

Building Trusted Software

A Practical Approach

Summary: This seminar is focused on “how” to write trusted software without requiring the team to become security experts.

There is an abundance of advice from security experts on “what” you should do. While this advice is generally sound it normally assumes you or your team needs to become security experts.

This seminar boils down all of this advice and provides recommendations for: 1) How to modify your existing processes. 2) How to achieve the goal of creating trusted software. 3) How to know when to bring in security help.

Intended Audience: Product and Project Managers
Architects

Prerequisites: Development and Validation team leads
No security experience is expected.
Some experience with software processes and tools.
The desire to improve the level of trust and security in the software you or your team produces.

Format: 14 Instructor led sessions and a final exam

Custom schedules with optional language specific sessions (Java/C/Web) are possible.

All sessions include discussion, case study, hands on labs, post session challenges and detailed reading lists.

Session 1: Current state of the art

Case study: Microsoft - The birth of SDLC
How does good working code become insecure?
Lab: The Perpetrator's Project Plan

Session 2: "The A's"

Case Study: OAuth - Session fixation attack
Authorization
Authentication
Access Control
Audit
Awareness
Attestation
Lab: Spoofing email

Session 3: Leveraging Hardware and Platform Features

Case Study: Intel - Hardware privilege escalation
Virtualization
Whole disk encryption
Trusted Processing Modules (TPM)
Remote PC management
Lab: Cracking Passwords

Session 4: Clouds and Browsers

Case Study: Twitter - CSRF attack
Cloud, SaaS and browsers
TLS/IP-SEC/IPv6
Lab: HTTP Fuzzing

Session 5: Software Development Lifecycle

Case Study: NIST - Software Quality Study
Models: Discuss how the selection of a software development model affects the delivery of trusted software. The following models will be examined and compared: (Waterfall, Spiral, RUP, and Agile)
Roles: Discuss how typical role definitions need to evolve to consistently deliver trusted software.
Lab: Planning Poker

Session 6: Process Improvement

Case Study: Intel - Retrospective Process

Gap Analysis: Organizations operate in certain modes with limited resources. This session will probe the gap between your current resources and what is needed to deliver trusted software.

General *models:* The following process improvements models will be examined and compared: (Benchmarking, Action Research, ISO 9000, Six Sigma, GQM and CMMI).

Security centric models: Inspired by Waterfall and CMMI models will be reviewed: (SDL, CLASP, Touchpoints, OpenSAMM, and BSIMM)

Lab: Running Retrospectives

Session 7: Trusted Computing Base (TCB)

Case Study: Federal Reserve - FedLine

Code Ownership: Compare approaches for securing code you own (internal) vs. code you use (external).

Client Trust: Discuss what needs to be considered when the client is outside your control.

Lab: XML digital signature UDF attack

Session 8: Threat Modeling

Case Study: Chaos Comm. Congress - Breaking MD5

Discuss how threat modeling can be employed to identify first, second and third order attacks.

Checklists and attack libraries

Lab: HTTP threat tree

Session 9: Static Analysis

Case Study: NIST - SAMATE Project

This session will discuss the use of static analysis tools, the types of errors they can detect. Common tool chain settings will be reviewed and the concept of proof carrying code presented.

Lab: "Hello World" code review

Session 10: Dynamic Analysis

Case Study: Fortify - "The death of penetration testing"

This session will discuss the use of dynamic analysis approaches like black box testing and runtime analysis.

Lab: Penetration testing with Nessus

Session 11:Validation

Case Study: Watchfire - Scanning for compliance.
Six basic principals will be presented in this session moving validation from checking for implementation of requirements to providing greater assurance of more reliable software.
Lab: Using HazOp to test for the negative.

Session 12:Privacy

Case Study: UCLA Medical Center - The Enquirer
Identifying and controlling Personally Identifiable Information (PII) and Protected Health Information (PHI).
Law enforcement access to data
Lab: Digital Forensics with Helix

Session 13:Standards, Regulations and Compliance

Case Study: Heartland and Hannaford - Data loss
Information security: ISO17799, NIST 800, ITIL, FIPS
Privacy: PCI-DSS, HIPPA
Audit: SOX, COBIT
Government: FISMA, DoD IA, DISA (STIGs)
Lab: Sniffing and XSS

Session 14:Survivability

Case study: W3C - Fixing broken HMAC crypto
Product Security Incident Response Teams (PSIRT)
Forum of Incident Response and Security Teams (FIRST)
Common Weakness Enumeration (CWE)
Computer Emergency Response Team (CERT)
Lab: Telling the world you made a mistake.

Session 15:Exam

The exam will contain short answer and case study type questions.