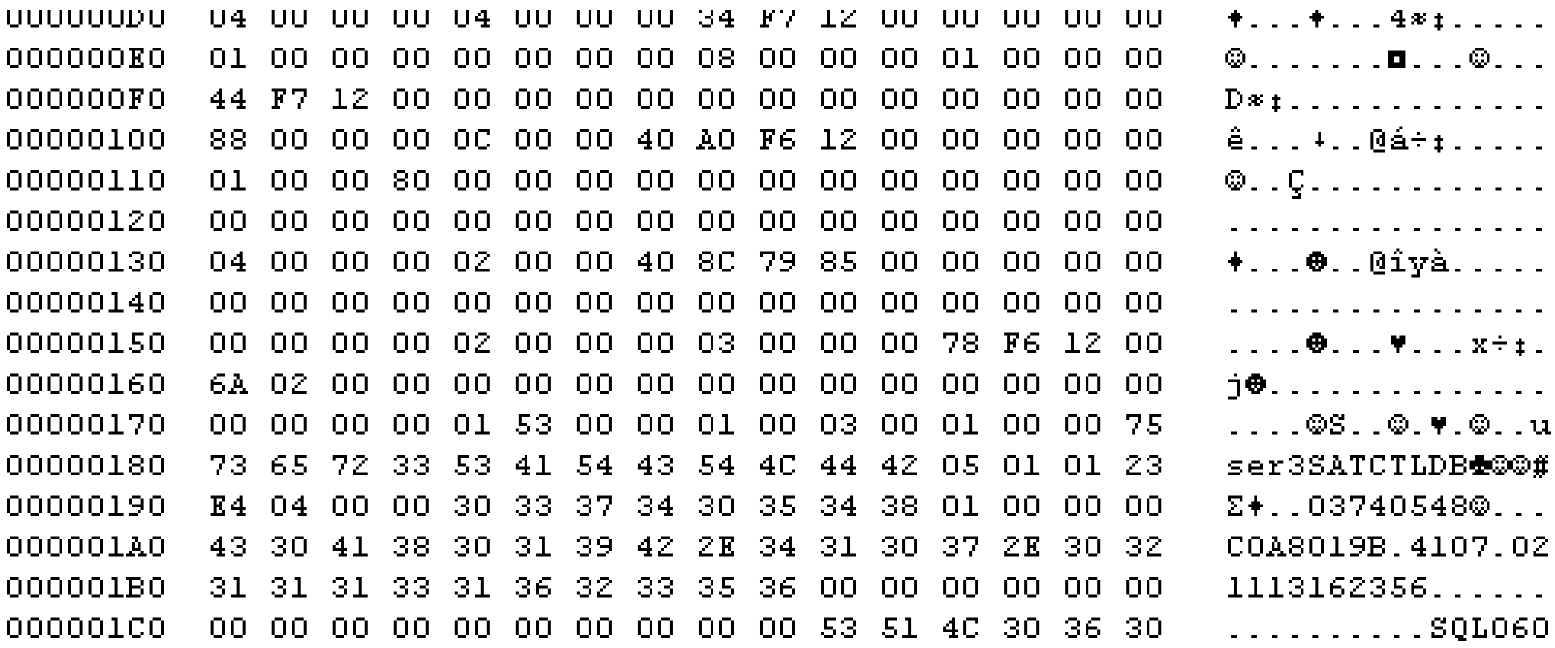
- Controlling where authentication occurs
- NEVER allow server to accept CLIENT authentication
- Allows client to authenticate the user (*trust_allclnts*)
  - inherently insecure
- Recommended settings
  - DCE_ENCRYPT
  - SERVER_ENCRYPT
  - KRB_SERVER_ENCRYPT
- Discouraged setting
  - CLIENT
  - SERVER

Encryption During Authentication

- Controlling passwords as they traverse the network
  - ALWAYS require *_ENCRYPT protocols
  - All passwords encrypted by DB2 at client before sent to 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   ♦...☻..@îyà.....
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♦..03740548☺...
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
```

APPLICATION SECURITY, INC.

