Web 2.0, Consumerization, and Application Security

Chenxi Wang, Ph.D.
Principal Analyst
Forrester Research
OWASP, New York City
September 25, 2008
Today’s enterprises face multitude of challenges

- Business-driven Model (ROI everything)
- Compliance
- Virtualization
- Deperimeterization & open architecture
- Consumerization

Consumerization triggers business changes
Consumerization

- Is at the forefront of innovation
- Fosters new business models
- Promotes new social structures
- have become part of enterprise fabric
Three Lenses to View Consumerization

- Adaptive Web
  - Social Computing
  - Social Networks
  - Software as a Service
  - Wikis
  - Flash/Flex
  - AJAX
  - XML
  - Microformats
  - Open APIs
  - Widgets
  - RSS
  - Blogs
  - Tagging
  - Podcasts
  - Enterprise 2.0
  - Rich Internet Applications
  - Information Workplace

- Crowd Sourcing
  - Enabling Technologies
  - Core Applications and Features
  - Behavioral Shifts

- Collective Intelligence
Consumer technologies foster new business

Example: Random House book widget

• Provides top-selling books in your ZIP code

• New viral distribution channel

• Creates new business opportunities

Click to read
Title & author
Browse
Order

Entire contents © 2008 Forrester Research, Inc. All rights reserved.
Collaboration is part of enterprise strategy

“Which of the following are likely to be one of your IT organization’s major software technology initiatives for the next 12 months?”

Implement an enterprise collaboration strategy

- Not on our agenda: 1%
- Not a priority: 17%
- Priority: 34%
- Critical priority: 33%
- Don't know or does not apply to me: 15%

Collaboration is priority for 50% of organizations

Near 50% of businesses view it as a priority

Base: 2,252 Software IT decision-makers at North American and European companies
Source: Forrester Enterprise And SMB Software Survey, Q3, 2007
Many are building serious applications on top of consumer technologies

- Crew portal (mission-critical app)
- Compliance documentation management

- AMD Central intranet
- Global partner sites (mission-critical)

- “Trusted workplace” for multi-company collaboration
- Sharing of critical design documents (airplane design docs, CAD drawings)
This is how a distributed team could work . . .
This is how a distributed team could work . . .
Perhaps this is what happens . . .
Consumerization accentuates every aspect of deperimeterization
Consumerization accentuates every aspect of deperimeterization
But, organizations are wary of consumer technologies

- Greater number of security risks
  - Attackers are targeting collaboration and Web 2.0 applications.
  - Staying ahead of all the issues is difficult.
- Increased complexity of IT operations and management
- Further dilution of the enterprise boundary
Web 2.0 Security risks
Internet is becoming more dangerous . . .

Source: "All Your IFRAMEs Point to Us," Google, February 4, 2008
Innocent participants in malware infection chain

- Fraudsters exploited a XSS vulnerability to inject a modified login form onto Banca Fideuram’s Web page
- User’s account info is sent to a server in Taiwan
Data theft is rampant ...

<table>
<thead>
<tr>
<th>Low</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2</td>
<td>$5</td>
</tr>
<tr>
<td>$25</td>
<td>$35</td>
</tr>
<tr>
<td>$200</td>
<td>$300</td>
</tr>
<tr>
<td>$25,000</td>
<td>$40,000</td>
</tr>
</tbody>
</table>

- Payment card
- Magnetic stripe data
- Full account information
- Zero-day malware

Other items for sale

- eBay, PayPal accounts
- Driver’s licenses
- Birth certificates

Source: The Aegenis group, USENIX 2008
Web 2.0 highlights application security

• Increased attack surface
  » Client side state matters
  » Code injection is a top security risk

• Developing secure web 2.0 apps is challenging
  » Security testing is more complex
  » Authentication/Authorization logic is more complex

• Traditional trust model is broken
Security has to be baked in rather than painted on ...
Why Security during development?

