Agenda

- Statistics
- Threats
- Policies and Guidelines
- Resources
- Next Steps
- Questions
<table>
<thead>
<tr>
<th></th>
<th>All Websites</th>
<th>SSL Only Websites (44% of websites use SSL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAD High/Critical/Urgent Issue</td>
<td>83%</td>
<td>81%</td>
</tr>
<tr>
<td>CURRENT High/Critical/Urgent Issue</td>
<td>64%</td>
<td>58%</td>
</tr>
<tr>
<td>Vulnerability Resolution Rate</td>
<td>61%</td>
<td>58%</td>
</tr>
<tr>
<td>Average # of High/Critical/Urgent Vulnerabilities</td>
<td>16.7</td>
<td>9.7</td>
</tr>
<tr>
<td>Average # of serious unresolved vulnerabilities</td>
<td>6.5</td>
<td>4%</td>
</tr>
</tbody>
</table>

Source: 2009 Whitehat Security Statistics
Vulnerabilities Used by Hackers

- SQL Injection: 19%
- Content Spoofing: 10%
- Insufficient Authentication: 11%
- Unknown: 11%
- Insufficient Anti-Automation (DoS/Brute Force): 10%
- Configuration/Admin Error: 8%
- Cross-site Scripting (XSS): 8%
- Cross-site Request Forgery (CSRF): 5%
- DNS Hijacking: 5%
- Worm: 3%
- Other: 10%

Source: The Web Hacking Incidents Database Report – August 2009
Intent of Hack

- Defacement/Planting Malware: 28%
- Information Leakage/Stealing Sensitive Data: 26%
- Disinformation: 19%
- Monetary Loss: 11%
- Downtime: 4%
- Link Spam: 4%
- Phishing: 2%
- Other: 6%

Source: The Web Hacking Incidents Database Report – August 2009
## Mapping from 2007 to 2010 Top 10

<table>
<thead>
<tr>
<th>OWASP Top 10 – 2007 (Previous)</th>
<th>OWASP Top 10 – 2010 (New)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2 – Injection Flaws</td>
<td>A1 – Injection</td>
</tr>
<tr>
<td>A1 – Cross Site Scripting (XSS)</td>
<td>A2 – Cross Site Scripting (XSS)</td>
</tr>
<tr>
<td>A7 – Broken Authentication and Session Management</td>
<td>A3 – Broken Authentication and Session Management</td>
</tr>
<tr>
<td>A4 – Insecure Direct Object Reference</td>
<td>A4 – Insecure Direct Object References</td>
</tr>
<tr>
<td>A5 – Cross Site Request Forgery (CSRF)</td>
<td>A4 – Insecure Direct Object References</td>
</tr>
<tr>
<td>&lt;was T10 2004 A10 – Insecure Configuration Management&gt;</td>
<td>A5 – Cross Site Request Forgery (CSRF)</td>
</tr>
<tr>
<td>A10 – Failure to Restrict URL Access</td>
<td>A6 – Security Misconfiguration (NEW)</td>
</tr>
<tr>
<td>&lt;not in T10 2007&gt;</td>
<td>A7 – Failure to Restrict URL Access</td>
</tr>
<tr>
<td>A8 – Insecure Cryptographic Storage</td>
<td>A8 – Unvalidated Redirects and Forwards (NEW)</td>
</tr>
<tr>
<td>A9 – Insecure Communications</td>
<td>A9 – Insecure Cryptographic Storage</td>
</tr>
<tr>
<td>A10 – Malicious File Execution</td>
<td>&lt;dropped from T10 2010&gt;</td>
</tr>
<tr>
<td>A6 – Information Leakage and Improper Error Handling</td>
<td>&lt;dropped from T10 2010&gt;</td>
</tr>
</tbody>
</table>
Vulnerabilities and Threats

- **Passwords**
  - Default / no passwords / easy to guess passwords ..

- **Misconfiguration (at all levels)**
  - default / samples / improper ACLs ..

- **Buffer overflows**
  - Badly written applications either provided by vendor or otherwise

- **Web Application Vulnerabilities**
  - SQL Injection / XSS ..

