Secure development and the SDLC

Presented By Jerry Hoff

@jerryhoff
Agenda

• Part 1: The Big Picture
• Part 2: Web Attacks
• Part 3: Secure Development
• Part 4: Organizational Defense
Part 1: The Big Picture
Non stop hacking…
It seems like 2014 has been one endless series of massive hacks. Home Depot. Target. Neiman Marcus. Michaels. JPMorgan Chase. It’s been one hack after the other, each dumping anything from customer credit card numbers to mailing addresses into the wild.

The latest one is a big one: the United States Postal Service.

Here’s what we know:

- The hack was seemingly focused not on nabbing customer credit cards, but on email data. The hackers likely had access to confidential data on all 800,000 USPS employees. That includes names, Social Security numbers, addresses, and pretty much anything you’d put on a job application.
- Customer credit card information seemingly wasn’t exposed. However, anyone who used USPS customer support from January 1st to August 16th of 2014 might have had information stolen, depending on what information they provided to the CS rep: names, addresses, telephone numbers and email addresses.
- The intrusion was first detected in mid-September, nearly 2 full months before being disclosed. The USPS says this delay was because “communicating the breach immediately would have put the remediation actions in jeopardy...”
Hundreds of PHL security cams hacked, posted online
November 11, 2014 12:37pm

Here's one more reason for owners of security cameras to change their devices’ default passwords: their private lives could be made public by a hacker's website.

The Cybercrime Economy
U.S. weather system hacked, affecting satellites
By Jose Pagliery @Jose_Pagliery November 12, 2014: 2:34 PM ET

Hackers attacked the U.S. weather system in October, causing a disruption in satellite feeds and several pivotal websites.

The National Oceanic and Atmospheric Administration, NOAA, said that four of its websites were hacked in recent weeks. To block the attackers, government officials were forced to shut down some of its services.
Hacking Tops List of Crimes Americans Worry About Most

by Rebecca Riffkin

WASHINGTON, D.C. — As the list of major U.S. retailers hit by credit card hackers continues to grow this year, Americans are more likely to worry about having credit card information they used in stores stolen by computer hackers than any other crime they are asked about.

Sixty-nine percent of Americans report they frequently or occasionally worry about this happening to them. Having e-commerce or smartphone hacking (62%) is the only other crime that worries the majority of Americans.

Crime Worries in U.S.

How often do you yourself worry about the following things—frequently, occasionally, rarely or never? How often...
U.S. Hacked Despite $10B Spent Yearly on Security

Billions wasted, secrets exposed by federal employees’ carelessness & online playtime

November 11, 2014
The Intelligencer / Wheeling News-Register

They have clicked links in bogus phishing emails, opened malware-laden websites and been tricked by scammers into sharing information.

Federal employees and contractors scattered across more than a dozen agencies, from the Defense and Education departments to the National Weather Service, are responsible for at least half of federal cyberincidents each year since 2010, according to an Associated Press analysis of records.
why?
warum?
perche?
Web Site / Web Application
HTTP, HTML, JavaScript, CSS
Cookies, SVG, Plugins, Add-ons
iFrames, Flash, WebSockets
Client side database...

20 year old legacy!
Browsers inconsistent
Vulnerability

• Weakness that can be exploited to cause harm

• Each vulnerability has a “impact”

• Each vulnerability has a “likelihood”
OWASP Top 10 2013

• Big daddy of all web risk documents

• Do not base your security program on a “Top 10” list
  - Hoff’s Law

• http://www.owasp.org
Current “Top 10”

A1 Injection
A2 Broken Auth and Session Management
A3 Cross-Site Scripting (XSS)
A4 Insecure Direct Object References
A5 Security Misconfiguration
A6 Sensitive Data Exposure
A7 Missing Function Level Access Control
A8 Cross-Site Request Forgery (CSRF)
A9 Using Known Vulnerable Components
A10 Unvalidated Redirects and Forwards
Email: jerry@owasp.org  Twitter: @jerryhoff

