Continuous Prevention Testing

By Andre Gironda

Toorcon 9, October 2007

Bio

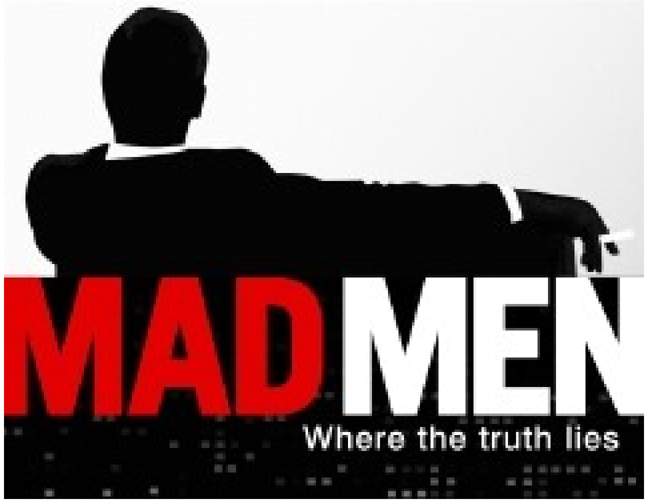
- Andre Gironda
- Projects involved in
  - OWASP Tools Project
  - WASC WASSEC
  - NIST SAMATE Web Application Scanner Focus Group

Web scanner challenges

- Logical flaws
- Crawling HTTP and Ajax
- Scraping [malformed] HTML and scripts
- False negatives, false positives
- Reports sit on desks

Current situation

- RIA / RCP frameworks
- Marketing vs. security
- Software weaknesses
  - CWE scoring (Wysopal)
  - CVE data (FX, ModernApps)



Outline of this talk

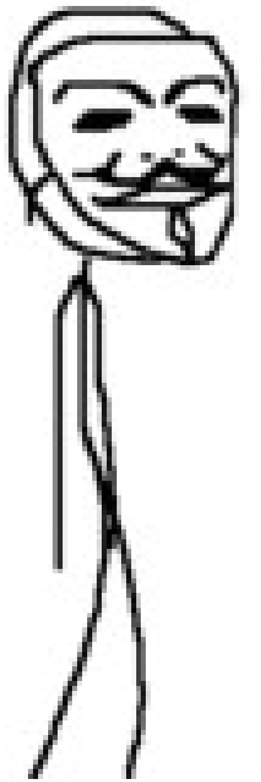
- The vulnerability problem
- Developer testing and inspection
- Automated software testing
- Process improvements
- Security testing improvements

The vulnerability problem

- Identify and code around security weaknesses
- Provide formulas and metrics
  - Modelers vs. measurers (Jaquith)

Development: Epic fail #1

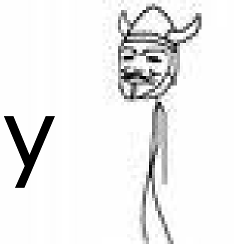
- Commercial software: “2x size every 18 months” on average
  - Developer education
  - Security {people|process|tech}
- One of your developers knows how to fix everything
- One of your developers is continually allowed to check-in security-related defects



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

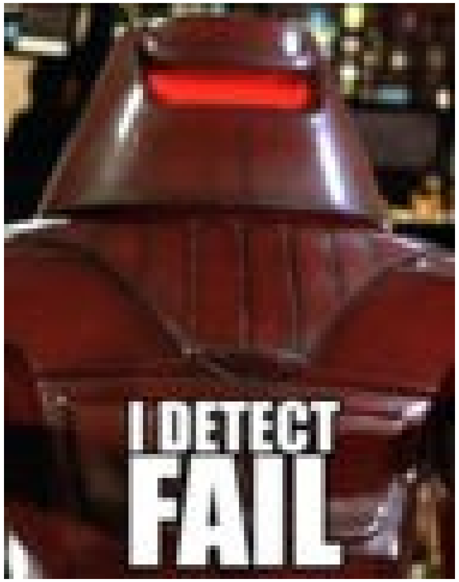


- Component tests (DB stubs, mock objects)
- Continuous integration server
  - ThoughtWorks Buildix boot CD
    - Subversion, Trac, CruiseControl, User manager
  - Atlassian JIRA, FishEye, Bamboo
- Prioritization of defect fixes with issue tracking
  - Code metrics

Inspection! Review the code

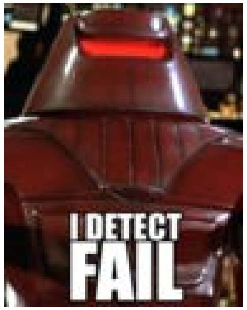
- Major builds – securecoding (SC-L)
- Fagan inspection
- Peer review
  - Author
  - Reviewer
  - Moderator
- Continuous inspection at each check-in

Automated testing: Fail #2



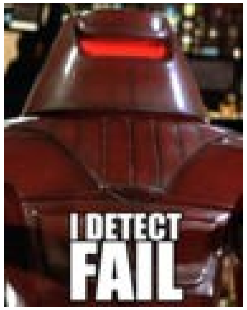
- Automated software testing (for quality)
  - Finds 30% of the possible defects
  - Eats up 50%-80% of the development budget

Websites in outer space



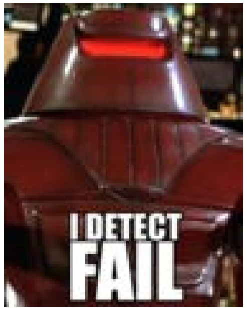
- Safety testing (NASA) vs. security testing
  - Model checking
  - Smart fuzz testing
  - Concolic unit testing
- Two motivations to fuzz fat apps (Evron)
  - Fuzz before release: *security vendors*
  - Fuzz before purchase: *financials, retail*

