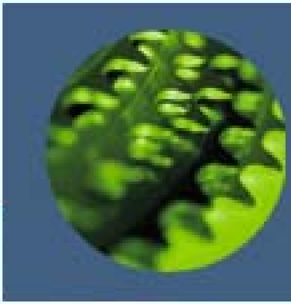


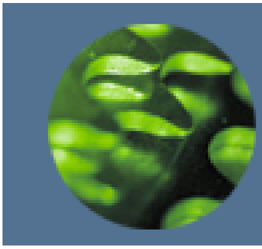
Web Application Security

Presented to:
ISSA – Grand Rapids Chapter
January 19, 2007

Cliff Barlow
Director Security Services

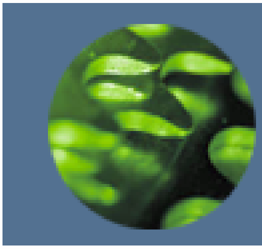
Jim Segreti
Senior Consultant

KoreLogic (Security)



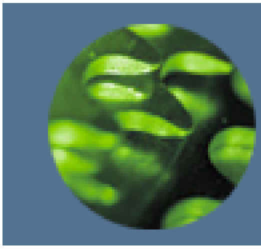
Presentation Goals

- Provide attendees with an understanding of web application security fundamentals.
- Educate attendees on best practices and common vulnerabilities that KoreLogic observes during web application assessments.
- Increase awareness of common attacks and how to strengthen a web application’s resistance to attack.
- Security Testing Considerations
  - Tools vs. Hands-On (Differences in techniques)
- Application Security Best Practices
  - Application Security Program
  - Effective Presentation of Test Findings
  - Security Metrics
  - Testing Approach



Disclaimers

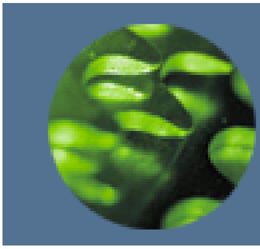
- In one hour, we can’t cover every aspect of web application security—we have selected those we believe to be most relevant.
  - We typically take 1-2 days for intro courses.
- The briefing is not intended to teach coding (but we will show how to avoid/correct flaws).
- Sample techniques outlined in this presentation are intended to be performed only by authorized individuals.
- Attempts to perform unauthorized tests are illegal.
- Security testing is expressly forbidden unless authorized.
- The recommendations we provide are considered best practices. However, each security decision should be risk-based and balanced with business requirements.



The Business Case

Regulatory Drivers - Why Secure Your Applications Anyway?

- Gartner Group research:
  - 75% Of Hacks Occur at the Application Level
  - It is estimated that by 2009, 80 percent of Companies will have suffered an application security incident, and as a result, will react by creating roles in the Application Development and testing organizations to insure that security is handled at the application level.
- Network perimeter is blurring to accommodate “externalization”. Some organizations now think of security architectures as "zones of risk" and "zones of trust.“ (Gartner)
- Payment Engines (VISA, AMEX, Banks) driving new apps standard to require testing/compliance before able to use services.
- Business partners and customers are increasingly more security conscious ….. they are starting to test your applications before they agree to do business with you.
- Corporate reputation is an intangible and valuable asset.
  - A publicized security incident - *even if no harm was done* - will seriously impact client trust.



Exposure Incidents

2006 – U.S. Dept of Education

- 21,000 Loan records exposed
- Cause: Faulty software upgrade

2006 – PortTix (Portland, Maine)

- Approximately 2000 credit card records exposed
- Probable cause: Inadequate input validation

2006 – Orange County Controller (Florida)

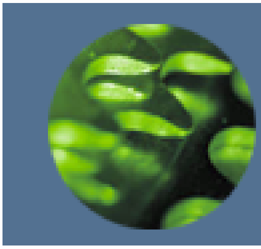
- Personal Documentation Exposed (unspecified amount)
- Cause: Lack of data sanitization

2005 – ChoicePoint

- Losses estimated in excess of $15 million
- Cause: Ineffective client screening

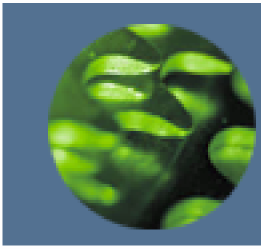
2005 – LexisNexis

- Disclosure of 280K+ personal records
- Probable cause: Ineffective password management