OWASP ASVS

• Application Security Verification Standard (2014)

• Superb checklist

• DOWNLOAD THIS NOW (your homework)

V2. Authentication
V3. Session Management
V4. Access Control
V5. Malicious Input Handling
V7. Cryptography at Rest
V8. Error Handling and Logging
V9. Data Protection
V10. Communications
V11. HTTP
V13. Malicious Controls
V15. Business Logic
V16. File and Resource
V17. Mobile
Risk

• Risk = vuln likelihood * vuln impact

• Typically ranked “critical” through “low”

• Risk based approach = assign each risk a $$ amount
Security Control

• Mitigates an vulnerability

• For each major vulnerability, there is a corresponding security control

• Examples: cookie flag, http header, encoders, ORM, validation....
End of Part 1: The Big Picture
Part 2: Web Attacks
Finding and Exploiting Victims
Am I Secure?

bin ich sicher?

sono sicuro?
Advanced Google Search

about:  link:  site:  filetype:  inurl:
Finding file types

• site:example.com filetype:___

• Sometimes determine technologies (asp, php, jsp, aspx, cfm, pl)

• find interesting file types (.pdf, .docx, .xlsx, .txt, .readme)

• find very interesting file types (.log, .old)
Getting more specific

• **inurl:**

• Searches for keywords in URL

• Use this to find all “wordpress” login screens...

filetype:php inurl:wp-admin inurl:admin.php "lost your password"
Email: jerry@owasp.org  Twitter: @jerryhoff

http://www.exploit-db.com/google-dorks/
• Web Server
• SSL
• Namespace Provider
• Email
• Hosting
• CMS
• Analytics & Tracking
• JavaScript Library
• Document Info
• Encoding
• Server Information

http://builtwith.com/
Email: jerry@owasp.org  Twitter: @jerryhoff

Search Master Copy of CVE

You can search for a CVE number if known. To search by keyword, use a specific term or multiple keywords separated by a space. Your results will be the relevant CVE Identifiers.

By CVE Identifier

Submit

MITRE

Search Results

There are 667 CVE entries that match your search.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE-2014-5465</td>
<td>Directory traversal vulnerability in force-download.php in the Download Shortcode plugin 0.2.3 and earlier for WordPress allows remote attackers to read arbitrary files via a .. (dot dot) in the file parameter.</td>
</tr>
<tr>
<td>CVE-2014-5368</td>
<td>Directory traversal vulnerability in the file_get_contents function in downloadfiles/download.php in the WP Content Source Control (wp-source-control) plugin 3.0.0 and earlier for WordPress allows remote attackers to read arbitrary files via a .. (dot dot) in the path parameter.</td>
</tr>
<tr>
<td>CVE-2014-5342</td>
<td>Multiple cross-site request forgery (CSRF) vulnerabilities in the Disqus Comment System plugin before 2.7.6 for WordPress allow remote attackers to hijack the authentication of administrators for requests that conduct cross-site scripting</td>
</tr>
</tbody>
</table>

http://cve.mitre.org/cve/cve.html
End of Part 2:
Web Attacks
Finding and Exploiting Victims
Part 3: Secure Development
This is a big topic…

großes Thema…

grande tema…
Just for starters…
<table>
<thead>
<tr>
<th>Category</th>
<th>Solutions</th>
</tr>
</thead>
</table>
| Authentication        | Centralize!  
|                       | Shiro / Spring Security  
|                       | Active Directory  
|                       | Single Sign On / Access Management                                      |
| Authorization         | Centralize!  
|                       | External / URL based (Siteminder)  
|                       | Application / URL based (Filters)  
|                       | Authorization Annotations                                                  |
| Session Management    | Protection Session Cookie (httponly, secure, timeouts, cryptographically strong) |
| Database / SQL Injection | Parameterization / ORM                                                 |
| Injection             | Input validation / Encoding                                                |
| **Malicious Input**     | Centralized, standardized input validation
|                        | Contextual encoding of all untrusted input |
| **Crypto**             | Hashing: SHA-2 (soon SHA-3)  
|                        | Symmetric: AES  
|                        | Asymmetric: RSA, ECC  
|                        | Protect Keys: Don’t store in DB, use file system protections, read only, key rotation policy |
| **Error Handling / Logging** | No details in error messages  
|                        | Log all the things! |
| **Data Protection**    | No Cache  
|                        | No Autocomplete |
| **Communcation**       | HTTPS all the things (appropriately configured) |
| **3rd Party Libraries** | Track all 3rd party libs in deployment  
|                        | Check for known vulns, CVEs, etc.. |
Know thy frameworks...
Android

