

» Software Security Today & The “OWASP”

“Web Application & Web Services Security in the Real World”

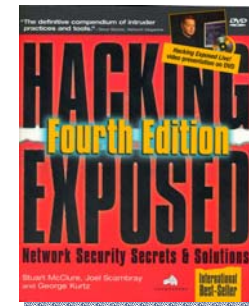
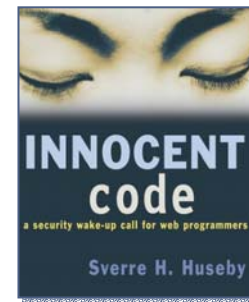


Contents:

- » What's the Scope of the Problem?
- » Costs & Industry Observations
- » How Do You Build Secure Software?
- » Notes From the Field
- » OWASP

Speaker Bio – Mark Curphey, Foundstone Inc.

- » Academic
 - Masters Degree in Information Security (Royal Holloway, University of London) - Cryptographer
- » Experience
 - Foundstone - Director of S3i (Boston / California)
 - Chares Schwab - Information Security Director (Application Security)
 - ISS - Consulting Manager
 - ING Bank, Dresdner Bank, EBRD, UK Gov and NATO, (Europe)
- » Industry
 - OWASP - Founder and Chair (<http://www.owasp.org>)
 - Book Forewords
 - New Book 2004 / 2005 – “Building Security into the Software Development Life Cycle”
- » Software Security Interests
 - Metrics and Measurement
 - Modeling Security with UML
 - .NET Security and the C# Language
 - Software Process Management



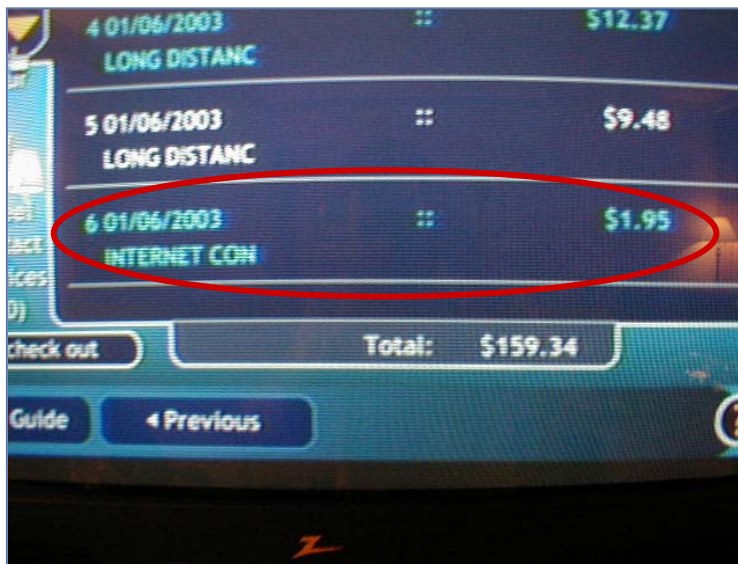
Preview

- » What Is The Scope Of The Problem?
- » Costs and Industry Observations
- » How Do You Build Secure Software?
- » Notes From the Field
- » OWASP – The Open Web Application Security Project

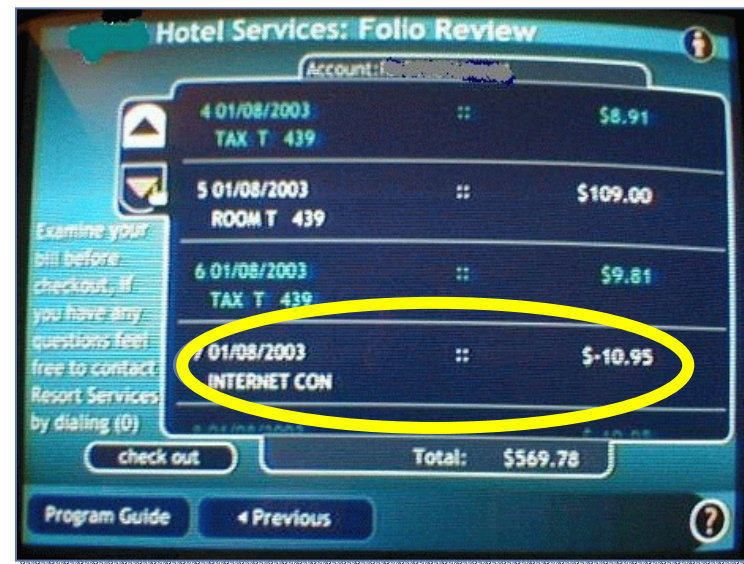
OWASP Top Ten

- » **Unvalidated Input**
- » Broken Access Control
- » Broken Authentication and **Session Management**
- » Cross Site Scripting (XSS) Flaws
- » Buffer Overflows
- » **Injection Flaws**
- » Improper Error Handling
- » Insecure Storage
- » Denial of Service
- » **Insecure Configuration Management**

Parameter Manipulation



Before



After

Unvalidated Input

The Negative Values Are Not Checked

A different version of "How to be a Millionaire"!

Address https://www.mcafee.com/usa/.../MCAfee.../...

Update The Following/Remove Items From Your Cart.
To Update an item, change the quantity or options and click update.
Click delete to remove an item from the cart.