Trend Towards Application Level Attacks, Malware, and Trojans

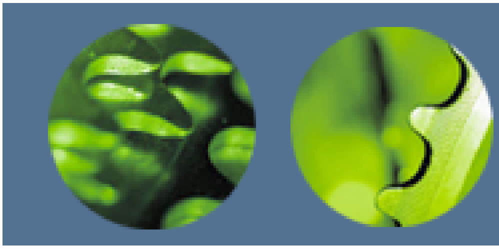
- Attackers follow the path of least resistance:
  - 3-5 Years ago, most all of the “top 10” vulnerability lists cited network based attacks.
  - Today, two of the top Windows vulnerabilities on the SANS/FBI top 20 list are application-level exploits.
  - In our last 10 assessments, a majority of the vulnerabilities were at the application layer.
- The network perimeter is more hardened, yet more open than ever before.
  - Disallowed services are more truly disallowed.
    - Fragmented attacks, half-open scans, and other advanced network based attacks are quickly becoming obsolete due to new firewall technology, and the proper configuration of firewalls.
  - However, allowed services are more “open” than ever before.
    - Applications
    - Malware and Spyware
    - B2B LANS and VPNs can be vectors for attacks.
    - Email-based Trojans are common.
    - Browser-based vulnerabilities are a serious threat.



…Threats Evolve

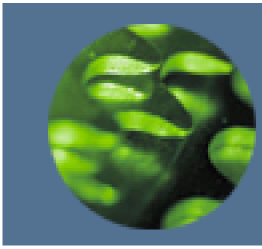
- Then: "In the old days, you were up against a teenager without a girlfriend, working alone in his bedroom."
- Now:
  - Professional criminals (identity theft), espionage (business and international) and terrorism have moved in …
  - 80 percent of the malicious-code samples “the company” receives are written by online criminals to make money through identity theft or hacking. Just 5 percent are written by immature hackers or “script kiddies”.
    - Researchers at Kaspersky Lab receive 5,000 samples of malicious code each month, double what they received 1 year ago.
    - Kaspersky Lab's database of malicious code has grown by 50 percent in the last year, to more than 150,000 records.

Source: Paul Roberts, 10/17/05



Common Web Application Vulnerabilities

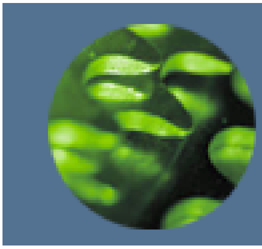
Vulnerabilities	Examples
Supporting Infrastructure	➢ **Un-patched web server leaves entire application at risk**
Inadequate Input Validation	➢ **SQL Injection / System Command Injection**
Confidentiality	➢ **Exposure of Sensitive Information**
Authentication	➢ **Network-level authentication, transaction-level, user-level**
Sign-on Process	➢ **Account harvesting, password harvesting**
Session Management	➢ **Lack of session invalidation, predictable IDs**
Information Leakage	➢ **HTTP headers / Developer comments**
Server-side Executables	➢ **Manipulation of executables (e.g., CGI, ASP, servlets).**
Sign-off Process / Caching	➢ **Resumed connections allowed, cached information**
Transaction Level Issues	➢ **Hidden Form elements, HTTP methods**
Privilege Escalation Of Data	➢ **User ‘A’ accessing User B’s Data**
Privilege Escalation of Functionality	➢ **User ‘A’ accessing functionality intended for User ‘B’**



The Top 5

Specific Examples Of Web Application Faults

1) SQL Insertion via login and password page. Led to complete backend database compromise.
2) Faulty substitution cipher on banking application led to viewing of .JPGs of every user of the application’s cancelled checks.
3) Logic error enabled one user to see another user's private information. This caused an ECommerce app to be down for 4 days. Loss of business $50,000.00.
4) Buffer Overflow of a single variable crashes a website on a mainframe.
5) Shell-script based web application with no input checking gave a back door into entire intranet of large financial institution.



KoreLogic's "Distilled" 8

Web Application Vulnerabilities

1. Access Control = Relationship between user action and corresponding login ID
2. Configuration Management = Web and application servers, supporting infrastructure
3. General Application Engineering = Logic and functionality, application interconnection
4. Authentication = Authorizing access to the application
5. Error Handling = Sanitization of error messages and conditions
6. Input Validation = Client and server side input filtering
7. Data Protection = Encryption and confidentiality
8. Session State Maintenance = Cookies and other means of maintaining session integrity

Take Perspective of Attackers

It is difficult to anticipate the attack or profile who is attacking because of advantages that the attacker has…

- A needle in a haystack (esp. true on high volume sites).
- May need only to find one hole.
- Time is an ally. Simply wait until next 0day exploit.
- Tremendous amount of resources are available on the Internet (chat/irc, hack sites, google, etc.).