System integration testing



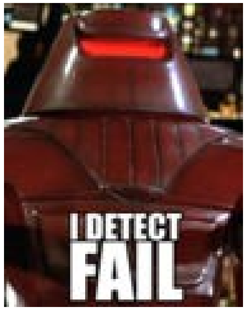
- Test the code in working server environment
- Components work with all other components
- Script-driven, domain-specific languages
  - Protocol drivers, proxy fuzzers
- Data-driven test frameworks

Functional testing



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

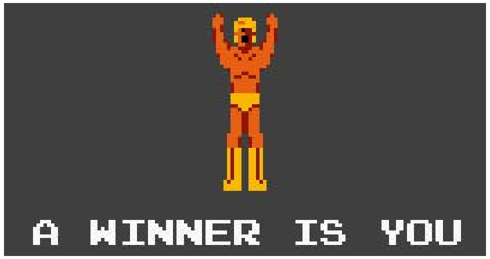
Regression testing



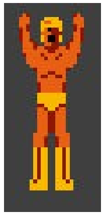
- Re-test the application for the same bugs
- CVE finds a chance >15% to cause a new defect at least as severe as the fixed issue
- Web application security defects are completely ignored 90% of the time, YoY
- Regression testing vs. maintenance testing

Process improvements: Win #1

Design reviews with threat-modeling



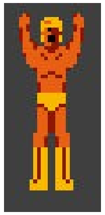
Attack-trees	MITRE CAPEC CWE	WASC TC
Seven pernicious kingdoms		OWASP T10
STRIDE	ITU-T X.805	Trike



Secure development 101

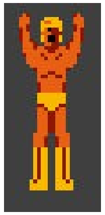
- Continuous-prevention development
  - Write a unit test to check for known vulnerabilities
  - Add it to your daily builds (i.e. build server doing continuous integration)
  - Bonus: assert others by looking for defect’s fix
- Better workflow methodologies and tools
  - Code review
  - Architecture review

Secure development lifecycle



- Expensive to implement
- Only Microsoft does this today
- If SecurityCost > SDLCost Then SDL

Security and quality metrics

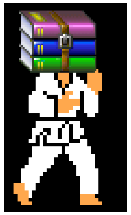


- Business scorecards, 6S tools – you!
- ISAC’s – information sharing (Geer)
- Application security vendors / consultants
- MITRE / securitymetrics.org
- OWASP / WASC / ISECOM / NIST
- Data breaches (Shostack)

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

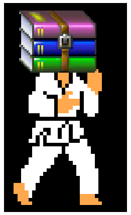


- Binary classification: No gold standard test
  - Sensitivity (positive test that ground beef has E.Coli)
  - Specificity (negative test that ground beef does not have E.Coli)
- Developers want higher specificity
- Security folks prefer higher sensitivity
- Provide good benchmarks and analysis from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

Software security standards

- XPath and AVDL tool support
- The wisdom of crowds / community reputation systems (Ryan & Ramya)
- 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



Logical flaws	Multiple credentials
Crawling HTTP and Ajax	Application drivers
Scraping XHTML, XML, malformed HTML and scripts (JS, Flash, etc)	Better parsers, domain specific languages
False negatives False positives	Binary classification: sensitivity
Reports sit on desks	Submit to issue tracking (or XML out)

References

Robert Auger: http://www.cgisecurity.com/articles/scannerchallenges.shtml

L.Suto: http://ha.ckers.org/blog/20071014/web-application-scanning-depth-statistics/

OWASP DC on RIA: http://www.owasp.org/index.php/RIA_Security_Smackdown

Java RCP: http://www.eclipse.org/org/press-release/20071015_raprelease.php

CWE scoring, Chris Wysopal: https://securitymetrics.org/content/attach/Metricon2.0/Wysopal-metricon2.0-softwar

FX / Felix Linder: http://conference.hackinthebox.org/hitbsecconf2007kl/?page_id=130

Security Metrics: Modelers vs. measurers - http://safari5.bvdep.com/9780321349989/ch02lev1sec2?imagepage=13

Continuous Integration book - http://www.testearly.com

Mark Curphey – Types of testing: http://securitybuddha.com/2007/09/03/the-art-of-scoping-application-security-reviev

Promoting Warnings to Errors: http://safari5.bvdep.com/9780596510237/enabling_useful_warnings_disabling_usele

PMD: http://pmd.sf.net CheckStyle: http://checkstylesf.net FindBugs: http://findbugs.sf.net

CT-Eclipse: http://ct-eclipse.tigris.org EMMA: http://emma.sf.net http://www.eclemma.org

Buildix: http://buildix.thoughtworks.com Java metrics: http://metrics.sf.net

References (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
Proxy Fuzzing:
 http://www.darknet.org.uk/2007/06/proxyfuzz-mitm-network-fuzzer-in-python/
GPath with XmlParser and NekoHTML:
 http://sylvanvonstuppe.blogspot.com/2007/08/ive-said-it-before-but.html
Canoo WebTest: http://webtest.canoo.com Jameleon: http://jameleon.sf.net
Twill: http://twill.idyll.org MaxQ: http://maxq.tigris.org
OpenQA Selenium: http://openqa.org WebDriver: http://code.google.com/p/webdriver/
HDIV: http://hdiv.org Topps meat E.Coli
 http://rationalsecurity.typepad.com/blog/2007/10/topps-meat-comp.html
Ryan and Ramya at GTAC: http://www.youtube.com/watch?v=YCatiB8d100
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_ch