Qty	Part No	Name	Options	Price	Total	Update
<input type="text" value="-20000"/>				\$39.98	\$-799,600.00	<input type="button" value="Update"/>
<input type="text" value="-5000000"/>	006B	The Following Items Have Not Been Added		\$39.98	\$-199,900,000.00	<input type="button" value="Update"/>

TOTALS

Sub Total **\$-200,699,600.00**

Shipping Type	Shipping	Total
2-3 Day USPS	\$0.00	\$-200,699,600.00

Broken Authentication and Session Management

» Session Management Example

Time based with randomly incremented number appended

- EE51091718351065
 - EE51091718351703
 - EE51091718352354
 - EE51091718352411
-
- » Keys created on 09/17 at 6:35 PM, EST
 - » 10,000 possibilities for 20 minute window

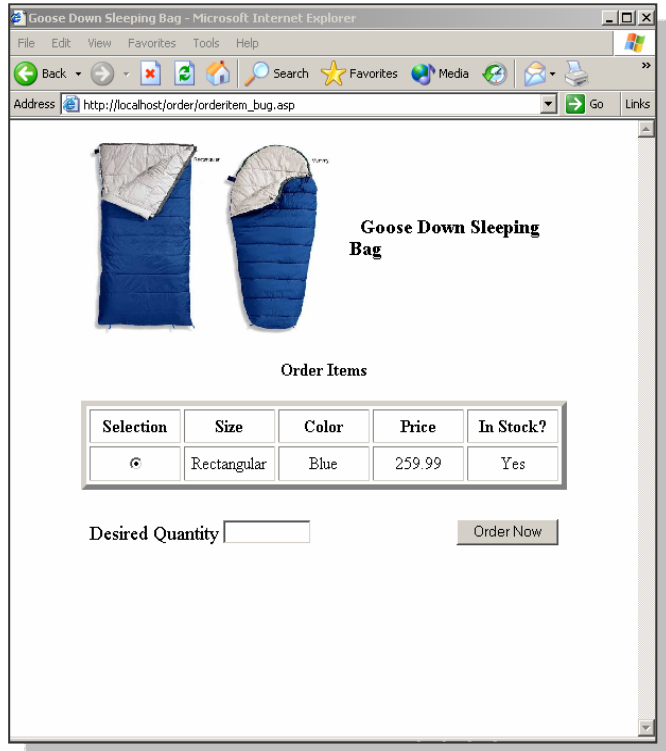
Injection Flaws

» SQL Injection Example

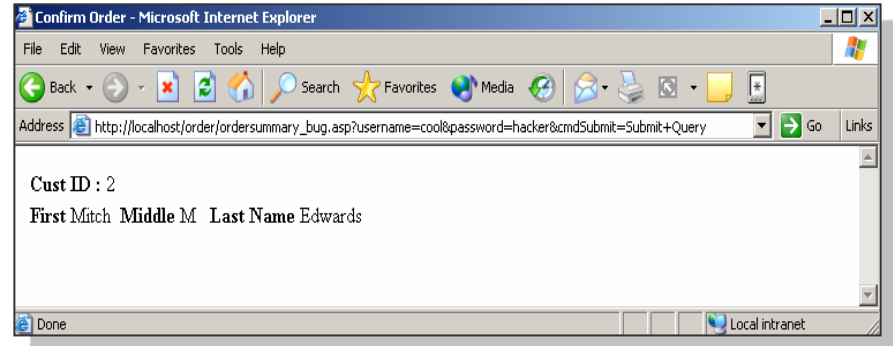
```
http://www.site/balance.asp?account_id=755+OR+1=1;--  
SELECT * FROM bankacct WHERE userID=755 OR 1=1;--;
```

- » This would return all rows from the table
- » Note: Whether or not the data would be displayed depends on the rest of the code
- » Often Attackers Will Use Core Database Functionality like xp_cmdshell to Launch Attacks

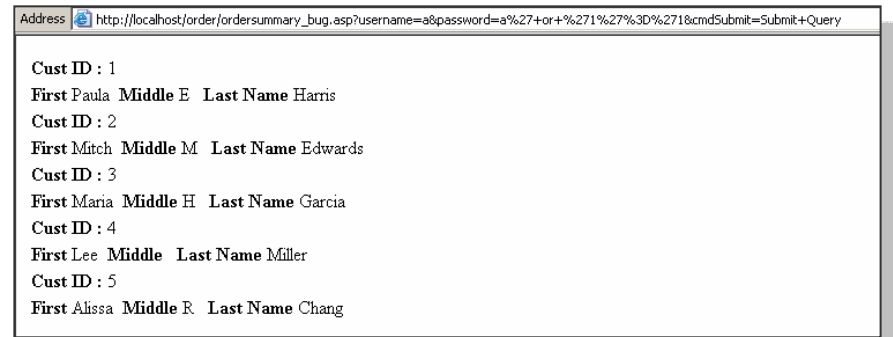
SQL Injection



Step 1



Step 2



Step 3

Remote Administration



» Software Security Today & The “OWASP”