- **Search Engine**
  - Search engines help find all of the above problems and more, much more ...
Threats Considerations @ all Layers

WhiteHat Security

“well-known” vulnerabilities

Customer Web Applications
- Business Logic Flaws
- Technical Vulnerabilities

Third-Party Web Applications
- Open Source / Commercial

Web Server
- Apache / Microsoft IIS

Database
- Oracle / MySQL / DB2

Applications
- Open Source / Commercial

Operating System
- Windows / Linux / OS X

Network
- Routers / Firewalls

2006 © Copyrights WhiteHat Security
Google Hacking Basics

Putting advanced operators together in intelligent ways can cause a seemingly innocuous query...

- Work with URLs to construct faster and smarter queries
  - "inurl:" , "site:"
- You can use "filetype:"
  - Exclude common file types with "-ext:"
- Use "intitle:" if you are looking for a very specific setting
  - Or "intext:"

http://johnny.ihackstuff.com
"# mysql dump" filetype:sql

A standard example of an easily found config file that contains too much info is filetype:ini inurl:ws_ftp

You may find "actual" databases through Google as well

  filetype:mdb
An example of ws_ftp.ini file

Usernames and Passwords
My Brief Attempt @ Google Hacking

- inurl:admin intext:ssn filetype:xls
  - Student Roster.xls
  - Student Grade.xls

- inurl:admin intext:password filetype:xls
  - Password.xls
  - StudentDetails.xls
Files with username, passwords, SSN, etc.
Login pages....default settings
Printers
Registry settings
Webcams
Ask the bad guys.....
Or at least try and think like them!
Google hacking made easy

http://johnny.ihackstuff.com
Google hacking made easy ......
<table>
<thead>
<tr>
<th>Date</th>
<th>Title</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>2003-06-24</td>
<td>squid cache server reports</td>
<td>These are squid server cache reports. For example, an institution stands...</td>
</tr>
<tr>
<td>2003-06-24</td>
<td>Ganglia Cluster Reports</td>
<td>These are server cluster reports. Great for info gathering. Lessie, what were those server names again?...</td>
</tr>
<tr>
<td>2003-06-24</td>
<td>ICG chat logs, please...</td>
<td>ICG (<a href="http://www.icq.com">http://www.icq.com</a>) allows you to store the contents of your online chats into a file. These folks have their entire ICG directories online. On p...</td>
</tr>
<tr>
<td>2003-06-24</td>
<td>Financial spreadsheets: finance.xls</td>
<td>&quot;Hey! I have a great idea! Let's put our finances on our website in a secret directory so we can get to it whenever we need to!&quot;...</td>
</tr>
<tr>
<td>2003-06-24</td>
<td>Financial spreadsheets: finance.xls</td>
<td>&quot;Hey! I have a great idea! Let's put our finances on our website in a secret directory so we can get to it whenever we need to!&quot;...</td>
</tr>
<tr>
<td>2003-06-24</td>
<td>SQL data dumps</td>
<td>SQL database dumps. LOTS of data in these. So much data, in fact, I'm pressed to think of what else an evil hacker would like to know about a target...</td>
</tr>
<tr>
<td>2003-06-24</td>
<td>mt-dospace.cgi files</td>
<td>These folks had the technical prowess to unpack the movable type files, but couldn't manage to set up their web servers properly. Check the mt.cf...</td>
</tr>
<tr>
<td>2003-06-24</td>
<td>AIM buddy lists</td>
<td>These searches bring up common names for AOL Instant Messenger &quot;buddy lists&quot;. These lists contain screen names of your &quot;online buddies&quot;...</td>
</tr>
<tr>
<td>2003-06-24</td>
<td></td>
<td>This brings up sites with PHPmb(). There is a SQL dump and even more that you find...</td>
</tr>
</tbody>
</table>
Google Search: "Dumping data for table"

Admin rates this entry 5 out of 10.
Added by: Admin
Hits: 4268
Score: 5

SQL database dumps. LOTS of data in these. So much data, infact, I'm pressed to think of what else an evil hax0r would like to know about a target database.. What's that? Usernames and passwords you say? Patience, grasshopper.....

Comments:
2004-07-06 09:46:30 (murfie):
Try "Dumping data for table (username|user|users|password)" -site:mysql.com -cvs > for clean results..

Thanks to ThePsyko and Johnny for this one!
Easy way to find some interesting data…….