www.AppSecInc.com

Default Username/Passwords

- Installed with the database
  - db2admin/db2admin
  - db2as/ibmdb2
  - dlfm/ibmdb2
  - db2inst1/ibmdb2, db2inst2/ibmdb2, etc…
  - db2fenc1/ibmdb2, db2fenc2/ibmdb2, etc…
- Others installed with 3rd party products
- Ensure you have enabled password management features
  - Password lockout to 10
  - Password expiration to 90 days

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

SetUID/SetGID Files

- Allows programs to be run under effective rights
- Program runs under file owners permissions
- Not under permissions of person executing
- Can often lead to elevated privileges for attackers
- Buffer overflows in files
- Files trust environmental variables
- Should remove all setuid/gid if not needed
- Particularly if file is owned by root

Locking down database privileges

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

Application Security, Inc.

Securing DB2 in a Web Application

Hack-proofing DB2

Can attacks go through a firewall?

- YES!!!
- Firewall configuration
  - Block access through port 523, 50000-5000?
  - 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?

- Modify the query
- Change:
  - Select * from my_table where column_x = ‘1’
- To:
  - Select * from my_table where column_x = ‘1’ UNION select salary from payroll 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!

Selecting from other Tables

- To select data other than the rows from the table being selected from.
- UNION the SQL Statement with other tables or views.

Sample ASP Page

```
Dim sql
Sql = “SELECT PRODUCTNAME FROM
PRODUCT WHERE ProductName=’” &
product_name & “’”