- Android Application Secure Design / Secure Coding Guidebook
- July 2014
- 450+ page document
- Japan Smartphone Security Association (JSSEC) Secure Coding Group
Struts 1: cleartext password in datasource config
Struts 2: using auto-executed plugins that are picked up from the classpath
Android: custom permission, exported receiver/provider/service, insecure version, touchjacking
Hibernate: cleartext password in connection config
JSF: client state saving config
JSF: developer mode enabled config
JPA: cleartext password in connection config
J2EE: auto-executed web fragments
J2EE: verb tampering (HTTP Verb bypass)
J2EE: session expiration
J2EE: secure / httponly flags
J2EE: error page
J2EE: url rewriting
WebSphere: serve servlets by class name

...... hundreds of platform specific configuration rules ......
Browser Security

• Strict-Transport-Security: Enforces HTTPS
• X-Frame-Options: Anti-clickjacking
• X-XSS-Protection: Anti-Reflected XSS
• Content-Security-Policy: Anti-XSS, etc..
• X-Content-Type: “nosniff”, prevents browser from guessing a content-type

https://www.owasp.org/index.php/List_of_useful_HTTP_headers
Part 4: Organization Defense
Simplified Implementation of the Microsoft SDL

Released November 4, 2010
For the latest information, please see http://www.microsoft.com/sdl.
Maturity Levels

Each of the twelve Security Practices has three defined levels that generally represent:

0: Implicit starting point representing the activities in the Practice being unfulfilled
1: Initial understanding and ad hoc provision of Security Practice
2: Increase efficiency and/or effectiveness of Security Practice
3: Comprehensive mastery of Security Practice

Education & Guidance

<table>
<thead>
<tr>
<th>EG 1</th>
<th>EG 2</th>
<th>EG 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OBJECTIVE</strong></td>
<td>Offer development staff access to resources around the topics of secure programming and deployment</td>
<td>Educate all personnel in the software life-cycle with role-specific guidance on secure development</td>
</tr>
</tbody>
</table>
| **ACTIVITIES** | A. Conduct technical security awareness training  
B. Build and maintain technical guidelines | A. Conduct role-specific application security training  
B. Utilize security coaches to enhance project teams | A. Create formal application security support portal  
B. Establish role-based examination/certification |
Maturity

• Assess your organization based on a maturity model

• Pick a target level based on organizational risk tolerance

• Plan a path to achieve the target level

• Direction: top-down (C-Level) - cascade to BA, Architects, Developers, QA, Deployment…

https://www.owasp.org/index.php/List_of_useful_HTTP_headers
Often overlooked…

• Standardized security controls
• Secure coding guidelines
• Dependency management
• Framework-specific security training
Example: Mozilla Playdoh

Playdoh

Mozilla’s Playdoh is a web application template based on Django.

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Playdoh is simply a pre-configured Django project that adds some essential modules and middleware to meet the following goals:

- Enhance the security of your application and its data
- Achieve optimal performance in the face of high traffic
- Localize content in multiple languages using Mozilla’s L10n standards
- Use the best tools and best practices to make development fun and easy

https://playdoh.readthedocs.org/en/latest/
Backlog

• What about existing applications?

• Quantify the risk (risk-based approach)

http://www.riskmanagementinsight.com/media/docs/FAIR_introduction.pdf
That’s it!

Questions? Fragen? Domande?