Continuous Prevention Testing

By Andre Gironda

OWASP October 2007
Bio

- Andre Gironda
- OWASP Phoenix, Chicago, MSP
- Other projects involved in
  - WASC WASSEC
  - NIST SAMATE Web Application Scanner Focus Group
Web scanner challenges

- Logical flaws
- Crawling HTTP and Ajax
- Scraping [malformed] HTML and scripts
- False negatives / positives
- Coverage
- Reports sit on desks
Current situation

- RIA / RCP frameworks
- Marketing vs. security
- Software weaknesses
  - CWE scoring (Wysopal)
  - CVE data (Linder, ModernApps)
Outline of this talk

- OWASP: Problems to solve
- Developer testing and inspection
- Automated software testing
- Process improvements
- Security testing improvements
OWASP: Problems to solve

- Identify and code around security weaknesses
- Provide guides and metrics
  - Modelers vs. measurers (Jaquith)
Development: Epic fail #1

- Commercial software: “2x size every 18 months” on average
  - Developer education
  - Security {people|process|tech}
- One of your developers knows how to fix everything
- One of your developers is continually allowed to check-in the same security-related defect over and over and over again
Intake testing: Keep the bar green

- Unit testing, “Never in the field of software development was so much owed by so many to so few lines of code.” – Martin Fowler pretending to be Winston Churchill
  - Developer freebies in their IDE/SCM (e.g. promotion of warnings to errors)
  - Static code analysis
  - Coding standards
  - Continuous-testing IDE with decision-condition coverage
Smoke testing: Build every day

- Component tests (DB stubs, mock objects)
- Continuous integration server
  - ThoughtWorks Buildix boot CD
    - Subversion, Trac, CruiseControl, User manager
  - Atlassian JIRA/Confluence, FishEye, Bamboo
- Prioritization of defect fixes with issue tracking
  - Code metrics
Inspection! Review the code

- Major builds – securecoding (SC-L)
- Fagan inspection
- Peer review
  - Author
  - Reviewer
  - Moderator
- Continuous inspection at each check-in
Automated testing: Fail #2

- Automated software testing (for quality)
  - Finds 30% of the possible defects
  - Eats up 50%-80% of the development budget
Websites in outer space

• Safety testing (NASA) vs. security testing
  – Model checking
  – Smart fuzz testing
  – Concolic unit testing

• Two motivations to fuzz fat apps (Evron)
  – Fuzz before release: security vendors
  – Fuzz before purchase: financials, ecommerce
System integration testing

- Test the code in working server environment
- Components work with all other components
- Script-driven, domain-specific languages
  - Protocol drivers, proxy fuzzers
- Data-driven test frameworks
Functional testing

• Test the client
• Simulate or drive browsers and plug-ins
  – Application drivers
• Repeatable tests
• Capture/playback test frameworks
Regression testing

• Re-test the application for the same bugs
• CVE finds a chance >15% to cause a new defect at least as severe as the fixed issue
• Web application security defects are completely ignored 90% of the time, YoY
• Regression testing vs. maintenance testing
## Process improvements: Win #1

Design reviews with threat-modeling

<table>
<thead>
<tr>
<th>Attack-trees</th>
<th>MITRE CAPEC</th>
<th>CWE</th>
<th>WASC TC</th>
<th>OWASP T10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seven pernicious kingdoms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STRIDE</td>
<td>ITU-T X.805</td>
<td></td>
<td></td>
<td>Trike</td>
</tr>
</tbody>
</table>
Secure development 101

• Continuous-prevention development
  – Write a unit test to check for known vulns
  – Add it to your daily builds (i.e. CI server)
  – Bonus: Assert others by looking for defect’s fix

• Better workflow methodologies and tools
  – Code review
  – Architecture review
Secure development lifecycle

- Expensive to implement
- Only Microsoft does this today
- If SecurityCost > SDLCost Then SDL
Security and quality metrics

- Business scorecards, 6S tools – you!
- ISAC’s – information sharing (Geer)
- Application security vendors / consultants
- MITRE / securitymetrics.org
- OWASP / WASC / ISECOM / NIST
- Data breaches (Shostack)
Security testing today: Win #2

- Complete automation, “default mode”
- Fully automated scanning solution
- Don’t exist for quality or safety testing
- Why would they exist for security testing?
Medical testing and biostats

- Binary classification: No gold standard test
  - Sensitivity (positive test that ground beef has E.Coli)
  - Specificity (negative test that ground beef does not have E.Coli)
- Developers want higher specificity
- Security professionals want higher sensitivity
- Provide good benchmarks and analysis from weakness and vulnerability statistics
Software security standards

- XPath and AVDL tool support
- The wisdom of crowds / reputation systems
- Popular IDE and build server code metrics (e.g. Fortify SCA, Microsoft VS2k8)
- Secure frameworks (e.g. HDIV, .NET 3.5)
  - Perfection is achieved not when there is nothing left to add, but rather when there is nothing left to take away
Web scanner improvements

<table>
<thead>
<tr>
<th>Logical flaws</th>
<th>Multiple credentials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crawling HTTP and Ajax</td>
<td>Application drivers</td>
</tr>
<tr>
<td>Scraping [malformed] HTML and scripts</td>
<td>Better parsers, domain specific languages</td>
</tr>
<tr>
<td>False negatives / positives</td>
<td>Binary classification: sensitivity</td>
</tr>
<tr>
<td>Code coverage</td>
<td></td>
</tr>
<tr>
<td>Reports sit on desks</td>
<td>Submit to issue tracking (or XML out)</td>
</tr>
</tbody>
</table>
Refs

OWASP DC on RIA: http://www.owasp.org/index.php/RIA_Security_Smackdown
CWE scoring, Chris Wysopal: https://securitymetrics.org/content/attach/Metricon2.0/Wysopal-metricon2.0-software-weakness-scoring.ppt
FX / Felix Linder: http://conference.hackinthebox.org/hitbsecconf2007kl/?page_id=130
Promoting Warnings to Errors: http://safari5.bvdep.com/9780596510237/enabling_useful_warnings_disabling_useless_ones_and_promoting
Refs (cont’d)

SecureCoding Mailing-list: http://www.securecoding.org/list/
Atlassian (formerly Cenqua) Crucible: http://www.atlassian.com/software/crucible/
Concolic testing: http://osl.cs.uiuc.edu/~ksen/cute/
Fuzzing in the corporate world, Gadi Evron:
http://events.ccc.de/congress/2006/Fahrplan/events/1758.en.html
GPath with XmlParser and NekoHTML:
Twill: http://twill.idyll.org MaxQ: http://maxq.tigris.org
HDIV: http://hdiv.org Topps meat E.Coli
Microsoft Visual Studio 2008: http://www.eweek.com/article2/0,1895,2192515,00.asp
http://en.wikipedia.org/wiki/Binary_classification
Brian Chess & Katrina Tsipenyuk:
http://securitymetrics.org/content/attach/Welcome_blogentry_010806_1/software_che ss.ppt