Where Do Attackers Look? Answer: Where there is…

- Low Risk of Detection
- High Chance of Success
- High Payoff
- Low Effort or Cost

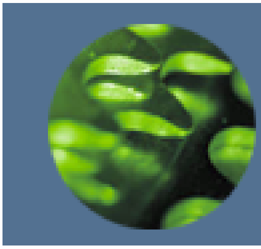
Threat Modeling
a.k.a "Mental Hacking"©

Assume that someone out there will intentionally try to break your application and may have access to greater resources than you expended in developing it.

"Don't underestimate the power of the 'dark side'..."

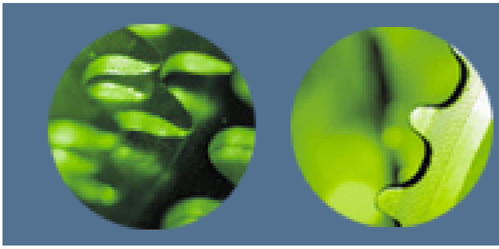
- Software disassembly techniques and tools are getting better all the time
- If your secrets are important enough, someone will attempt to get them

Think like an attacker … Develop attack scenarios and ensure your application can resist them.



Testing Keys

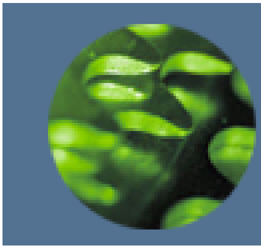
- Setting Scope
  - An application architecture can be very broad:
    - Web (Internet, Intranet, Extranet)
    - J2EE, .NET, Thick desktop clients, AJAX
    - Integration with Application Service Providers/Partners
    - Centralized authentication, SSO, Network Desktop login
    - Multi tier infrastructure/architecture
  - Who Developed the Application?
    - In-house development, COTS, Outsourced/hosted applications
  - What drives the security assessment scope?
    - Criticality of the application
    - Data sensitivity
- Core Skills of Assessors
  - Seasoned in business and risk management.
  - Not tool jockeys.
  - Provable skills, certified team…more importantly, “Street Smart”.
  - Senior team, recognized in industry, having completed many similar assessments
  - Flexibility to shape services to your specific needs, utilize/build your requirements to determine your business risk and can provide you practical data, strategies and remediation approaches that address your specific business objectives.



What About Application Vulnerability Scanners?

Scanning Tools:

- Good for less critical applications and time-consuming types of test.
- Use only as a starting point for business-critical applications.
- Why pay someone to run a tool you can run.
- Useful for less critical applications, but beware of limitations:
  - Cannot detect logic issues (e.g., a non-administrative user that is able to perform administrative tasks).
  - Only tests known vulnerabilities (that have been published).
  - Cannot evaluate client-side code (e.g., VB Script, ActiveX).
  - Cannot work with applications that use non-HTML interfaces (e.g. Java).
  - Limited ability to correlate events and correlate minor issues that may lead to larger exposure.
  - Propensity for false results.
  - Time-consuming to run on large applications.
  - Nothing beats a big brain. A smart creative person will always find things automated tools miss.



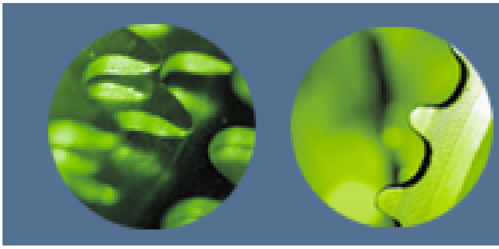
Expert (Hands-On) Testing

“Expert Testing” involves the notion of web application analysis that an experienced analyst performs, usually with the use of automated utilities for performing task-specific functions.

Approach:

- Use automated tools during development and for less critical applications, but…..
  - Should only be a part of a business-critical application test.
  - And…why pay someone to run a tool you can run.
  - If you scan only, be sure to caveat your findings by saying so.
  - Tools are important part of security toolkit, but be aware of constraints.

- Use Expert Testing for security QC and business critical applications:
  - Defacto standard becoming required use of Third party Expert Testing by business partners and industry standards (i.e. PCI PABP/PASS).

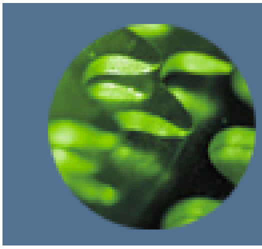


Web Application Tools: What Tools Can Help Me?