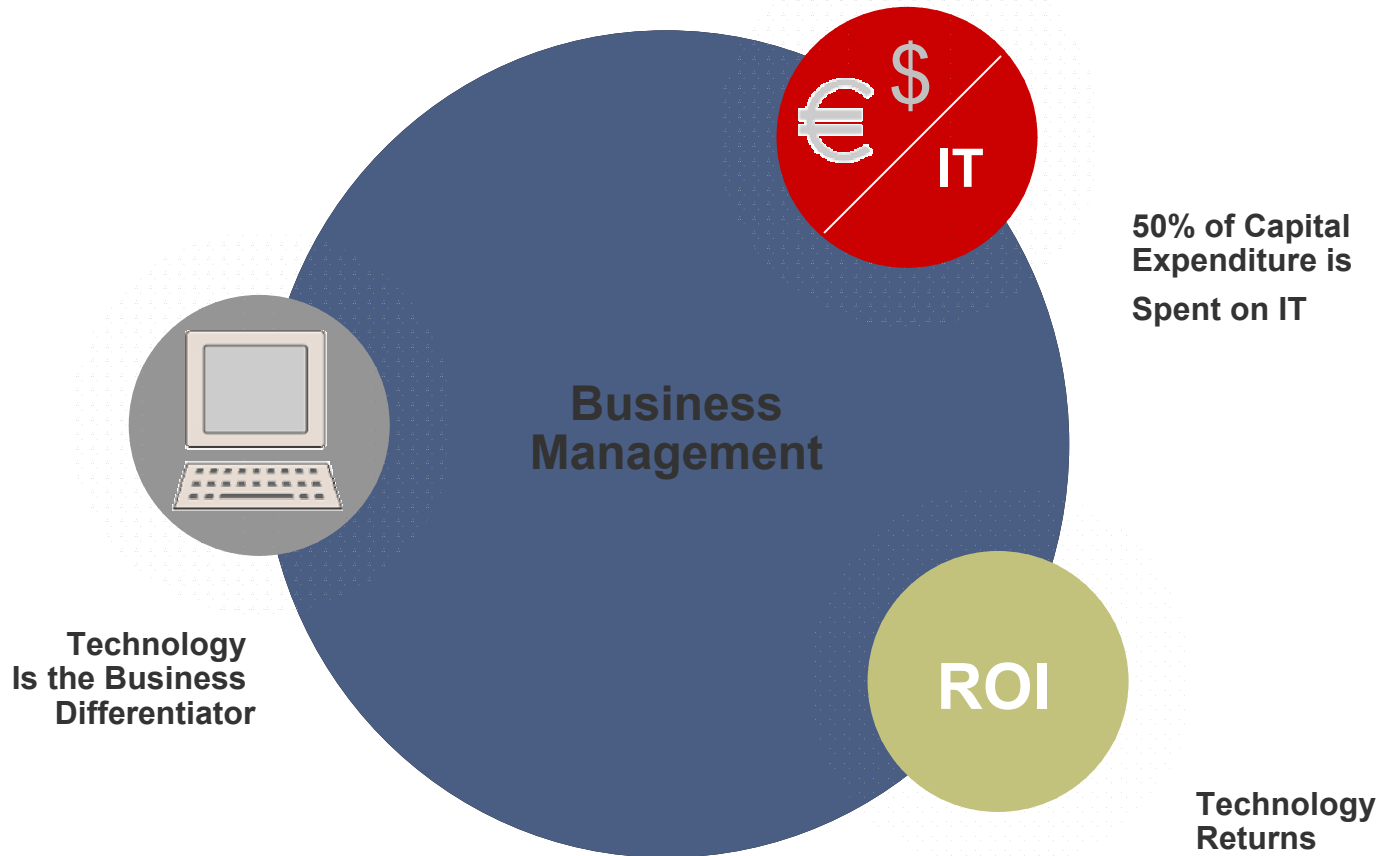
“Web Application & Web Services Security in the Real World”



Contents:

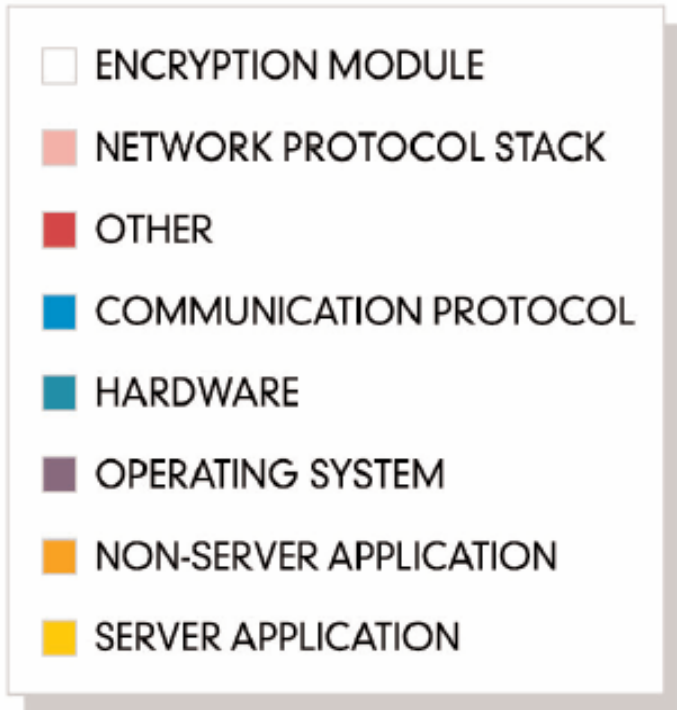
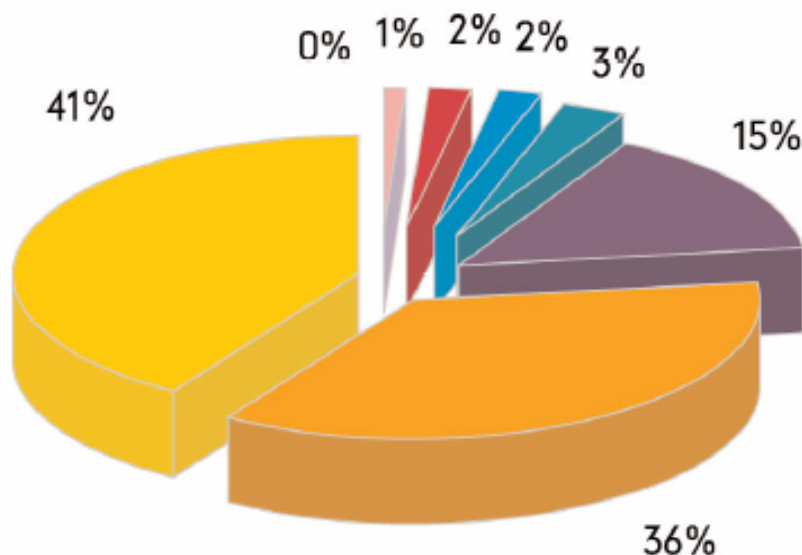
- » What's the Scope of the Problem?
- » **Costs & Industry Observations**
- » How Do You Build Secure Software?
- » Notes From the Field
- » OWASP

The Business Landscape



How Many Vulnerabilities Are Application Security Related?

92% of reported vulnerabilities are in applications, not networks



SOURCE: NIST

How Much Does This Cost?

