CONTENTS

CLOUD COURSES
Cloud Computing In the Business Environment.......................... 6
Cloud Governance and Security.............................. 7
Cloud Economics, Metrics and Migration
............................................................................. 8

CYBERSECURITY COURSES
Foundations of Secure Software .......... 11
Managing Secure Software Development
................................................................. 12
Secure Software Coding ..................... 13
Secure Software Design.................... 14
Art of Hacking................................. 15
Codiscop........................................... 17

  Introduction to Cryptography
  OWASP Top 10
  Developing Securely for PCI DSS
  Foundations of Mobile Security
  Foundations of iOS Security
  Foundations of Android Security
  Defensive Programming for iOS
  Defensive Programming for Android

EMBEDDED SYSTEMS & IoT COURSES
Embedded System................................. 27
Multi-Core Video Series .................... 28
Quarto............................................... 29
CLOUD COURSES
CLOUD COMPUTING IN THE BUSINESS ENVIRONMENT

Course Description:

Managers are often faced with having to decide if, and how to upgrade their IT infrastructure, and how to pay for it. In an environment of tight budgets and soaring hardware and software costs, they are also looking for alternatives to making huge investments that will have to be upgraded again and again. The Cloud can be that solution. Managers need information to make intelligent decisions however.

Questions pertaining to Cloud economics, security, regulation and governance, metrics and migration are introduced and discussed. In the final analysis, managers must be able to answer key questions – is the Cloud the right place for my IT infrastructure and data? Is it a good business decision? How do I migrate to the Cloud? This course introduces and discusses these and other key concepts.

Course Modules:

Module 1 – Introduction to the Cloud
- Language of Cloud Computing
- Provenance of Cloud Computing
- Today’s Players in Cloud Computing
- 2 Quiz Questions

Module 2 – Economics of the Cloud
- The Economics of the Cloud
- Measuring Net Present Value, Benefit Ratios, Payback Periods
- Where’s the Payback?
- 2 Quiz Questions

Module 3 – Migrating to the Cloud
- How do you characterize your organization?
- What are your mission critical systems?
- Evolution or Big Bang?
- 2 Quiz Questions

Module 4 – Business of Cloud Security
- Overview of principle security issues
- What kind of data can be entrusted to the Cloud?
- What kinds of data should not be entrusted to the Cloud?
- 2 Quiz Questions

Module 5 – Cloud Governance
- Managing Regulation within the Cloud
- 2 Quiz Questions

Self-Paced Learning  Approximate Time: 90 Minutes
PDH  1.5
CEU  0.15
Course Cost:  
Member: $45
Non-Member $65
Course Description:

Migrating IT infrastructures to the Cloud is becoming a cost effective solution to the ever increasing burden of maintaining complex software and hardware systems. As with any new concept that plays a central role in business, the Cloud must be understood. Specifically, the differences between managing an in-house IT infrastructure versus a Cloud-based infrastructure are critical.

Seeing the Cloud as a process rather than a product, implementing a governance plan and developing Cloud metrics are all part of Cloud governance.

Additionally, securing data and processes within the Cloud are central to any migration considerations. Issues such as risk profiles, what information should and should not be committed to the Cloud, roles and rules, and access control must be taken into account. This course will examine these and other key concepts central to Cloud governance and data security.

Course Modules:

Module 1 – Governance in the Cloud
• Introduction
• Key governance concepts
• Governance as a process vs. a product
• The essential elements

Module 2 – Implementing Governance in the Cloud
• Introduction
• Governance in SaaS, PaaS, IaaS
• Key governance metrics
• Managing the governance process
• Module 3 – Cloud Security
• Introduction
• The risk profile
• Understanding the risks of the Cloud
• Deciding what information belongs in the cloud

Module 4 – Maintaining Security
• Introduction
• Protecting your cloud based systems
• Who has access?
• Roles, rules and access
Course Description:

Deciding to migrate a business' IT infrastructure to the Cloud involves a great deal more than the realization that one may not have to purchase all of the software, hardware and IT infrastructure nominally required to run the business. First, is it a sound economic decision? How can we measure net present value? Benefit ratios? Payback periods?

Are there metrics we can use to determine if the Cloud is a viable solution? Measuring the Cloud requires that we examine issues of security, usage and response time in terms of metrics – can these be quantified? Once a decision has been made to migrate, how can we devise a plan? Questions like how can I characterize my business; what are my mission critical systems, and do I follow the evolution or the Big Bang concept must be addressed.