Tools that can help evaluate an application:

- SPIKE and SpikeProxy http://www.immunitysec.com/resources-freesoftware.shtml
- Burp Proxy http://portswigger.net/proxy/
- Paros Proxy http://www.parosproxy.org/index.shtml
- Stunnel (SSL Wrapper) http://www.stunnel.org/
- SWAAT (Web App Source Code Analyzer) http://sourceforge.net/projects/taof
- OWASP.org has multiple Web Application Security Tools WebScarab , Pantara, etc (including insecure applications you can test against).

  http://www.owasp.org/index.php/Category:OWASP_Download



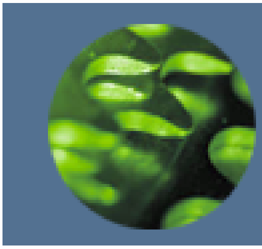
Application Security Metrics

Process Metrics

- Is a SDL Process used? Are security gates enforced?
- Secure application development standards and testing criteria?
- Security status of a new application at delivery (e.g., % compliance with organizational security standards and application system requirements).
- Existence of developer support website (FAQ's, Code Fixes, lessons learned, etc.)?
- % of developers trained, using organizational security best practice technology, architecture and processes.

Management Metrics

- % of applications rated “business-critical” that have been tested.
- % of applications which business partners, clients, regulators require be “certified”.
- Average time to correct vulnerabilities (trending).
- % of flaws by lifecycle phase.
- % of applications using centralized security services.
- Business impact of critical security incidents.

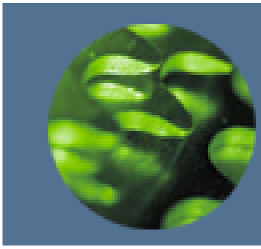


Application Security Metrics

Vulnerability Metrics

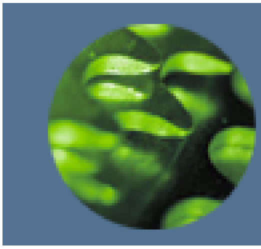
- Number and criticality of vulnerabilities found.
- Most commonly found vulnerabilities.
- Reported defect rates based on security testing (per developer/team, per application).
- Root cause of “Vulnerability Recidivism”.
- % of code that is re-used from other products/projects*.
- % of code that is third party (e.g., libraries)*.
- Results of source code analysis**:
  - Vulnerability severity by project, by organization.
  - Vulnerabilities by category by project, by organization.
  - Vulnerability +/- over time by project.
  - % of flaws by lifecycle phase (based on when testing occurs).

Source: * WebMethods, ** Fortify Software



To Protect an Application….

1. Start with a secure infrastructure.
2. Bake security in to the system – don’t try to bolt it on afterwards:
   1. Secure Application Development Life-Cycle.
   2. Secure Architecture – technical and non-technical controls.
   3. Train application developers in secure web-based coding practices.
3. Test new systems prior to production … Re-test regularly
4. Stay abreast of new vulnerabilities.



What are the most critical steps organizations can take to protect its web applications?

1 Training – developers, testers, project managers, business unit management. ➔ There is no better ROI or marketing for application security than training

2 Third party testing of mission-critical applications ➔ Ensures you have a handle on your immediate risk

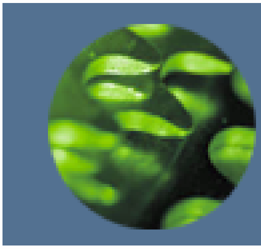
3 Security monitoring to detect and respond to attacks (app-specific IR). ➔ Ideally, you detect/thwart. If not, must be able to tell stakeholders about impact

4 Develop metrics ➔ The language of management

5 Collaboration between development, infrastructure, and security teams. ➔ Each group depends on the other for security

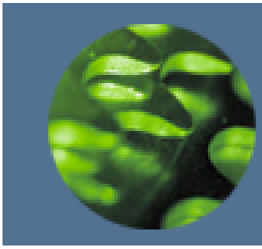
6 Implement an SDL. ➔ Incremental roll-out is fine

7 Over several years, develop an application security program. ➔ The end game



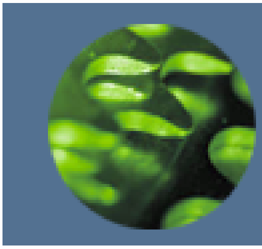
Resources

- The Open Web Application Security Project (OWASP): www.owasp.org.
  - KoreLogic leads two OWASP projects
    - Metrics: http://www.owasp.org/index.php/Category:OWASP_Application_Security_Metrics_Project
    - Security Assessment Standards http://www.owasp.org/index.php/Category:OWASP_Application_Security_Assessment_Standards_Project
- Secure Software Development Life Cycle:
  - "The Security Development Lifecycle", Howard and Lipner,
  - "Security in the Software Lifecycle", DHS, Cybersecurity Div.
- "Writing Secure Code", Howard and LeBlanc
- "A Clinic to Teach Good Programming Practices", Matt Bishop, http://nob.cs.ucdavis.edu/bishop/talks/2006-cisse-2/clinic.html
- CERT Secure Coding Standards, https://www.securecoding.cert.org/confluence/display/seccode/CERT+Secure+Coding+Standards



