Introduction to Secure Software Architecture

Anil Somayaji and Kenneth Ingham

October 1, 2009

1 Course overview

This course is designed to teach software architects the basics of how to create secure software systems. The emphasis is how the organization, features, and interfaces of an application influence its security. General security principles and specific design strategies are discussed. Case studies of successful and unsuccessful designs from the commercial and open source world are presented.

2 Course objectives

Add course objectives here.

3 Student background

If you are attending this class, then we assume that Students should be experienced software architects who are looking to create more secure software.

4 Logistics

The class lasts three days. No class os specified. The class uses the following software:

- Firefox (on Linux distribution but needed for Windows)
- Missing from arch-embedded.tex

No class network information specified.

The class needs a web server for the class web site. The instructor's laptop may be this web server; otherwise the machine provided in the classroom for the instructor is a good choice. This machine obviously will need web server software installed.

5 Class outline

1. Introduction (Lecture: 15; Lab: 0)
   (a) Class Introductions
   (b) Class Logistics
      i. Class schedule
      ii. Breaks
      iii. Question policy
      iv. Break room and restroom locations
      v. Assumptions about your background
   (c) Typographic conventions
   (d) What the class covers
2. Secure software architecture (Lecture: 50; Lab: 30)
   (a) What is a secure program?
   (b) Why is Security Important?
   (c) The Challenge
   (d) Approaches
   (e) The Importance of Architecture
   (f) Approach of this class
   (g) Key Lessons
   (h) Lab
3. Threat models and risk management (Lecture: 45; Lab: 55)
   (a) Introduction
      i. Example
   (b) The threat model
   (c) The assets you are protecting
   (d) Attackers
   (e) Common attack goals
   (f) Mitigating threats
   (g) Examples
      i. Password vault
      ii. Web-based timesheet
   (h) Risk analysis
   (i) Failures of Imagination
   (j) Summary
   (k) Lab
4. Cryptography Overview (Lecture: 70; Lab: 55)
   (a) Introduction
      i. Cryptographic Applications

    ii. Open design
  (b) Cryptographic Primitives
    i. Cryptographic hash functions
    ii. Symmetric key encryption
    iii. Public key encryption
  (c) Digital signatures
  (d) Public Key Management
    i. The Problem
    ii. Certificates
    iii. Trust Models
    iv. Example: PGP/GnuPG
    v. Example: SSL/TLS
    vi. Overview
      A. The server
      B. The client
  (e) Random numbers
  (f) Parameter sizes
  (g) Insecure Cryptography
    i. Key management errors
  (h) Do not innovate in cryptography
  (i) Summary
  (j) Lab

5. Programming Languages (Lecture: 45; Lab: 45)
  (a) Introduction
  (b) Specifications
  (c) Architecture
  (d) Interfaces
  (e) Implementation
  (f) Domain-specific languages
  (g) Discussion

6. C (Lecture: 45; Lab: 45)
  (a) Introduction
  (b) History
  (c) Requirements
  (d) Threat Model
  (e) Security Architecture
  (f) Vulnerabilities & Exploits
  (g) Responses
  (h) Analysis
  (i) Lessons

(j) Discussion

7. Java (Lecture: 45; Lab: 45)
   (a) Introduction
   (b) History
   (c) Requirements
   (d) Threat Model
   (e) Core Java Security Concepts
   (f) Java 1.0 Security Architecture
   (g) J2SE Security Architecture
   (h) J2ME and J2EE Security Architectures
   (i) Vulnerabilities & Exploits
   (j) Analysis
   (k) Lessons
   (l) Discussion
8. JavaScript (Lecture: 35; Lab: 45)
   (a) Introduction
   (b) History
   (c) Asynchronous JavaScript and XML (AJAX)
   (d) Requirements
   (e) Threat Model
   (f) Security Architecture
   (g) Vulnerabilities and Exploits
   (h) Responses
   (i) Analysis
   (j) Lessons
   (k) Discussion
9. Operating Systems (Lecture: 45; Lab: 45)
   (a) Introduction
   (b) Features
   (c) Components
   (d) Kernel
   (e) Processes & Threads
   (f) Virtual Memory
   (g) Libraries
   (h) Networking & IPC
   (i) Access Controls
   (j) Limitations
   (k) Future Directions
   (l) Discussion
10. UNIX and Linux (Lecture: 45; Lab: 45)

(a) Introduction
(b) History
(c) Requirements
(d) Threat Model
(e) Security Architecture
(f) Vulnerabilities & Exploits
(g) Analysis
(h) Lessons
(i) Discussion

11. Microsoft Windows (Lecture: 45; Lab: 45)

(a) Introduction
(b) Requirements
(c) Threat Model
(d) Security Architecture
(e) Vulnerabilities & Exploits
(f) Responses
(g) Analysis
(h) Lessons
(i) Discussion

12. Network Servers (Lecture: 45; Lab: 45)

(a) Introduction
(b) The Challenge
(c) Architecture prototypes
    i. Multithreaded
    ii. Multiple homogeneous processes
    iii. Multiple heterogeneous processes
(d) Persistent Storage
(e) Insights
(f) Discussion

13. Mail Transfer Agents (MTAs) (Lecture: 45; Lab: 45)

(a) Introduction
(b) History
(c) Requirements
(d) Threat Model
(e) Security Architecture
    i. *sendmail*
    ii. Exim
    iii. Postfix
    iv. qmail
    v. Microsoft Exchange

(f) Vulnerabilities & Exploits
(g) Analysis
(h) Lessons
(i) Discussion

14. OpenSSH (Lecture: 45; Lab: 45)
    (a) History
    (b) Requirements
    (c) Threat Model
    (d) Applying separation of privilege to OpenSSH
        i. Analysis
    (e) Lessons
    (f) Discussion
15. Application Architecture (Lecture: 45; Lab: 45)
    (a) Introduction
    (b) Networked Desktop Applications
    (c) Trusted vs. Untrusted Data
    (d) Active Content
    (e) Access Controls
    (f) Monolithic architecture
    (g) Cloud Computing
    (h) Future Directions
    (i) Discussion
16. Mozilla Firefox (Lecture: 45; Lab: 45)
    (a) Key Points
    (b) Analysis
    (c) Lessons
    (d) Discussion
17. Internet Explorer (Lecture: 45; Lab: 45)
    (a) Key Points
    (b) Analysis
    (c) Lessons
    (d) Discussion
18. Microsoft Office (Lecture: 45; Lab: 45)
    (a) Key Points
    (b) Analysis
    (c) Lessons
    (d) Discussion
19. Embedded system architecture (Lecture: 30; Lab: 45)

(a) Introduction
(b) Threats and attacks
(c) Issues affecting vulnerabilities
(d) Vulnerability case studies

20. Final lab (Lecture: 0; Lab: 120)
    (a) Lab