|  |  |
| --- | --- |
|  | **GRADUATE SCHOOL OF ENGINEERING AND MANAGEMENT**  **Department of Electrical and Computer Engineering**  **CSCE 526: Secure Software Design and Development**  **Course Syllabus**  **FA 2017** |

|  |  |
| --- | --- |
| **Meeting Times** | Tue and Fri, 1000-1200 |
| **Location** | Bldg 646, Rm 212 |
| **Instructor** | Dr. James Okolica, ctr, Assistant Professor of Computer Science |
| **Office Location** | Bldg 640, Rm 313A |
| **Office hours** | The hour following in-class meetings (by Outlook appointment) |
| **Contact Information** | [james.okolica.ctr@afit.edu](mailto:james.okolica.ctr@afit.edu), [jsokolica@yahoo.com](mailto:jsokolica@yahoo.com) [CSCE 526] in subject line |

**Course Description:**

This course provides instruction for secure development of software. We place emphasis on managerial and technical aspects of software development. This includes coverage of federal software development policies, software design and implementation decisions, and numerous errors software developers make when writing source code. A comprehensive project highlights the course, providing valuable insight into how malicious hackers use poorly written software to gain control of a computer, and potentially, an entire network.

|  |  |
| --- | --- |
| **Credits** | 4.0 |
| **Prerequisites** | None, C programming |

**Student Learning Objectives:**

|  |  |
| --- | --- |
| **1** | Identify security vulnerabilities in software and design specifications. |
| **2** | Write secure software and design specifications. |
| **3** | Develop the ability to quickly read and critically analyze peer reviewed papers. |
| **4** | Identify and succinctly describe key characteristics of a specific lesson. |

**Required Books and Resource Materials:**

* Howard, M., LeBlanc, D., & Viega, J. (2010). 24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them (1st ed.). New York, NY, USA: McGraw-Hill, Inc.
* Laptop/ Tablet that you can bring to class
  + <https://www.onlinegdb.com> <https://app.codesignal.com> <https://scholar.google.com>

**Recommended/Optional Books and Resource Materials:**

* Deitel, How to program C (6 ed or newer)

**Policies:**

1. **Attendance:** Attendance at all class sessions and exams is mandatory for military and civilians assigned to AFIT as full-time students except for extenuating circumstances. Scheduled classes and exams are defined by the instructor and they are documented in the course schedule. Part-time students are expected to attend scheduled classes, and absences should be explained to the instructor. The student should provide advance notice, if possible. (References: Student Handbook, Graduate School Catalog)
2. **Academic Integrity:** All students must adhere to the highest standards of academic integrity. Students are prohibited from engaging in plagiarism, cheating, misrepresentation, or any other act constituting a lack of academic integrity. Failure on the part of any individual to practice academic integrity is not condoned and will not be tolerated. Individuals who violate this policy are subject to adverse administrative action including disenrollment from school and disciplinary action. Individuals subject to the Uniform Code of Military Justice may be prosecuted under it. Violations by government civilian employees may result in administrative disciplinary action without regard to otherwise applicable criminal or civil sanctions for violations of related laws. (References: Student Handbook, ENOI 36 – 107, *Academic Integrity*)
3. **Academic Grievance:** AFIT and the Graduate School of Engineering and Management affirm the right of each student to resolve grievances with the Institution. Students are guaranteed the right of fair hearing and appeal in all matters of judgment of academic performance. Procedures are detailed in ENOI 36 – 138, *Student Academic Performance Appeals*.
4. **Testing Policy:** Stated on the respective exams. Make-ups only with prior coordination.
5. **Late Assignments and Make-Ups:** Accepted only with prior coordination

**Grading Scheme/Policy:**

Capstone Paper or Game (Synthesis) 45%

Midterm (Theory) 20%

4-Phase Code Review Exercise (Application) 15%

Article Reviews 12%

In-class work (Participation & Effort) 8%

Final course grades will be no stricter than the following scale:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **A-** | **B+** | **B** | **B-** | **C+** | **C** | **C-** |  | **D** | **F** |
| **93+** | **90+** | **87+** | **83+** | **80+** | **77+** | **73+** | **70+** |  | **65+** | **<65** |

**Syllabus Schedule:**

*Course assignments, due dates and other requirements may be subject to change.* The course syllabus is a general plan for the course; deviations announced to the class by the instructor may be necessary.