76%

of Software and Applications Tested Have Serious
Design and Implementation Flaws
- Foundstone Survey

\$60B

in Cost to USA Economy from Poor Software Quality
-US Dept of Commerce

\$3B

Cost of Insecure Software to the Financial Services
Industry
- NIST Survey 2002

100x

100 Times More Expensive to Fix Security Bug at
Production Than Design
- IBM Systems Sciences Institute

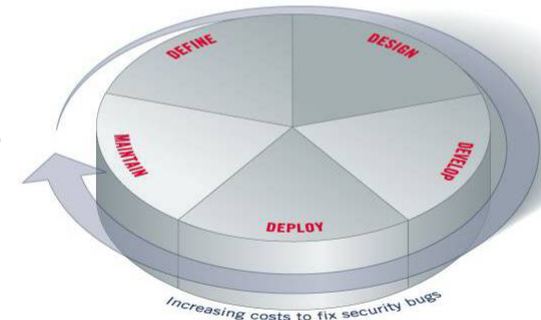
Costs of Insecure Software

- » NIST Survey
 - Software Errors Cost U.S. Economy \$59.5 Billion Annually
 - Cost Financial Services Industry \$3.8 Billion Annually
- » IBM Survey
 - Design and Architecture – Best Time To Address Security
 - Development – Costs 6 Times More Than Design
 - Production – Costs **100** Times More than Design
- » Incident Handling
 - Cross Site Scripting - \$140K (financial services site having been made public)
 - Denial of Service - \$1.8 Million (e-commerce (computer e-tailer))
- » Power Blackouts !

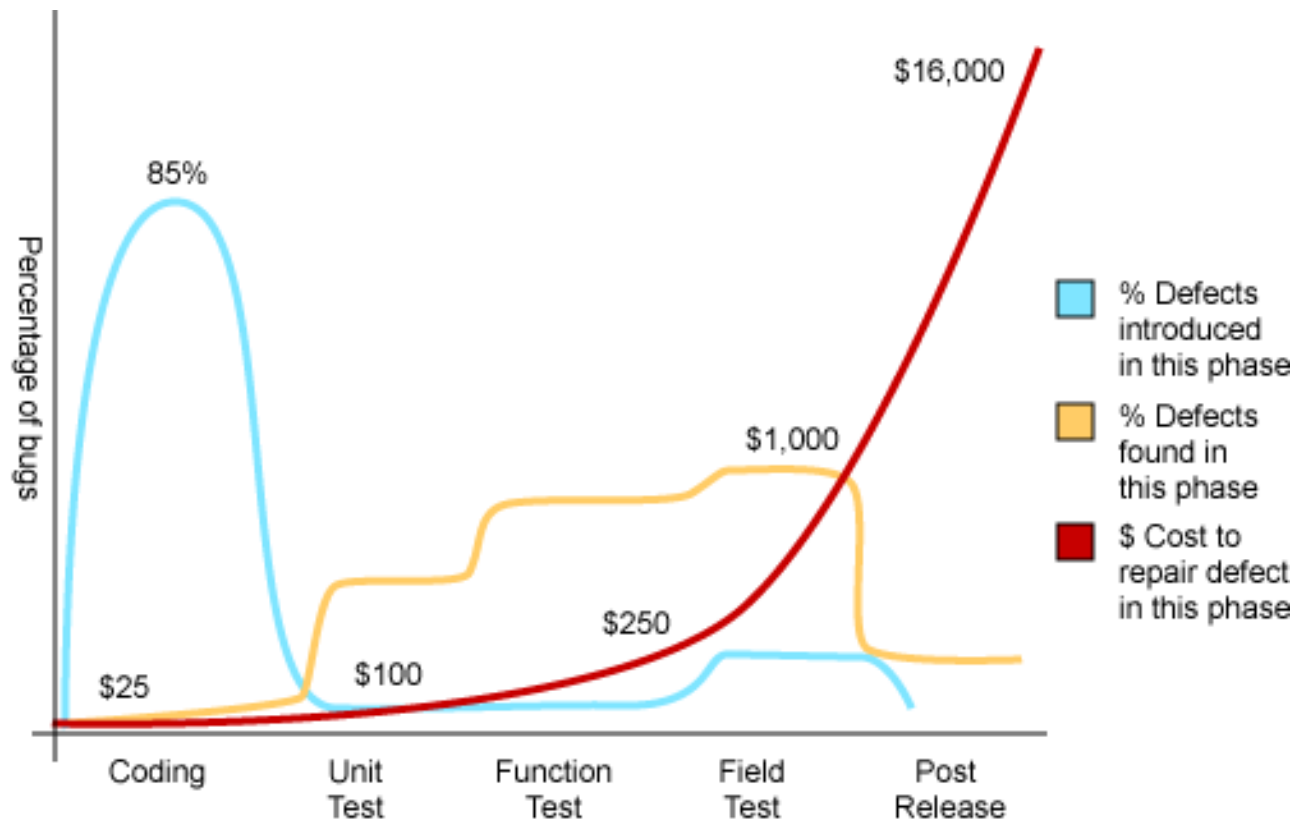
IBM System Sciences Institute Survey

Unit Cost to fix a software security hole in the deployment phase of an SDLC is 100 x that of fixing the same hole at the design stage

Exponential cost the later you catch (leave) it!



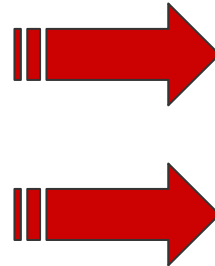
Cost of Fixing Defects



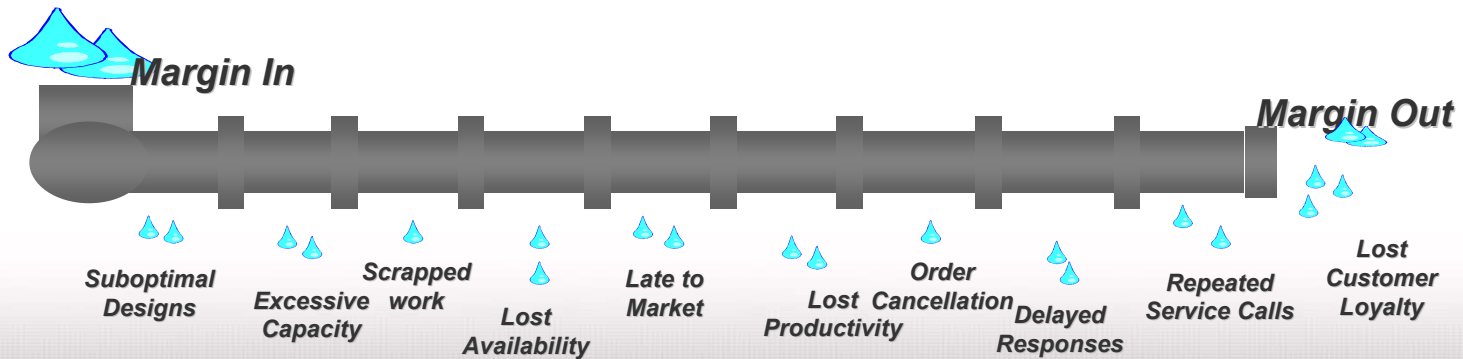
Source: *Applied Software Measurement*, Capers Jones, 1996