And, once there, how can a business establish sustainable operations within the Cloud?

These and other key issues with migrating to the Cloud are addressed for the business management team contemplating a move into the Cloud.

Course Modules:

Module 1 – Economics of the Cloud
- Introduction
- Net Present Value; Benefit Ratios; Payback Periods
- Common Infrastructure and Utility Pricing
- Economics of being Disconnected

Module 2 – Measuring the Cloud
- Introduction
- Security Metrics
- Usage Metrics
- Response Time Metrics

Module 3 – Cloud Migration Plan
- Introduction
- How do you characterize your organization?
- What are your mission critical systems?
- Evolution or Big Bang?

Module 4 – Sustainable Operations in the Cloud
- Introduction
- Now that you’re there, what do you do?
- Maintenance in the Cloud?
- System Administration?

Module 5 – Migrating to the Cloud
- How do you characterize your organization?
- What are your mission critical systems?
- Evolution or Big Bang?
- 2 Quiz Questions
CYBERSECURITY COURSES
Course Modules:

Module 1-Introduction to Software Security
- Significance/importance of software security
- Software security terminology
- Software security resources (both online and offline)

Module 2-Sources of software security threats
- Hardware level threats
- Code level threats
- Detailed design level threats
- Architecture level threats
- Requirements level threats

Module 3-Software security best practices
- Security requirements elicitation
- Secure architecture best practices
- Detailed secure design best practices
- Secure coding best practices
- Secure deployment and operations best practices

Module 4: Software security lifecycle
- General software lifecycle vs. secure software lifecycle
- Architectural risk analysis
- Measurement
- Project management

Module 5: Software security testing
- Code analysis
- White box testing
- Penetration testing
- Security testing tools

Module 6: Careers in SW Security – outlook and preparation

Course Description:
Software security is becoming increasingly important due to the numerous emerging threats exploiting software vulnerabilities. This course provides a broad overview of various software security threats and some of the most effective countermeasures used to thwart both well-known and newly emerging software security threats.

By taking this course, software practitioners will learn how to build security into their software products throughout its lifecycle. Although it is impossible to accomplish “perfect security” in software, the participants will be exposed to the best practices and tools available today to minimize their chance of falling victim to a common software security attack.
Course Description:

Whenever a software system is developed and deployed, there is always an entity that will attempt to corrupt or misuse that software. Software security is thus a critical thread that must run through the entire software development life-cycle. The development of secure software is the ultimate goal.

Software security involves a layered approach designed to deal with myriad threats and attack vectors based on an overall security policy that takes into account a variety of threat scenarios, the ROI of software security, the security supply chain, software assurance risk management and system evolution in an environment of never-ending threats. This course examines software security from the perspective of managing secure software development.

Course Modules:

Module 1 – Introduction to Secure Software Development
- Nature of software
- Software development life-cycle
- Software as a system
- Basics of security

Module 2 – Security in Detail
- Layered approach to security
- Threats and attack vectors
- Security policy and its importance
- Security from an ROI perspective
- Security supply chain

Module 3 – Software Security Risk Management
- Risk management overview
- Incident response
- In-house versus COTS
- Testing for security
- System evolution

Module 4 – Assurance Management
- Ownership issues
- Assurance management

Self-Paced Learning  Approximate Time: 200 Minutes

<table>
<thead>
<tr>
<th>PDH</th>
<th>3.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEU</td>
<td>0.34</td>
</tr>
<tr>
<td>Course Cost:</td>
<td></td>
</tr>
<tr>
<td>Member: $100</td>
<td></td>
</tr>
<tr>
<td>Non-Member $135</td>
<td></td>
</tr>
</tbody>
</table>
SECURE SOFTWARE CODING

Course Description:
Secure coding is becoming increasingly important due to various emerging threats. Although many of the existing secure coding best practices are programming language and domain-specific, there are also language and domain-neutral knowledge to be obtained by software developers. This course focuses on this attainment of comprehensive but practical secure coding knowledge. By exposing the audience to all the major secure coding concepts and tools available today, the primary goal of this course is to quickly improve the attendees’ overall security quality of coding practices and to prepare for learning more language and application-specific secure coding techniques.