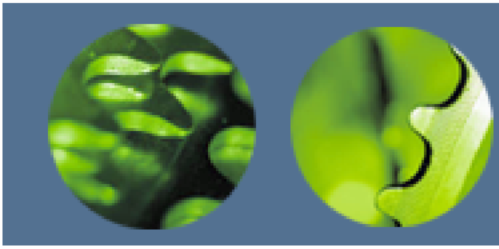
Resources

- Building Secure Software: How to Avoid Security Problems the Right Way http://www.amazon.com/exec/obidos/ASIN/020172152X
- Code Complete, 2nd Edition http://www.amazon.com/exec/obidos/ASIN/0735619670
- Secure Coding: Principles and Practices http://www.amazon.com/exec/obidos/ASIN/0596002424
- Secure Programming Cookbook for C and C++ http://www.amazon.com/exec/obidos/ASIN/0596003943
- Securing Java: Getting Down to Business with Mobile Code, 2nd Edition, http://www.amazon.com/exec/obidos/ASIN/047131952X
- Security Engineering: A Guide to Building Dependable Distributed Systems http://www.amazon.com/exec/obidos/ASIN/047138922
- Exploiting Software (Hoglund, Gary McGraw)
- J2EE Security for Servlets, EJBs, and Web Services, http://www.amazon.com/exec/obidos/ASIN/0131402641/



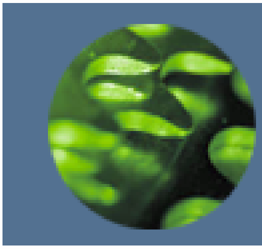
A CIO’s Checklist for Web Application Security

- ✓ Does our information security policy specify what types (Asset Classification) of applications <u>must</u> be tested?
- ✓ What is our process for evaluating the security of vendor-developed applications prior to purchase?
- ✓ Do we bake security into the system in the architecture and design phases of the project rather than trying to paint it on later? Do we:
  - ✓ Provide security advice to development team?
  - ✓ Perform threat / risk assessments to determine appropriate controls?
  - ✓ Offer standardized, re-useable security solutions?
- ✓ Is the application supported by a secure network and system infrastructure?
- ✓ Are the application developers trained in secure web application coding practices?
- ✓ Do we have effective collaboration between the development, test, infrastructure, and security teams?



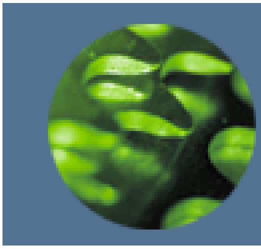
A CIO's Checklist for Web Application Security

- ✓ Do we have a program of compliance audits, vulnerability assessments, penetration tests, and management reviews?
- ✓ Do we require third party testing of mission-critical applications?
  - ✓ If feasible, offer this at no charge to business units to encourage testing
  - ✓ Be wary of "cheap" assessments – it usually means a scan that you might as well perform yourself
- ✓ Do we track metrics that help show benefits of well designed apps (e.g., "before and after")? For example, trend analysis showing:
  - ✓ Fewer high risk vulnerabilities
  - ✓ Reduction in well known, avoidable vulnerabilities
- ✓ Do we test critical new systems and infrastructure for vulnerabilities prior to placing them in production?
- ✓ Is testing performed in an environment representative of the future operational environment?
- ✓ How do we capture and share testing lessons learned?



A CIO’s Checklist for Web Application Security

- ✓ Do we have a managed change control process?
- ✓ How do we keep abreast of new vulnerabilities and fixes?
- ✓ Do we have a closed loop system of ensuring patches are applied?
- ✓ Do we have an effective threat monitoring program to detect/respond to attacks?
  - ✓ Network-based IDS, host-based IDS, Firewalls, Server logging
- ✓ How do we anticipate and respond to business partner and client inquiries about the security of our applications?
- ✓ To highlight problems early, avoid delays to the project production date and develop secure code, do we offer web application developer support such as:
  - ✓ Application security design support
  - ✓ Secure Web Application Development Standards
  - ✓ Testing Criteria
  - ✓ Developer Support (FAQ's, Code Fixes..)
  - ✓ Interim testing support and tools



Contact Information

Cliff Barlow

cbarlow@korelogic.com

www.korelogic.com

(office) 269-982-1707

Jim Segreti

jsegreti@korelogic.com

www.korelogic.com

(office) 410-867-9102