Observations of the Industry (Business Perspective)

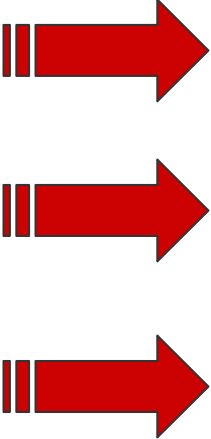
- » Sailing with the Wind
- » Time is the Enemy
- » Growth at All Costs
- » Revolutionary Offers
- » Horizontal for Breadth
- » Geographical Coverage
- » Catching the Next Wave

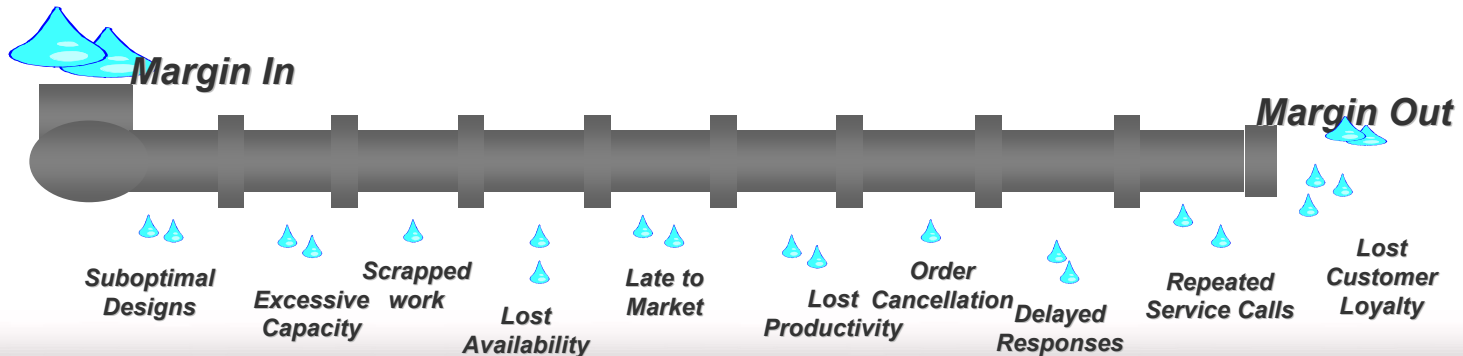


- » Sailing into the wind
- » Waste is the Enemy
- » Cash Flow at All Costs
- » Evolutionary Offers
- » Vertical for Depth
- » Domain Expertise
- » Fixing the Leaky Pipe



Observations of the Industry (Technical Perspective)

- | | | |
|--|--|---|
| <ul style="list-style-type: none"> » Development Frameworks <ul style="list-style-type: none"> - "Safe" Languages - Architectural Advances (CLR) - Security Features » New Technology <ul style="list-style-type: none"> - App Scanners - App IDS » Security Standards <ul style="list-style-type: none"> - Languages - Architectures - Techniques |  | <ul style="list-style-type: none"> » Development Frameworks <ul style="list-style-type: none"> - Complexity - Lack of Understanding (Not Education) - Not Being Used ! » New Products <ul style="list-style-type: none"> - Sold as Silver Bullets - Point Solutions - Don't Scale, Immature - Too Late in SDLC » Security Process <ul style="list-style-type: none"> - No Security in the RUP Yet ! - Vendors Setting Standards Doesn't Work |
|--|--|---|



If cars were built like applications....

1. 70% of all cars would be built without following the original designs and blueprints. The other 30% would not have designs.
2. Car design would assume that safety is a function of road design and that all drivers were considerate, sober and expert drivers.
3. Cars would have no airbags, mirrors, seat belts, doors, roll-bars, side-impact bars, or locks, because no-one had asked for them. But they *would* all have at least six cup holders.
4. Not all the components would be bolted together securely and many of them would not be built to tolerate even the slightest abuse.
5. Safety tests would assume frontal impact only. Cars would not be roll tested, or tested for stability in emergency maneuvers, brake effectiveness, side impact and resistance to theft.
6. Many safety features originally included might be removed before the car was completed, because they might adversely impact performance.

If cars were built like applications....

7. 70% of all cars would be subject to monthly recalls to add major components left out of the initial production. The other 30% wouldn't be recalled, because no-one would sue anyway.
8. The after-market for safety devices would include such useful products as training wheels, screen doors, elastic seatbelts and devices that would restrict the car's top speed to 3mph, if found to be unsafe (which would be always).
9. Useful safety could be found, but could only be custom retro-fitted, would take six months to fit and would cost more than the car itself.
10. A DOT inspection would consist of counting the wheels and making recommendations on wheel quantity.
11. Your only warning indicator would be large quantities of smoke and flame in the cab.
12. You could only get insurance from one provider, it would be extremely expensive, require a duplicate DOT inspection, and you might still never be able to claim against the policy.

» Software Security Today & The “OWASP”

“Web Application & Web Services Security in the Real World”



Contents:

- » What's the Scope of the Problem?
- » Costs & Industry Observations
- » **How Do You Build Secure Software?**
- » Notes From the Field
- » OWASP

How Do You Build Better Software?

- » You Build Better People
- » **You Build Better Process**
- » You Build Better Technology

Why Create a Security Enhanced SDLC?

