Outsourcing: Financial Dream or Security Nightmare?

Nebraska CERT 2007

Presented By:
Rohyt Belani

Intrepidus Group

- Information security consulting company
- Services include:
  - Application Security
  - Network Security
  - Mobile Security
- Located in Chantilly, VA & NYC
- Internationally acclaimed experts:
  - Presented at Black Hat, DefCon, Hack In The Box, OWASP
  - Written articles for SecurityFocus, SC Magazine
  - Quoted in Forbes, InformationWeek, Hacker Japan, BBC UK

Outsourcing: The Business Drivers

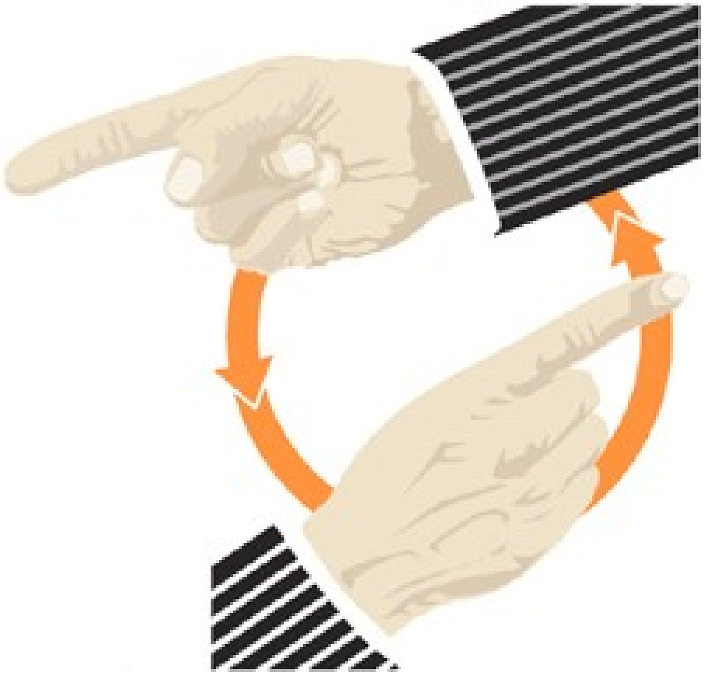
- □ Effective Cost Structure
- □ Strong Knowledge Base
- □ 24 X 7 Work Model

Some Perspective…

- □ 84% of (500) companies interviewed outsourced application development -- InformationWeek
- □ Outsourcing of enterprise applications growing at 7.3% annually – Gartner
- □ B2B and B2C applications are top candidates – CIO Insight

Security: Who’s Job Is It?

- There was an important job to be done
- Everybody was sure that Somebody would do it
- Anybody could have done it, but nobody did it
- Everybody thought that anybody could do it, but nobody realized that Everybody wouldn't do it.
- It ended up that everybody blamed somebody when nobody did what anybody could have done



As A Result…

- Recurring Vulnerabilities
- Higher Cost of Fixing Security Bugs
- Regulatory Violations
- Backdoors
- And Sour Relationships…

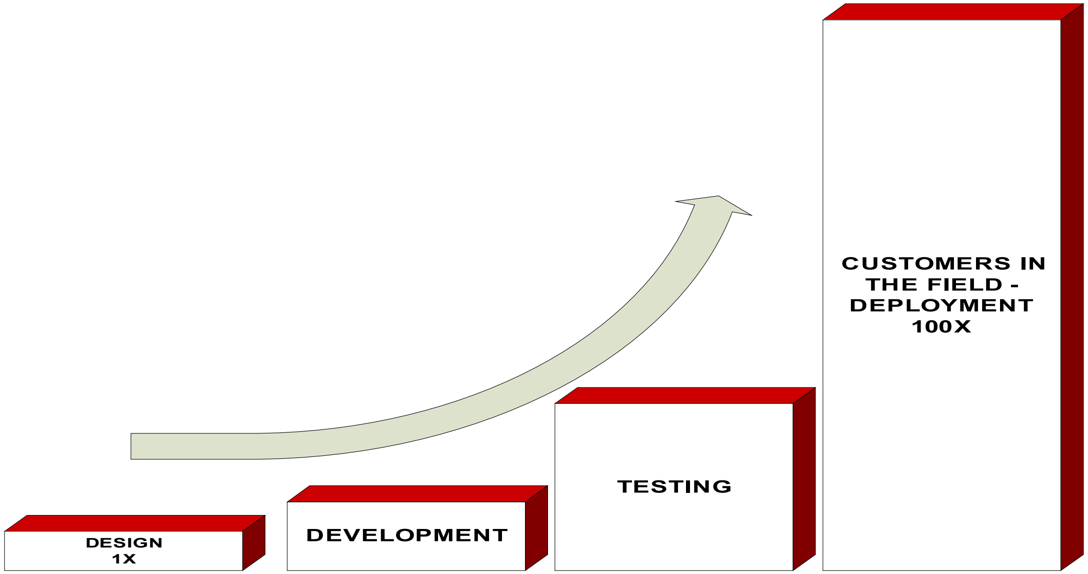
Recurring Vulnerabilities

Excerpt from a Quarterly Report for a Bank

Area of Assessment	Application 1	Application 2	Application 3	Application 4	Application 5
Server Vulnerabilities	X			X	
Authentication					
Authorization	X	X		X	X
Input Validation	X	X	X	X	X
Error Handling		X			
Session Management			X	X	