Set rs = Conn.OpenRecordset(sql)
‘ return the rows to the browser
```

Valid Input

- Set the product_name to :
  - DVD Player
- The SQL Statement is now:
  - SELECT PRODUCTNAME FROM PRODUCT WHERE ProductName=’DVD Player’

Hacker Input

- Set the product_name to :
  - test’ UNION select username, password from dba_users where ‘a’ = ‘a
- The SQL Statement is now:
  - SELECT PRODUCTNAME FROM PRODUCT WHERE ProductName=’test’ UNION select tabname from SYSCAT.TABLES where ‘a’=‘a’

Preventing SQL Injection

- Validate user input
- Parse field to escape single quotes to double quotes
- Bind variables – don’t concatenate SQL strings
- Right way
  - SET v_dynSQL = ‘UPDATE EMPLOYEE SET BONUS=? WHERE EMPNO=?’;
- Wrong way
  - SET v_dynSQL = ‘UPDATE EMPLOYEE SET BONUS=‘ || p_new_bonus || ‘ WHERE EMPNO=‘ || p_emp_no;

Application Security, Inc.

Database Vulnerabilities

Hack-proofing DB2

APPLICATION SECURITY, INC.

www.AppSecInc.com

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

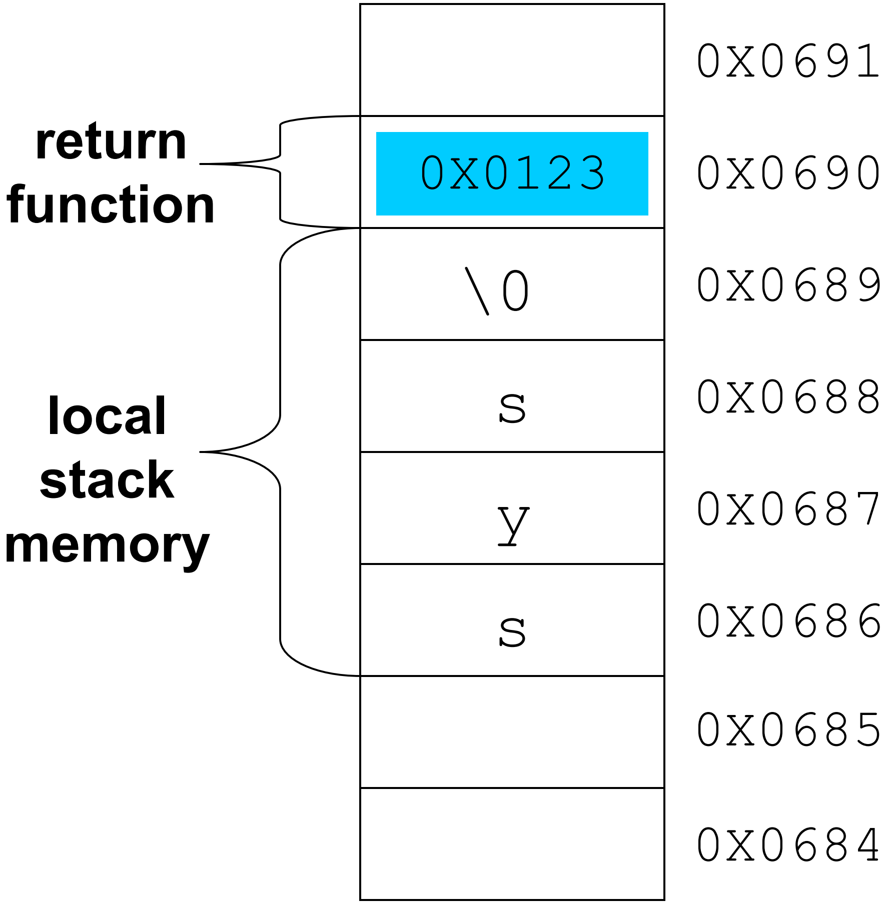
Stack is like a pile of plates

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



return function

local stack memory

	0X0691
0X0123	0X0690
\0	0X0689
s	0X0688
y	0X0687
s	0X0686
	0X0685
	0X0684

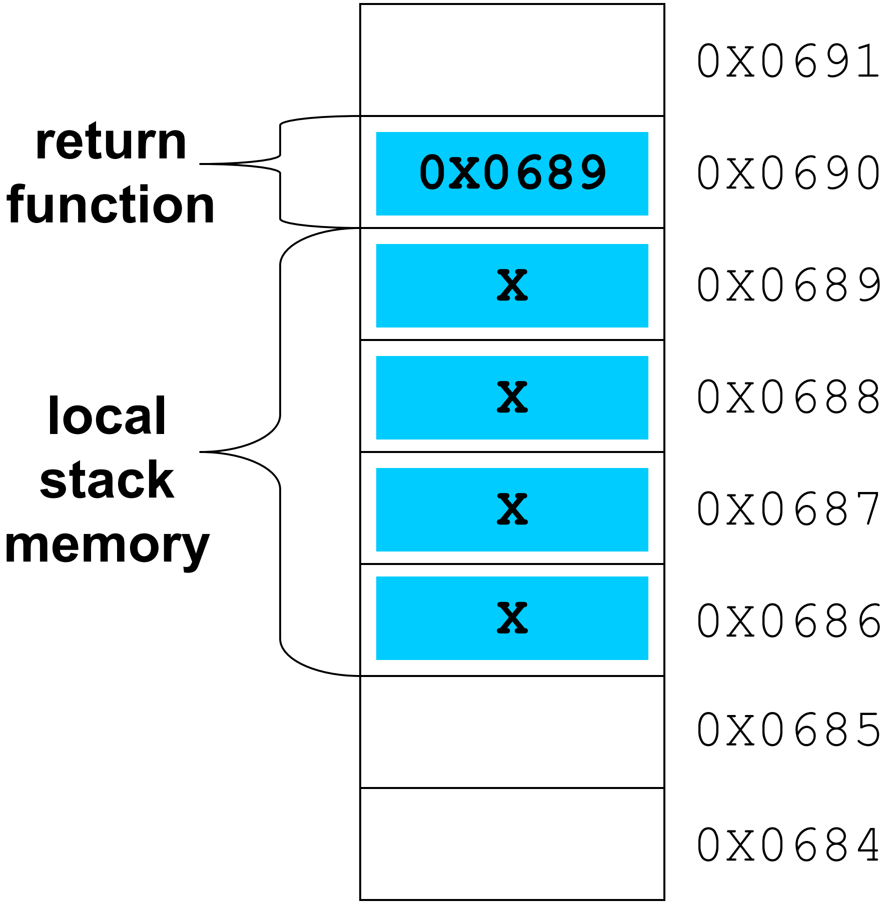
Mechanics of stack-based buffer overflow

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



Role of researchers and hackers

- Software vendors do not evaluate there own security correctly
- Historically all software riddled with security bugs
  - Everything from OS to database to custom apps
- Few vulns discovered in DB2 – good or bad?
  - BAD!