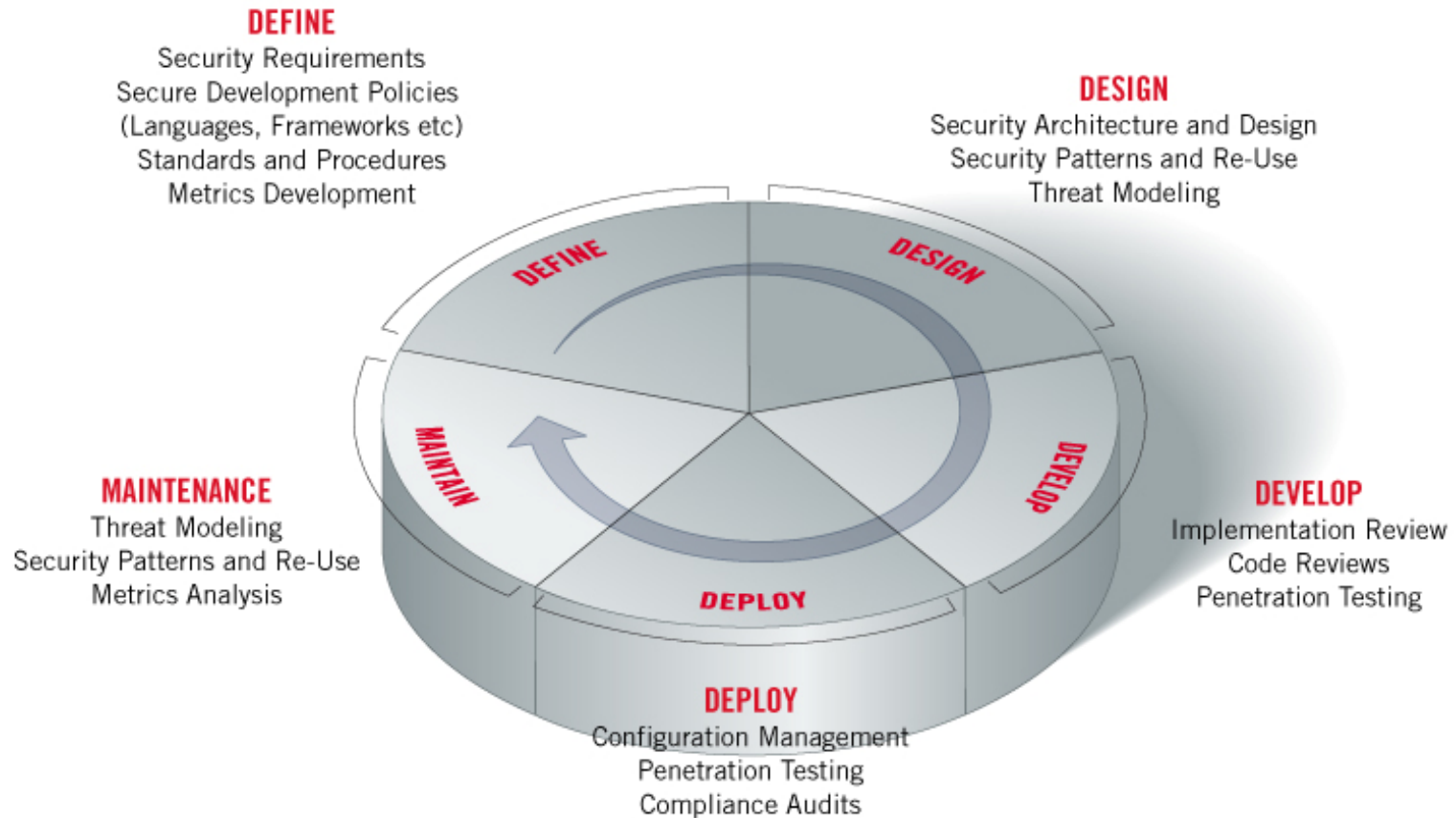
- » Risk management
 - Risk avoidance
 - Risk acceptance
- » Reliability
 - Secure software is predictable in the “face of danger”
 - Fail safe / fail closed
 - Survivability
- » Reduce costs
 - Avoid costly failures
 - Reuse components
 - Improve process efficiency
- » Deal with root causes
- » Deal with problems strategically
- » Continuous improvement through measurement

What Do We Mean By A “Security Enhanced” SDLC?

- » What we don't mean...
 - A scanning tool plugged into an IDE!
 - Pen testing just before deployment

- » What we do mean...
 - People
 - Process
 - Technology

Solutions and Techniques



Education and Awareness

- » Why are education and awareness so important?
 - Help people do the right thing
 - Publish rules if you want to enforce rules
 - Cost effective
 - Builds self sufficiency
- » Executive education
 - **Why** (economics of insecure software)
- » Project management education
 - **Why, what, where, when and how**
- » Developer education
 - Why, what, where, when and **how**
- » Operational education
 - Why, what, where, when and **how**

Education Tips

- » Executive education
 - Include examples of disasters
 - Relate to cost and benefits
- » Project management education
 - Relate to existing project management process
 - Focus on advantages to PM office
- » Developer education
 - Make relevant (e.g. Java to Java developers)
 - Use examples
 - Make it hands-on (e.g. Hacme Bank™)
 - Keep it short
- » Miscellaneous
 - Use existing delivery mechanisms
 - Personalize it
 - Measure it

Threat Modeling

- » Purpose
 - Identify where to spend security resources
 - What would a Hacker do?
 - Do you have the right countermeasures?
- » Methodologies
 - STRIDE (MS)
 - DREAD
 - OCTAVE
 - Your Own !

Threat Modeling – Cont.