Cost of Fixing Security Bugs

Relative Costs to Repair Software Defects at Different Stages of the Software Development Lifecycle



Source: National Institute of Standards and Technology

Regulatory Requirements

- PCI
- California Senate Bill No. 1386
- GLBA
- PIPED
- EFTA
- FISMA

PCI Compliance

6.5 Develop all web applications based on secure coding guidelines such as the Open Web Application Security Project guidelines. Review custom application code to identify coding vulnerabilities.

6.6 Ensure that all web-facing applications are protected against known attacks by applying either of the following methods:

- Having all custom application code reviewed for common vulnerabilities by an organization that specializes in application security
- Installing an application layer firewall in front of web-facing applications.

Note: This method is considered a best practice until June 30, 2008, after which it becomes a requirement.

California Senate Bill No. 1386

- Application should ensure the security and confidentiality of customer records and information, Sec.2 and Sec.4
- The application must not disclose to a nonaffiliated party any nonpublic personal information, Sec.2 and Sec.4

GLBA

“Vendor management programs must include establishing security requirements, acceptance criterion, and test plans, [and] reviewing and testing source code for security vulnerabilities”

Source: Federal Financial Institutions Examination Council (FFIEC) Information Security Handbook

A Report from the Trenches

Symptoms

- The CEO of a retail organization received an extortion threat of $250,000 via snail mail
- The threat – 125,000 customer credit card numbers would be posted on the Internet
- The response was demanded in the form of a footer on the main page of the retailer’s website

Response

- 72 hours were granted by the extorter
- 3 investigators X 3 days
- Who compromised the data?

What Followed?

- Web server log analysis – Nothing!
- Employee email inboxes reviewed – Nothing!
- Database login/logout activity reviewed – nothing suspicious
- Web application scanned for SQL injection flaws – No luck!
- Last resort – application code review

Racing Against Time

- > 100,000 lines of code
- Comprehensive code review was ruled out
- Resorted to scripted searches through code

Scripted Searches

- □ Did the code contain raw SQL statements?
- □ Searched for occurrences of the "SELECT" in the code

Regex = `.*SELECT.*`

- □ The search resulted in an overwhelming number of hits

Scripted Searches

- □ Searched for occurrences of the "SELECT *" string to identify SQL statements where the scope was not properly limited

  Regex = `SELECT \*.*FROM.*`

- □ The search resulted in 5 hits
- □ One of the hits was:

```
SELECT * FROM CardTable
```

The Code That Made The Call

```
NameValueCollection coll = Request.QueryString;
String[] arr1 = coll.AllKeys;
...
String[] arr5 = coll.getValues(arr1[4]);
string extra =
   Server.HtmlEncode(arr5[0]).ToString();

if (extra.Equals(“letmein”))
{
   Cmd = “SELECT * FROM CardTable”;
}

...
```

Eureka!

- □ Backdoor – an insider job?
- □ Reviewed code archives to detect addition of code
- □ The first check-in with this code was made by a developer contracted from a third-party in Asia
- □ Reviewed web server logs for additional parameter
- □ Source IP traced back to Asia!

Another One Bites The Dust…

- Development company was notified of this rogue activity

- Local law enforcement was cooperative

Bridging the Security Divide

- □ SLAs & Legalities
- □ Building Security Into the SDLC
- □ Security Testing
- □ Post-Mortem Review to Identify Systemic Causes of Vulnerabilities

SLAs & Legalities

- □ Define and Classify Security Vulnerabilities
- □ Document Security Requirements
- □ Require Detailed Documentation of Security Design
- □ Define Acceptance Criteria
- □ Require Security Aware/Trained Developers
- □ Security Maintenance

The push must come from the client!

Who Foots The Bill?

Client

- Must be willing to accept the extra line item in the bill. Yes, security is a value add!

Software Development Firm

- Hire security architects
- Train developers
- Build security into the SDLC

Building Security Into The SDLC

- Think security from the word go
- Assign a Risk Rating to the project
- Map out Regulatory Requirements to technical requirements
- Document Security Requirements
- Perform Threat Analysis during the design phase
- Perform Security Architecture Review
- Code Secure Software
- Test, Test, Test!

Security Testing..Trust, But Verify

- Review Source Code
  - Check for logic flaws
  - Check for back-end issues e.g. encryption of data
  - Check for backdoors!
- Penetration Testing
- Ensure the risk is below an acceptable level

Conclusion

- Drive towards outsourced development makes testing for security even more important
- The client need to ensure that all outsourcers are complying with your desired security requirements
- Build security requirements into SLAs
- Validate security before acceptance
- Development companies should view security as a competitive advantage…

Now I'm getting a little carried away

Thank You

www.intrepidusgroup.com

rohyt.belani@intrepidusgroup.com