|  |  |  |
| --- | --- | --- |
| **CLASS WEEK** | **IN-CLASS TOPICS** | **ASSIGNMENTS**  **(Out-of-class)** |
| Week 1  2-Oct | Introductions,  Software Engineering,  and  Secure Software Design | **Create an Account on C Arcade and GDBOnline**  [**https://app.codesignal.com**](https://app.codesignal.com)[**https://www.onlinegdb.com**](https://www.onlinegdb.com)  **Background Reading (Secure Software Design):**  Keshav, S., “How to Read a Paper”, 2007.  Purugganan, M. and Hewitt, J., “How to Read a Scientific Article”, 2004.  **Background Reading (Secure Software Design):**  Bell, E., “Secure Computer Systems: Mathematical Foundations”, 1973.  Saltzer, J and Schroeder, M., “The Protection of Information in Computer Systems”, 1975.  **Background Reading (Serious Games):**  Long, D. and Mulch, C., “Interactive Wargaming – CyberWar: 2025”, 2017.  **Preparatory Reading (Software Engineering/ Secure Software Design):**  Andreessen, “Why Software is Eating the World”, 2011.  McGraw, “Software Security”, 2004.  Whittaker, “Software’s Invisible Users”, 2001.  **Critical Reading (Course):**  Shostack, “Elevation of Privilege: Drawing Developers into Thread Modeling”, 2014.  Denning, T. et al., “Control-Alt-Hack: The Design and Evaluation of a Card Game for Computer Security Awareness and Education”, 2013. |
|  | Software Engineering,  Secure Software Design  and  C-Arcade/ GDBOnline | **Critical Reading:**  Tsipenyuk, K et al, “Seven Pernicious Kingdoms: A Taxonomy of Software Security Errors”, 2005.  **Preparatory Reading:**  Howard et al., “Sin 5: Buffer Overruns”, 2010.  Bright, “How Security Flaws Work: The Buffer Overflow”, 2015.  Aleph One, “Smashing the Stack for Fun and Profit”, 1996. |
| Week 2  9-Oct | Buffer Overrruns (Sin 5) | **Preparatory Reading (Buffer Overruns):**  Pincus, “Beyond Stack Smashing: Recent Advances in Exploiting Buffer Overruns”, 2004.  **Critical Reading (Buffer Overruns):**  Shacham, H., et al. “On the Effectiveness of Address-Space Randomization”, 2004.  Wagner, “A First Step Towards Automated Detection of Buffer Overrun Vulnerabilities”, 2000.  **Preparatory Reading (Format String Problems):**  Howard et al., “Sin 6: Format String Problems”, 2010.  Scut, Team Teso, “Exploiting Format String Vulnerabilities”, 2001.  Thompson, “Reflections on Trusting Trust”, 1984. |
| 12-Oct | Buffer Overruns and  Format String Problems | **TOPICS AND INTRODUCTION DUE**  **Critical Reading (Format String Problems):**  Cowan et al., “Format Guard: Automatic Protection From printf Format String Vulnerabilities”, 2001.  Shankar et al., “Detecting Format Strings Vulnerabilities with Type Qualifiers”, 2001. |
| Week 3  16-Oct | Format String Problems | **Preparatory Reading (Integer Overflows)**:  Howard et al., “Sin 7: Integer Overflows”, 2010.  Dietz et al., “Understanding Integer Overflows in C/C++”, 2015. |
| 19-Oct | Integer Overflows | **Critical Reading (Integer Overflows):**  Wang et al., “IntScope: Automatically Detecting Integer Overflow Vulnerability in x86 Binary Using Symbolic Execution”, 2009.  **Preparatory Reading (Poor Usability and Updates):**  Howard et al., “Sin 14: Poor Usability”, 2010.  Howard et al., “Sin 15: Not Updating Easily”, 2010.  Yee, “Aligning Security and Usability”, 2004. |
| Week 4  23-Oct | Integer Overflows  and  Poor Usability | **Preparatory Reading (Error Handling):**  Howard et al., “Sin 11: Failure to Handle Errors Correctly”, 2010.  Howard et al., “Sin 9: Catching Exceptions”, 2010.  **Critical Readings (Poor Usability):**  Schultz et al., “Usability and Security: An Appraisal of Usability Issues in Information Security Methods”, 2001.  Braz, C and Robert, J.M., “Security and Usability: The Case of the User Authentication Methods”, 2006. |