- » MS Approach
 - Step 1. Identify Assets
 - Step 2. Create an Architecture Overview
 - Step 3. Decompose the Application
 - Step 4. Identify the Threats
 - Step 5. Document the Threats
 - Step 6. Rate the Threats
- » Tools
 - Flip Chart and a Scribe
 - MS Free Tool
 - <http://msdn.microsoft.com/security/securecode/threatmodeling/default.aspx>

Metrics and Measurement

- » Measurement
 - Abstract Things – defects in process
 - Non-Abstract Things – defects in software
- » Measurements Have Units
 - Number of defects per 1,000 lines of code
 - % of questions correct in a test
- » Metrics
 - Defined criteria for measurement
 - Rate of improvement
 - Range of acceptable security
- » Score Cards
 - Custom Data Mining
 - Visibility
 - Six Sigma

Metrics and Measurement Tips

- » Classification
 - Recording measurements requires good (well thought out) classification schemes
 - OASIS WAS
- » Be Realistic
 - Don't over burden users with paperwork
 - Make it easy to do the right thing
- » Point of Capture
 - Record the measurement at the source where possible
- » What Will You Do With Data?
 - What you want to do with the data should drive what you capture
 - Archive - trends are very powerful!
- » Mechanisms
 - Various capture mechanisms probably already exist (with minor modifications)

Policy and Standards

- » Software Security Policy
 - Authn, Authz, Crypto
- » Language Standards
 - C, C++, Java, C#, XML, HTML
- » Framework Standards
 - .NET, J2EE
- » Technology Specific
 - Web
 - Client
- » Configuration Management
 - Application Server
 - Web Server
 - Database
 - Directory Services
 - Source Code Management

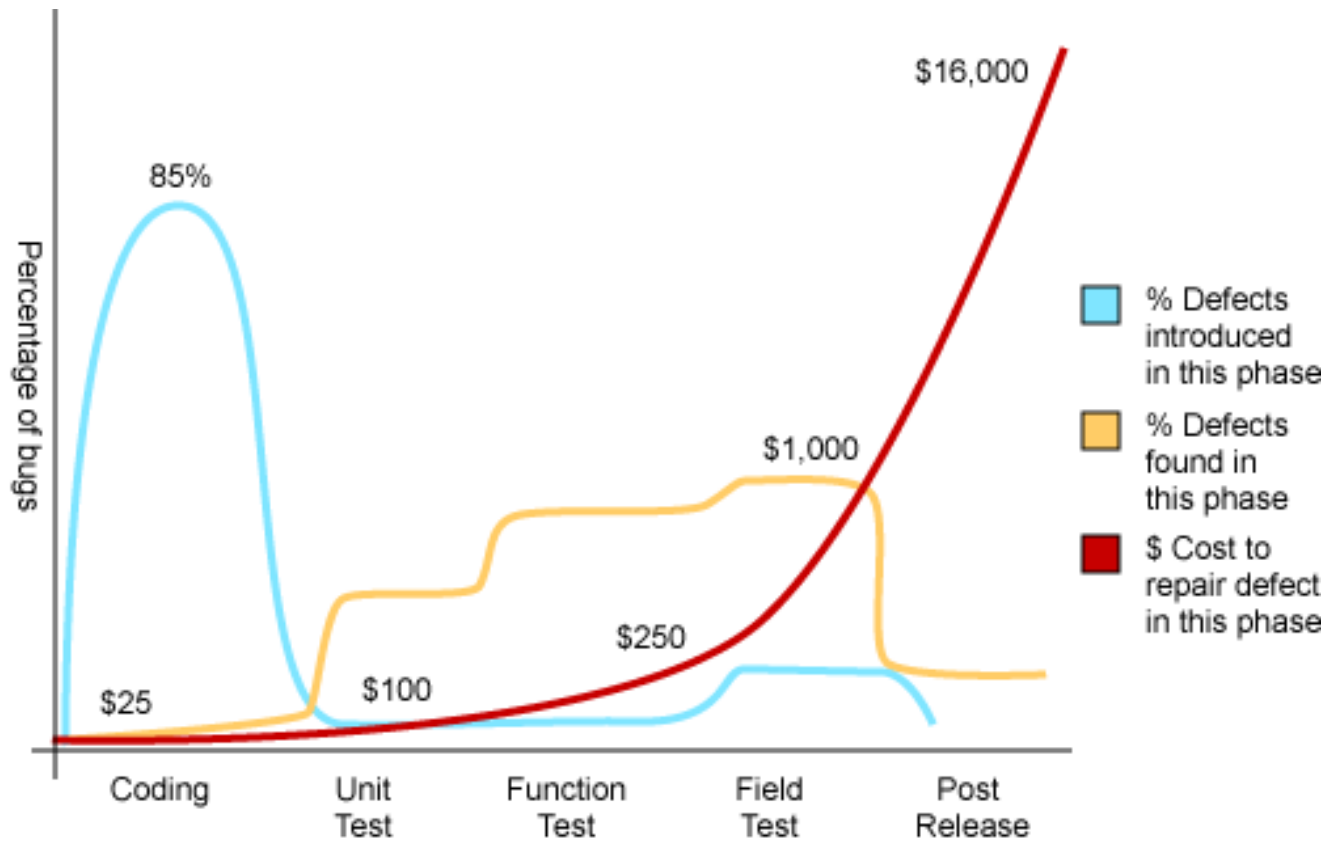
Patterns, Architectures and Reuse

- » Don't Reinvent the Wheel
 - Many patterns and architectures were designed with security in mind
 - Model View Controller
- » Architecture Rules !
 - Alleviates the need for a developer to re-invent it
 - Strategic solutions
 - Scalable
- » Reuse
 - Economies of Scale
 - Get it Right Once, Use It Right Many Times

Testing

- » Manual Inspections
 - Interviews
 - Documentation Review
 - Observance
- » Code Review
 - Static Code Review
 - Contextual Analysis
 - Run-Time Analysis
- » Penetration Testing
 - Not Very Effective for Software Security
 - Late in SDLC

Cost of Fixing Defects



Source: *Applied Software Measurement*, Capers Jones, 1996

Common Misconceptions When Building a Testing Program

- » *“We use penetration testing and automated scanners so we have it covered”.*
 - If you fail a penetration test you know you have a really, really bad problem. If you pass a penetration test you do not know that you don't have a really bad problem.
 - Best application scanner finds < 20% of web application security holes

- » *“We have an application firewall so we don't need to test for those sorts of holes”.*
 - Don't understand the business logic

- » *“We test everything before it goes live”.*
 - Dramatic cost implications
 - Usually implies black-box testing

- » **Key Message: In order to build better software, you have to build a better software development process.**

» Software Security Today & The “OWASP”

