An Introduction to Software Security and CSC591

Laurie Williams
williams@csc.ncsu.edu

Agenda

• What’s the course about
• Course details
Software Security

• The idea of engineering software so that it continues to function correctly under malicious attack
  – Not firewalling vulnerabilities
  – Not reacting through “penetrate and patch”

• Most software riddled with design flaws (50-60%) and implementation bugs (40-50%)

• Need to understand and manage software-induced security risks

Vulnerabilities/Attacks

• Vulnerability\(^1\) - a fault in software that allows an attacker to use it to violate a reasonable security policy for that system.

• Attack\(^2\) - techniques that attackers use to exploit the vulnerabilities in applications.

\(^1\) http://cve.mitre.org/about/terminology.html
\(^2\) https://www.owasp.org
**Trends in Software Security**

### CERT Cataloged vulnerabilities

- **Catalogued vulnerabilities**
- **Prorated**

From [http://www.cert.org/stats/](http://www.cert.org/stats/)

---

**Software Security**

Three pillars of software security

1. Risk management framework
2. Touchpoints
3. Knowledge

Software Security Touchpoints

Numbered according to effectiveness and importance or their “natural utility” per Gary McGraw in “Software Security”

Touchpoints (1 of 4)

1. Code review with a tool
   - Artifact: Code
   - Static analysis tools to identify common vulnerabilities
   - Example: buffer overflow

2. Architectural risk analysis
   - Artifact: Design and specification
   - Inspections and analysis approaches
   - Example: lack of authentication
Touchpoints (2 of 4)

3. Penetration testing
   – Artifact: Running system
   – Use application like an attacker
   – Example: Information leakage

4. Risk-based security testing
   – Artifact: Running system
   – Security features and attack patterns/misuse cases
   – Example: Elevation of privilege

Touchpoints (3 of 4)

5. Abuse cases
   – Artifact: Requirements
   – Think like an attacker
   – Example: Spoofing

6. Security requirements
   – Artifact: Requirements
   – Functional security requirements and emergent security properties
   – Example: No explicit description of necessary data protection
Touchpoints (4 of 4)

7. Security operations
   – Artifact: Fielded system
   – Network security, firewalls
   – Example: Insufficient logging to identify successful attacker

Course Progression

• Introduction
• Code-level vulnerabilities
• Design/architectural-level vulnerabilities
• Finding vulnerabilities
• Preventing vulnerabilities
• Risk analysis/management
Black Hat / White Hat

White hat:
Constructive
- Design
- Defense
- Security functionality

Black hat:
Destructive
- Attacks
- Exploits
- Breaking in!

http://vig-fp.pearsoned.co.uk/bigcovers/0321356705.jpg

Open Seminar (http://openseminar.org/se/)
Grade distribution

- 25% Midterm I (no unexcused)
- 25% Midterm II (no unexcused)
- 15% Attendance, Participation, Worksheet completion
- 10% Article Presentation (15 minutes with questions; provide one-page study guide)
- 25% Group Project (3-4 people)
Tell me and I forget.  
Teach me and I remember.  
Involve me and I learn.  

- Chinese Proverb

VCL: WebGoat

VCL: Healthcare IT Systems

- OpenMRS
- OpenEMR
- Tolven
- PatientOS

Questions about the syllabus . . . the class in general
?????????
Context and References

• OWASP
  - http://www.owasp.org/
  - 501c3 not-for-profit worldwide charitable organization focused on improving the security of application software

• SANS Institute
  - http://www.sans.org/
  - A huge training organization

• Common Weakness Enumeration (CWE)
  - http://cwe.mitre.org
  - 700 programming errors, design errors, and architecture errors that can lead to exploitable vulnerabilities
  - MITRE & US Department of Homeland Security’s National Cyber Security Division

http://cwe.mitre.org/
[2011] CWE/SANS Top 25 Most Dangerous Programming Errors

- (http://cwe.mitre.org/top25/)
- Often easy to find; easy to exploit
- Often allow attackers to completely take over the software, steal data, or prevent the software from working at all
- “Monster Mitigations” provided
  - Strategies for reducing or eliminating entire groups of the Top 25 weaknesses

Other Important Resources

- CERT (Resources)
  - http://www.cert.org/
- Build Security In (Resources)
  - https://buildsecurityin.us-cert.gov/bsi/home.html
- National Vulnerability Database (NVD)
  - http://nvd.nist.gov/
  - official vulnerability information on all known computer vulnerabilities (List of CVE with CVSS score)
- Common Vulnerabilities and Exposures (CVE)
  - http://cve.mitre.org/
  - dictionary of publicly known vulnerabilities with standard identifier number with status indicator, a brief description, and references to related vulnerability reports and advisories
- Common Vulnerability Scoring System (CVSS)
  - http://nvd.nist.gov/cvss.cfm
  - quantitative model for communicating the characteristics and impacts (severity) of IT vulnerabilities