Week 4

Help

Building Secure Software

Most of our time so far has been spent focusing on implementation-level issues: *bugs* that constitute vulnerabilities, and means of avoiding those bugs, mitigating them, or recovering from them. But we must also be concerned about *flaws*, which are security problems in a software system's design.

To address both flaws and bugs effectively, we need to consider security through the entire development process. In this unit, we will step back and look at what that entails, taking on the goal of *building security in*.

Learning Objectives

After the completion of this week's material, you will be able to:

- Identify how security-minded thinking, or *security engineering*, is integrated into the software development process
- Enumerate a series of *design principles* for writing secure software
- Explain how such principles can be violated, pointing to actual incidents
- Put these principles into practice by drawing inspiration from well-designed, secure systems

Video Lectures

- Designing and Building Secure Software: Introduction (7:23)
- Threat modeling (Architectural risk analysis) (9:25)

- Security Requirements (13:56)
- Avoiding Flaws with Principles (8:12)
- Design Category: Favor Simplicity (10:50)
- Design Category: Trust With Reluctance (12:32)
- Design Category: Defense in Depth, Monitoring/Traceability (5:20)
- Top Design Flaws (9:18)
- Case Study: Very Secure FTP Daemon (12:24)

Break out: Interview with Gary McGraw

In August 2014, Mike had the pleasure of interviewing Gary McGraw. Gary is a celebrated author and authority on software security, which he practices professionally as the CTO of Cigital, Inc. In this interview we discussed many things relevant to this week's topic of secure design and secure development. The interview is *optional* from an assessment perspective -- there will no quiz questions on it (but note it may help cement topics in the lectures). We hope you find it interesting!

Mike Hicks interviews Gary McGraw (40:52). Highlights, indexed by time:

- start-3:47 Gary's background and early activities in software security
- 3:48-8:10 Software security: What, why, and who
- 8:10-14:55 Security throughout the development lifecycle: Secure *design*, not just secure coding
- 14:56-18:27 Moving towards more secure development practices
- 18:27-26:58 Building Security In Maturity Model (BSIMM)
- 26:58-32:42 Measuring security: How can BSIMM help?
- 32:42-37:19 Looking ahead: Improving secure design
- 37:19-40:52 Closing: Free time, art, music, and computer science

---

Supplemental Links

These links go into more depth about topics covered during lecture.

- Protection of Information in Computer Systems, by Saltzer and Shroeder. Classic paper from 1975 that is still highly relevant

today.

- Bruce Schneier's plea for simplicity in computer systems.
- DARPA's analytical framework for cybersecurity
- Top 10 Security Design Flaws, brought to you be the IEEE Center on Secure Design
- Very Secure FTPD; here is some description of VSFTPD's design, implementation, and assumptions of trust (and their impact on the implementation).
- Gary McGraw's book, Software Security: Building Security In

---

Quiz

The quiz for this week covers all of the material for this week. You must submit the quiz no later than the start of week 6. You will have three attempts to complete the quiz, at two hours per attempt. It consists of 16 questions, and if you are well versed in the material it should take no more than 30 minutes (but longer if you have to go back and look things up, obviously).

Project

There is **no new project** this week. Don't forget that Project 2 on exploiting web application vulnerabilities, issued last week, is still outstanding, and is due in two weeks.

Created Wed 9 Apr 2014 6:16 AM PDT

Last Modified Wed 19 Nov 2014 5:10 PM PST