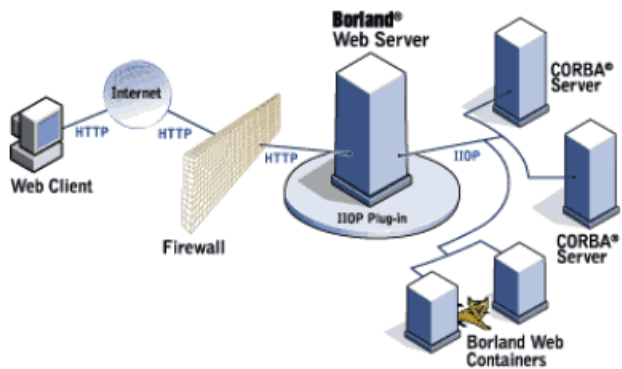
“Web Application & Web Services Security in the Real World”



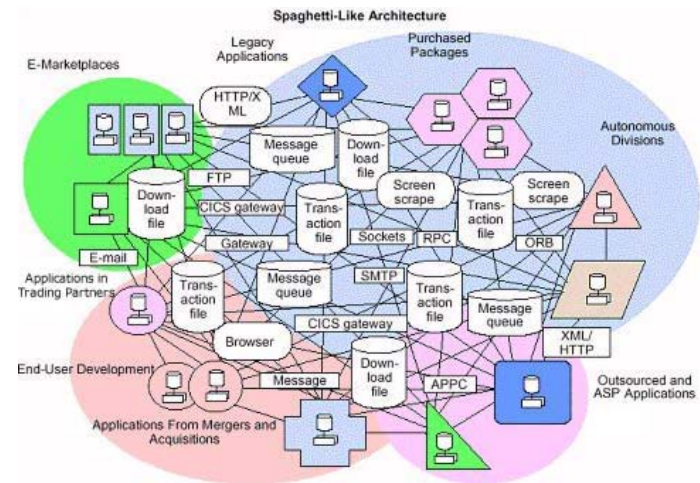
Contents:

- » What's the Scope of the Problem?
- » Costs & Industry Observations
- » How Do You Build Secure Software?
- » **Notes From the Field**
- » OWASP

Modern E-Commerce Architectures



Idealized



Real World

Top Challenges Facing Organizations Today?

- » **Lack of awareness**
 - Media (recent Network Computing, eWeek, SecurityWire)
 - Network security companies / consultants
- » **Lack of time**
 - Time to market is king
 - Security time is rarely accounted for
- » **Lack of skill**
 - Security people are not traditionally developers
 - Traditional security techniques don't always translate
- » **Lack of quantifiable data**
 - Return on security investment
 - Business advantage through security
 - What are our competitors doing?
- » **Lack of direction**
 - What should we do?
 - How can we do it?
- » **Lack of money**
 - Perceived need to spend money on fancy security technology
- » If you get it wrong or you “dabble”, software security will cost you money and get in the way of your development process. However
- » If you get it right, software security will make your business more reliable and predictable and save money.

Top Tips

- » Think Strategically
- » There Are No Silver Bullets
- » The SDLC is King
- » Education is a Very Powerful Tool
- » Don't Reinvent the Wheel
- » The Devil is Often in the Details
- » Develop Metrics and Measure Improvement
- » Make it Easy to do the Right Thing, Hard to do the Wrong Thing
- » Be Patient (Culture Doesn't Change Overnight)

Ounce Labs – Prexis / Insight

The screenshot displays the Ounce Labs Prexis/Pro interface. The top navigation bar includes 'HOME', 'DETAILS REPORT', 'PREFERENCES', and 'LOGOUT'. A 'HELP' button is also present. The left sidebar shows a 'Navigation Tree' with options like 'Curphey', 'Manual', 'Nessus', and 'Nessus New'. The main content area shows project details for 'nessuswx', including the assessment date (2004-Jun-02 14:04:31), configuration (Default), mode (Quick), and language (C and C++). Below this, the base directory is listed as 'c:\documents and settings\administrator.nylab\desktop\foundstone\nessuswx\'. A table follows, detailing vulnerability metrics for various files.

Filename	V-Density	# Vulnerabilities			Lines of Code
		High	Medium	Low	
cmdline.cpp	177.52	3	0	0	169
connect.cpp	179.74	20	31	30	1,402
crypto.cpp	1,861.76	7	3	2	136
csv.cpp	323.96	3	1	1	96
dbcheck.cpp	1,432.87	51	20	26	937
dbload.cpp	210.79	7	1	13	343
diff.cpp	669.56	18	2	9	542
enx.cpp	701.24	11	3	6	162
exception.cpp	1,225.00	5	0	3	188
execute.cpp	359.90	44	12	22	1,262

- Static Analysis
- Contextual Analysis
- Metrics Reporting

<http://www.ouncelabs.com>

» Software Security Today & The “OWASP”

“Web Application & Web Services Security in the Real World”



Contents:

- » What's the Scope of the Problem?
- » Costs & Industry Observations
- » How Do You Build Secure Software?
- » Notes From the Field
- » **OWASP**

© 2007 Foundstone, Inc. All rights reserved. Foundstone, the Foundstone logo, and OWASP are trademarks of Foundstone, Inc. McAfee is a trademark of McAfee, Inc.

About the Open Web Application Security Project

- » Founded in Sept 2000
- » Mission: Dedicated to Sharing Knowledge and Building Open Source Software Relating to Web Application Security
- » All Work Copyrighted to the Free Software Foundation and Released Under Approved Open Source Licenses

OWASP Foundation

Steering Committee



Documentation Projects

- » OWASP Guide (Version 2.0 Due Early 2005)
- » OWASP Top Ten
- » ISO17799
- » OWASP Testing (Part 1)
- » AppSec FAQ

Development Projects

- » oPortal
- » ANSA
- » WebScarab / Beretta
- » OCL
- » VulnXML Database
- » WebGoat
- » .NET Projects

Review

- » What Is The Scope Of The Problem?
- » Costs and Industry Observations
- » How Do You Build Secure Software?
- » Notes From the Field
- » OWASP – The Open Web Application Security Project

» **Thanks for Listening**



**Questions ?
Discussions ?**

Mark Curphey

mark.curphey@foundstone.com

www.foundstone.com