| 26-Oct | Poor Usability  and  Error Handling | **Preparatory Reading (Executing Code with Too Much Privilege):**  Howard et al., “Sin 16: Executing Code with Too Much Privilege”, 2010. |
| Week 5  30-Oct | Executing Code with Too Much Privilege  and  Code Review Exercise Phase 1  **(participation)** | **LITERATURE REVIEW/ GAME RULES DUE** |
| 2-Nov | **Guest Lecture?**  Code Review Exercise Phase 1  **(participation)**  [Dr. O. on leave] | **Preparatory Reading:**  Howard et al., “Sin 1: SQL Injection”, 2010.  Howard et al., “Sin 10: Command Injection”, 2010. |
| Week 6  6-Nov | SQL Injection,  and  Command Injection | **CODE REVIEW EXERCISE PHASE 1 DUE**  **Critical Reading (Command Injection):**  Su, Z and Wasserman, G., “The Essence of Command Injection Attacks in Web Applications”, 2006.  **Preparatory Reading (Information Leakage):**  Howard et al., “Sin 12: Information Leakage”, 2010. |
| 9-Nov | Information Leakage | **Critical Reading (Information Leakage):**  Chaabane, A., “You Are What You Like! Information Leakage Through Users’ Interests”, 2012.  Ristenpart, T., et al., “Hey You, Get Off My Cloud: Exploring Information Leakage in Third Party Compute Clouds”, 2009.  **Preparatory Reading (Cryptographic Sins):**  Howard et al., “Sin 17: Failure to Protect Stored Data”, 2010.  Howard et al., “Sin 19: Use of Weak Password-Based Systems”, 2010.  Howard et al., “Sin 20: Weak Random Numbers”, 2010.  Howard et al., “Sin 21: Using Cryptography Incorrectly”, 2010.  Wisnewski, C., “NIST’s New Password Rules – What You Need to Know”, 2016. |
| Week 7  13-Nov | Cryptographic and  Data Storage  Sins | **CODE REVIEW EXERCISE PHASE 2 DUE**  **Background Reading (Software Security):**  NIST 2010 – 800-37 - Guide for Applying the Risk Management Framework to Federal Information Systems  NIST 2010 – 800-39 - Managing Information Security Risk  Mills, S and Denman, T., “The Cybersecurity and Acquisition Life-Cycle Integration Tool”, 2017.  Shaud, J. et al., “The Quest for Defense Cybersecurity”, 2017  **Preparatory Reading (Cryptographic Sins):**  Howard et al., “Sin 13: Race Conditions”, 2010. |
| 16-Nov | Race Conditions | **Preparatory Reading (Software Security Testing):**  “Inside the Windows Security Push”, 2003.  McGraw, G., “Software Security Testing”, 2004.  McGraw, G., “Software Penetration Testing”, 2005.  McGraw, G., “Testing for Security During Development: Why We Should Scrap Penetrate-and-Patch”, 1997.  McGraw, G., “Misuse and Abuse Cases: Getting Past the Positive”, 2004.  Alexander, I., “Misuse Cases: Use Cases with Hostile Intent”, 2003. |
| Week 8  20-Nov | Software Security Testing | **CODE REVIEW EXERCISE PHASE 3 DUE**  **PAPER (and game if appropriate) DUE**  **Critical Reading (Software Security Testing):**  Manadhata, P. and Wing, J., “An Attack Surface Metric”, 2011. |
| 23-Nov | *Thanksgiving –* **IN-CLASS MIDTERM** | **Preparatory Reading (Web Application Sins):**  Howard et al., “Sin 2: Web Server-Related Vulnerabilities”, 2010.  Howard et al., “Sin 3: Web Client-Related Vulnerabilities”, 2010.  Howard et al., “Sin 4: Use of Magic URLs, Predictable Cookes, and Hidden Form Fields”, 2010.  Howard et al., “Sin 18: The Sins of Mobile Code”, 2010. |
| Week 9  27-Nov | Web Application and  Mobile Code  Sins | **Inform me by the end of class if you plan to submit a paper revision** |
| 30-Nov | Trust  and  Code Review Post-Mortem | **CODE REVIEW EXERCISE PHASE 4 DUE** |
| Week 10  4-Dec | **STUDENT PRESENTATIONS** | **(Optional Paper Re-submission)** |
| 7-Dec | **STUDENT PRESENTATIONS** |  |