85% of code flaws are introduced during coding

Source: Applied Software Measurement, Capers Jones, 1996
Why security during development - recap

► The #1 reason: to catch vulnerabilities before attackers do
► Reduce cost of fixing bugs and vulnerabilities
  ► Example: $30,000 post release, $3,000 integrated testing, $500 development
► Help to achieve regulatory compliance
  ► PCI, HIPAA, Sarbanes-Oxley, …
An application lifecycle view of security

Design & architecture: Design reviews and architecture audits to identify and eliminate design flaws
Implementation: Developer training and white box testing to identify code-level defects
QA testing: Black box tests to spot known security flaws
Operation: Continuous monitoring, assessment, analysis, and response to manage new vulnerabilities and threats
• Begin with requirements
  » What business asset does the software represent?
  » What assets of value are accessible from the software?
  » What protections must be provided for those assets?

• Design and architecture review
  » What kind of operational environment
  » What are the threats to my assets in this environment?
  » What threat-mitigation measures must be provided?
• Develop a company secure coding guidelines and standard
  » Libraries of approved security functions
  » Vulnerability remediation process
• Develop language-specific checklists
• Utilize code-level analysis tools
  » Static analysis tools
  » Security testing tools
• Code reviews and walkthroughs
Specific code-level guidelines

• Validate input from all untrusted data sources, including web inputs, environmental variables, network interfaces, and user controlled content
• Do not use any unsafe functions
• Sanitize all data sent downstream
• Use tried-and-true crypto packages: e.g., Bouncy castle,
• Verify security policies are enforced
• Do not hard-code default and test accounts
• Default deny: base access decisions on permission rather than exclusion
• Fault injection
• Fuzzing
• Black-box and automated penetration testing
• Augment with manual testing
• Whittaker and Thompson, *How To Break Software Security*, Addison-Wesley, 2004
• Secure configuration
  » Disable all default accounts at the end of installation
  » Force the user to set strong administrative passwords
  » Configure appropriate logging and auditing procedures

• Secure operation
  » Consider web application firewall
  » Periodically conduct penetration and black box testing
• Maintenance

  » Enforce all of your secure software development processes for maintenance releases of code
  » Integrate with change management procedures
  » Make sure that your maintenance engineers fully understand the intended use and architecture of the product and adhere to operational principles
  » Have an incident response plan
How are companies adopt SDLC?

• >65% companies have penetration testing and black box scanning

• Source code security technology is by far the smallest
  » Developers are under time-to-market pressure
  » Security tools render false positives are not acceptable
Comparing white-box and black-box

All security flaws

White-box analysis

Automated black box
1 + 1 > 2?

• Integrated whitebox and blackbox, e.g.,
  » Using black box tests to verify white-box findings
  » Use white-box analysis to guide black box tests
• Users should demand integrated vendor functionality
  » Only combine reports is not enough
Developer training is essential

- Defense in depth
- The principle of least privilege
- The principle of segregation
- Fail closed or fail secure
- Enforce known good traffic (white-list approach)
- Verify trust model
- Keep it simple. Complex designs increase the likelihood of security flaws.
- ...
Build your application security maturity

Reactive
- Predominately reactive measures: fix vulnerabilities if exploited
- Little SDLC capabilities
- Ad hoc and manual security testing

Systematic
- Systematic security testing for selective projects
- Established secure coding guidelines
- Ad hoc developer/tester training

Proficient
- Standardized application security process throughout
- Established SDLC practice
- Integrated remediation processes
- Internal compliance policies established

Center of excellence
- Mature, end-to-end SDLC program
- Centralized management of the program
- Systematic training
- Clearly defined success metrics
- Established self-evaluation process
A look at trends going forward
The consumer experience
The enterprise experience
The consumer and enterprise experience gap will disappear...
Prediction …

- Reputation will be pervasive
  - Reputation for program/code
- Ways to recognize invariants in the code
  - Innovative fingerprinting technology
- Ways to analyze software
  - Faster than current emulation technology
- Reputation will live in the cloud
- Optimization on the gateway
Application security is not
• Patching, encryption
• Even secure coding

Application security is
• Applying information security principles to software engineering and operations
• Process of designing, building, deploying, and maintaining software that cannot easily be misused for malicious purposes
Information resources for application security

• Open Web Application Security Project (OWASP) Top Ten at http://www.owasp.org

• Web Application Security Consortium (WASC) Threat Classification at http://www.webappsec.org

• Chenxi Wang, Ph.D. “Managing Application Security from beginning to end”, Forrester research report, October 2007.


Thank you.

Questions?


• Chenxi Wang
  Principal Analyst
  Forrester Research

• Email
cwang@forrester.com

• Blog
blogs.forrester.com/srm