- Translates to the fact there are many holes known to hackers but not to the rest of the world
- Our best hope is security researchers perform independent reviews of software
  - Communicate found security holes to vendor and wait for a patch

Installing the latest FixPak

Usually addresses latest buffer overflows
Most current versions

- V6.1 – Fix Pak 11
- V7.1 – Upgrade to V7.2
- V7.2 – Fix Pak 10a
- V8.1 – Fix Pak 3

Download from

- http://www-4.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/download.d2w/

Query compiler Denial of Service

- The Query Compiler contained a bug
- SELECT CASE can crash database engine
-
  ```
  SELECT CASE WHEN EXISTS
  (SELECT * FROM LOCAL_TAB1 EXCEPT
  SELECT * FROM LOCAL_TAB2)
  THEN 1 ELSE 0 END
  FROM (VALUES 1) AS X
  ```
- Requires no special privileges in database
- Fixed in: DB2 version 7.2 FixPak 7

Date/Varchar Denial of Service

- Handling of the YEAR function crashes causing the database to stop
- Malicious SELECT statement to crash database
  - SELECT * FROM EMPLOYEE WHERE YEAR(BIRTHDATE)=1999 AND FIRSTNME<''
- Requires no special privileges in database
- Fixed in
  - DB2 version 6.1 FixPak 8
  - DB2 version 7.2 FixPak 3

Discovery service DoS

- Service allowing DB2 instances to be located
  - Runs on UDP port 523
- Used by Client configuration utility
- This service typically receives a packet such as "DB2GETADDR SQL07020".
- If it receives a packet with a length greater than 20 bytes, the service will crash.
- Fixed in
  - DB2 version 7.2 FixPak 10a

Local buffer overflows (Unix/Linux)

- Multiple buffer overflows can be executed by operating system users
  - db2dart, db2licm, db2start, db2ckpwd
- Various utilities for managing licensing, authentication, starting instances, etc…
- All files are created as SetUID, several with root
- Any local Unix user can become root
- /db2as/sqllib/adm/db2dart 'perl -e 'print "A"x1287"
- Exploit code exists in the wild
- Fixed in
  - DB2 version 7.2 FixPak 10a

LOAD buffer overflow

- Invoked from the DB2 Command Center or DB2 Command Line Processor (CLP)
- Must have a user ID in the database
- User does need privilege to execute LOAD
- Supplying an overly long string as an argument to the LOAD command will cause a buffer overflow
  - CONNECT TO SAMPLE
  - LOAD FROM testAAAAAAA[long string]
- Fixed in
  - DB2 version 7.2 FixPak 10a

INVOKE buffer overflow

- Invoked from the DB2 Command Center or DB2 Command Line Processor (CLP)
- Must have a user ID in the database
- No special privileges are required
- Supplying an overly long string as an argument to the INVOKE command will cause a buffer overflow in the db2dari executable
  - CONNECT TO SAMPLE
  - INVOKE vwploadr.exe\AAAAAAA[long string]
- Exploit code exists in the wild
- Fixed in
  - DB2 version 7.2 FixPak 10a

JDBC Applet Server buffer overflow

- JDBC Applet Server and Control Center
  - Runs on port 6789 and 6790
- Capability to remotely administer DB2
- Buffer overflow sending a large packet
- Fixed in
  - DB2 version 6.1 FixPak 10
  - DB2 version 7.2 FixPak 4
- Is important for both the client and the server!
- Someone can gain control of the client and pigback into a database

Overflows in CONNECT packet

- DB2 network protocol is proprietary
- When a connect packet sent is 523
  - Crashes when random bytes are changed
- Sending long string to port 523
  - Can cause the administration configuration in memory to be overwritten
- Some issues seem to be fixed in
- Fixed in DB2 version 8.1 Fixpak 3
- Not fixed in DB2 version 7.2 FixPak 10a
- Prediction – we will see more serious exploits coming out in the hacker community around this one in the next few months

Application Security, Inc.

Resources, Conclusion, and Wrap up

Hack-proofing DB2

How to Combat Hackers

- Stay patched –
  - http://www-4.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/download.d2w
- Security alerts:
  - www. appsecinc.com/resources/mailinglist.html
- Security Discussion Board
  - www. appsecinc.com/cgi-bin/ubb/ultimatebb.cgi
- Check out security solutions at:
  - www.appsecinc.com

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

Questions?

- About
  - Vulnerabilities
  - Protecting your database
- Email us at:

info@appsecinc.com

www.appsecinc.com