You can use the same searches to look for data leakage by adding site:arizona.edu
Policies and Guidelines
House Bill 2043 - An Internet Web Site provided by the state shall contain a privacy statement to disclose the information gathering and dissemination practices related to the Internet. The Privacy Statement shall describe at a minimum the following:

- Notice regarding what services the web site provides
- A person’s ability to choose to proceed with the transaction and the alternatives available
- Who has access to the information the person provides
- What security measures are in place to protect the person’s private information and what information will be protected.

http://security.arizona.edu/privacy_statement
UA Department Privacy Statements

Departments that have written their own

- Office of Enrollment Management
  https://admissions.arizona.edu/policy/privacy.aspx

- Student Unions
  http://www.union.arizona.edu/privacy.php

- College of Agriculture and Life Sciences
  http://ag.arizona.edu/general/privacy.html

- Human Resources
  http://www.hr.arizona.edu/09_rel/privacy.php

- Bookstore (link at bottom left of page)
  http://www.uofabookstores.com/uaz/
Departments that have written an intro paragraph or two and then linked to UA’s

- Southwest Asthma & Allergy (AHSC)
  http://allergy.peds.arizona.edu/southwest/

- Library
  http://www.library.arizona.edu/about/access/privacy.html

Departments that link directly to UA Privacy Statement

- Registrar
  http://security.arizona.edu/index.php?id=857
UA Information Security Policy

UA’s Information Security Policy

- Minimum Security Standards
  - Application Security
  - Business Continuity and Disaster Recovery
  - Data Classification
    - Confidential University Data (CUD)
      - Personal Identifiable Information
        - Personal Information (Arizona Breach Notification)
      - Proprietary information
      - Confidential Non-Personally Identifiable Information
  - Incident Handling
  - Networked Devices
  - Server
Federal & State Legislation

- FERPA (expanded to include digital)
- HIPAA (applies to Business Partners)
- A.R.S 44-7501 – Arizona Breach Notification
- Payment Card Industry – Data Security Standard (PCI-DSS)
- Privacy Act of 1974
  - SSN Usage
- GLBA
For Application Developers

Tools
- Web Application Security Assessment Tool - Automated security and compliance assessment software for checking web applications for common vulnerabilities. It can be used in test, development and production instances to find all linked pages and to check sites for such vulnerabilities as SQL injection, cross-site scripting and buffer overflows.
- Vulnerability Scanning Tool - Automated vulnerability scanning software for identifying known vulnerabilities on your web server

Training Materials
The biggest threat to UA’s network security comes from its public websites and the web-based applications found there. A public website is generally accessible to anyone who wants to view it, making application security an issue. Vulnerabilities in web applications have inevitably attracted the attention of recreational and criminal attackers, who have devised techniques to exploit the vulnerabilities. Attacks on the web application layer now exceed attacks on the network.

Developers can mitigate these risks by becoming educated on the threats to application security and designing applications with security in mind. The following free resources provide both general and platform-specific information:
- AvIT Security Training - Language-agnostic security training developed by senior EMU software engineers
- Microsoft Security Guidance Training for Developers - This Clinic presents topics related to the essentials of application security (the importance of application security, security development practices, security technologies and secure development guidelines), threat defense (the need for secure code, defending against memory issues, cross-site scripting, SQL injection, canonicalization issues, cryptography weaknesses, Unicode issues, and denial of services attacks) and best practices for writing secure code (secure development process, threat modeling, risk mitigation and security best practices). The emphasis is on generally applicable material, but includes some demonstrations for Microsoft Visual Basic, Microsoft Visual C++, etc. Requires online registration.
- Dept. of Home Security Approved Secure Software Course - This course covers secure programming practices necessary to secure applications against attacks and exploits. Topics covered include fundamental concepts of secure software development, defensive programming techniques, secure design and testing, and secure development methodologies. Requires online registration.

UA Presentations
- Web Application Security - Presentation by Ed Murphy, UFS, for ACFUDE (PDF)
- Web-tailer - Web Application Security [video presentation] (PowerPoint presentation)

CWESANS
- Top 25 Most Dangerous Programming Errors - CWES - Broader than OWASP Top Ten, covering more general concepts
- Web Application Security Assessments - SANS
- Web Application Auditing Over Lunch - SANS
- SANS Top 20 Internet Security Attack Targets - SANS
- Application Database Security - SANS

OWASP
- Open Web Application Security Project
Security Performance Audit Findings and subsequent Requirements

- Application Security Standard
- Mandatory Training for all web developers
  - Currently gathering content
  - Will depend on department ISL’s to determine who needs to take training
- List of “critical” web applications
  - Web application inventory
  - Prioritize Web Apps by Criticality
- Security Review
  - Process for regular security reviews