Course Modules:

Module 1: Introduction to Secure Coding
• Significance/importance of secure coding [8, 11]
• Secure coding terminology [7, 12, 13]
• Secure coding principles [10, 15, 18, 20, 21]
• Threat modeling
• Secure coding resources (both online and offline) [14]

Module 2: Vulnerabilities and Exploits
• Buffer overflows
• SQL injection [2, 19]
• Cross-Site Scripting (XSS) [4, 9]
• Broken authentication and session management
• Insecure direct object references
• Security misconfiguration
• Sensitive data exposure

Module 3: Countermeasures
• Secure coding standards [3]
• Secure coding best practices/patterns [1, 16]
  • Intercepting validators
  • Sanitization
  • Session management
  • Authentication
  • Encryption
  • Password management
  • Access control
  • Error handling and logging
  • File management
  • Memory management

Module 4: Tools, Frameworks, and Services
• Microsoft Secure Development Process (SDP)
• Static analysis tools [17]
• Dynamic analysis tools
• Web application security frameworks [5, 6]
• Java-based enterprise application security frameworks
• Outsourcing
• Vulnerability tracking
• Module 5: Summary and Conclusion

Self-Paced Learning Approximate Time: 167 Minutes
PDH 2.8
CEU 0.28
Course Cost: Member: $100
Non-Member: $135
SECURE SOFTWARE DESIGN

Course Description:

This course offers a comprehensive coverage of practical knowledge in how to design secure software as well as insights on the significance of the role secure design plays during a software development life cycle. Some of the critical topics covered in this course include secure design principles and processes in addition to fundamental security concepts such as access control, encryption, etc. This course also devotes a significant amount of time to discussing well known secure design solutions including architectural patterns and design patterns focusing on security countermeasures and concludes with the discussion of software security analysis and evaluation as mechanisms to assess the effectiveness of the secure design solutions implemented in the form of source code.

Course Modules:

1. Secure Software Design Fundamentals
   1.1. Significance of Secure Software Design
   1.2. Secure Software Design Terminology
   1.3. Secure Software Design Process
   1.4. Secure Software Design Principles
2. Key Issues in Secure Software Design
   2.1. Security Control Types
   2.2. Access Control
   2.3. Encryption Methods
   2.4. Intrusion Detection and Prevention
   2.5. Recovery
   2.6. Accounting and Trust Management
3. Secure Software Structure and behavior
   3.1. Security Patterns
   3.2. Security Tactics
   3.3. Practical Examples
4. Software Security Analysis and Evaluation
   4.1. Analysis and Evaluation Techniques
   4.2. Measures and Metrics
5. Software Security and Formal Methods
   5.1. What are formal methods?
   5.2. Formal Methods in Secure Software Design
   5.3. Formal Methods and Common Criteria
   5.4. Formal Method Tools

Self-Paced Learning

<table>
<thead>
<tr>
<th>PDH</th>
<th>CEU</th>
<th>Course Cost:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.8</td>
<td>0.18</td>
<td>Member: $45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Non-Member: $65</td>
</tr>
</tbody>
</table>
ART OF HACKING COURSE

Course Description:
Protect Your Organization From Hackers By Thinking Like Them Hacker

As the worldwide cyber security threat increases daily, no company, large or small, is safe – and standard thinking won’t cut it when it comes to protecting your valuable IP, private data, and bottom line. Web developers, system administrators, SOC analysts, penetration testers, network engineers, and security innovators need to “think like the enemy” in order to anticipate the next moves and stay up to date on current threats.

Learning Objectives
1. Learn to Hack in Our Online, Real-World Lab
2. The full course is composed of five modules – three infrastructure hacking and two of web hacking, all at the introductory/intermediate level, but with significant challenge.
3. Certifications can also be received based upon demonstrated skill level – Art of Hacking Ninja and Art of Hacking Master Book Class
4. Classes are comprised of a theory side (25%) which teaches and tests knowledge of the subject
5. The real heart of the classes (75%) is the unmatchable HackLab where students get hands-on training and practice in the newest, most pertinent hacking techniques
6. All the newest cases, threats, hacks, and intrusions are continuously added to the HackLab to allow students to match themselves against the latest threats
7. Students have up to six months to complete e-learning and an additional six months for certification
8. Penetration testing by the experts at NotSoSecure can be added to assess and validate your company’s defenses and your team’s success in thwarting attacks

<table>
<thead>
<tr>
<th>Self-Paced Learning</th>
<th>Approximate Time: 588 Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>INF</td>
<td>331 min</td>
</tr>
<tr>
<td>WEB</td>
<td>257 min</td>
</tr>
<tr>
<td>PDH</td>
<td>9.8</td>
</tr>
<tr>
<td>CEU</td>
<td>0.98</td>
</tr>
</tbody>
</table>

Course Modules:

PART 1
- Module 1: The Art of Port Scanning
- Module 2: The Art of Brute-Forcing
- Module 3: The Art of Hacking Databases
- Module 4: Metasploit Basics

PART 2
- Module 5: Password Cracking
- Module 6: Hacking Unix
- Module 7: Hacking Application Servers
- Module 8: Hacking third party CMS software (WordPress, Joomla)

PART 3
- Module 9: Windows Enumeration
- Module 10: Hacking Third party software (browser, pdf, java)
- Module 11: Hacking Application Servers on Windows.
- Module 12: Post Exploitation- Dumping Secrets
- Module 13: Hacking Windows Domains

PART 4
- Module 14: Understanding HTTP protocol
- Module 15: Information gathering
- Module 16: Username Enumeration and faulty Password reset.
- Module 17: Issues with SSL/TLS
- Module 18: Authorization Bypass

PART 5
- Module 19: Cross Site Scripting (XSS)
- Module 20: Cross-Site Request Forgery (CSRF)
- Module 21: SQL Injection
- Module 22: XML External Entity (XXE) Attacks
- Module 23: Insecure File Uploads

<table>
<thead>
<tr>
<th></th>
<th>Member</th>
<th>Non-Member</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infrastructure Training – includes 3 courses: 1) Infrastructure Basics, 2) Hacking Unix, Databases and Applications, 3) Hacking Windows. Also includes: 1 year of the eLearning subscription and 7 days of lab time</td>
<td>$695</td>
<td>$895</td>
</tr>
<tr>
<td>Web Training – includes 2 courses: 1) Information Gathering, Profiling and Cross-Site Scripting 2) Injection, Flaws, Files and Hacks. Also includes: 1 year of the eLearning subscription and 7 days of lab time</td>
<td>$495</td>
<td>$695</td>
</tr>
<tr>
<td>Full Art of Hacking Training Course: Infrastructure + Web</td>
<td>$995</td>
<td>$1,295</td>
</tr>
<tr>
<td>Art of Hacking Certification: Infrastructure + Web</td>
<td>595</td>
<td>$795</td>
</tr>
<tr>
<td>Bundle of Full Training + Art of Hacking Certification</td>
<td>$1,495</td>
<td>$1,895</td>
</tr>
<tr>
<td>Extra Lab Time: 21 days</td>
<td>$250</td>
<td>$350</td>
</tr>
</tbody>
</table>
CODISCOPE
COURSES
INTRODUCTION TO CRYPTOGRAPHY

Course Description:

Cryptography is used to address issues of confidentiality, data integrity, data origin, authentication, entity authentication, and non-repudiation. Although cryptography does not eliminate security issues, it does make them more manageable by reducing the task of protecting a large amount of data to a matter of protecting a relatively small key. This course discusses the use of cryptographic algorithms and techniques as they are typically applied within the practice of information security.

Learning Objectives:

- Define cryptography and cryptographic primitives as they apply to software security practices
- Identify the most common cryptographic primitives and respective purposes
- Identify common cryptography errors and how to avoid them
- Make appropriate design decisions when implementing cryptographic controls into the information security process

Course Outline:

Cryptography and Cryptographic Primitives
- Uses of Cryptography

Common Cryptographic Primitives
- Encryption
  - Symmetric vs. Asymmetric Encryption
  - Common Types of Encryption
    - Block Ciphers and Stream Ciphers
    - Block Cipher Encryption Modes
    - Initialization Vectors
    - Block Cipher Padding Modes
    - Common Types of Asymmetric Key Encryption
  - Hash Functions
    - Cryptographic Hash Functions
    - Algorithms and Uses
    - Protecting Data Integrity
  - Message Authentication Codes (MAC)
    - Common Functions and Algorithms
    - How it Works
    - Problem and Solution
- Digital Signatures
  - Digital Signatures
  - Algorithms
  - Problem and Solutions
  - Putting It All Together – SSL

Security of Cryptographic Primitives and Protocols
- Cryptographic Primitive/Protocol Security
- Security of Algorithms Over Time
- Security Over Time – Lessons Learned

Choosing Your Cryptographic Primitives
- Typical Attackers and Attacks
  - Criminals
  - Kiddies/Amateur Hackers
  - Crime/Dedicated Hackers
  - Researchers
  - Government Agencies

Self-Paced Learning | Approximate Time: 90 Minutes
--- | ---
PDH | 1.5
CEU | 0.15

Course Cost:
- Member: $18
- Non-Member: $18
Course Outline:

The OWASP Top 10
- The Challenges of Software Security
- Classic Security Trade-offs

Injection
- Common Injection Vulnerabilities
- SQL, Command, and XML Injection
- Mitigations

Broken Authentication and Session Management
- Normal and Exploitation Workflows
- Brute Forcing, Session Fixation, and Session Hijacking
- Mitigation and Remediation Examples

Cross-Site Scripting
- Common Attacks and Exploitation
- Same Origin Policy and Malicious Script
- Stored and Reflected XSS
- How to Test for It
- Mitigation and Remediation

Insecure Direct Object References
- Exploitation and Examples
- How to Test for It
- Mitigation and Remediation

Security Misconfiguration
- Exploitation and Examples
- How to Test for Security Misconfiguration
- Mitigation and Remediation

Sensitive Data Exposure
- Exploitation and Attacks
- CIA of Information Security
- How to Test for It
- Mitigation and Remediation

Missing Function-Level Access Control
- Exploitation and Examples
- How to Test for Them
- Mitigation and Remediation

Cross-Site Request Forgery
- Exploitation and Example
- How to Test for It
- Mitigation and Remediation

Using Components with Known Vulnerabilities
- Exploitation and Example
- How to Test for It
- Mitigation and Remediation

Unvalidated Redirects and Forwards
- Exploitation and Example
- How to Test for It
- Mitigation and Remediation

Course Description:

Created for developers with experience developing web applications in any programming language, this course focuses on the most common security defects identified by the Open Web Application Security Project (OWASP). To accomplish this, the course describes in detail each item included in the 2013 OWASP Top 10 list. Each lesson describes a vulnerability and provides practical guidance for testing and remediation. Additionally, this course also presents practical walkthroughs that demonstrate how the vulnerabilities are exploited.

Learning Objectives:

- Describe the role of security in the software development lifecycle
- Strategize how best to create secure applications
- Recognize the details of and the causes behind secure coding errors and mistakes
- Describe how software security defects are exploited
- Efficiently utilize discovery methods for uncovering security defects
- Strategize practices to help prevent the most common mistakes and ultimately create more secure software

Self-Paced Learning | Approximate Time: 90 Minutes
--- | ---
PDH | 1.5
CEU | 0.15
Course Cost:
Member: $
Non-Member $
Course Outline:

Secure Coding Guidelines
- Introduction to the PCI DSS
  - PCI DSS Requirement 6.5
  - Software Security Vulnerabilities
  - Secure Software Development Lifecycle
- Injection Flaws
  - SQL Injection Exploitation
  - SQL Injection Remediation
  - Injection Mitigations
- Buffer Overflows
  - Understanding Buffer Overflow
  - Perform Proper Bounds Checking
  - Beware of Non-null-terminated Strings

Secure Coding Guidelines (continued)
- Insecure Cryptographic Storage
  - Testing for Insecure Cryptographic Storage
- Insecure Communications
  - Common Pitfalls
- Improper Error Handling
  - Stack Trace Example
- All “High Risk” Vulnerabilities
- Cross-Site Scripting
  - Common Attacks
  - Testing for It
  - Mitigation and Remediation
- Improper Access Control
  - Java Code Example
  - Testing for It
  - Mitigation and Remediation
- Cross-Site Request Forgery
  - Testing for It
  - Mitigation and Remediation
- Broken Authentication and Session Management
  - Exploitation
  - Common Broken Authentication Problems
  - Mitigation Remediation

Protecting Data in Memory
- Sensitive Account Data
- Sensitive Data in Memory
  - Data in Use as an Asset
  - Extracting Sensitive Data from RAM
- Managing Volatile Memory
- System-Level Extraction via Memory Dumps
- Forced Release of Sensitive Data
  - OS-Level Volatile Data Release
  - Secure Erase of Data After Use

Course Description:

Vulnerabilities to payment card security are a threat to everyone with a credit or debit card in their wallet. Every day, we effectively transmit highly personal and sensitive data about ourselves to strangers. If all goes well, only the intended recipients ever see our information. If not, the results can be disastrous. Thus the criticality of the Payment Card Industry Data Security Standard (PCI DSS). PCI DSS provides guidance to organizations that collect, process, transmit, or store cardholder data. In this course, you will learn about PCI DSS: the data it is intended to secure, its requirements, how to incorporate those requirements into code, and how to avoid common mistakes that can make your software vulnerable to attack.

Learning Objectives:

- Recognize which software security defects are addressed by PCI
- Strategize and utilize discovery methods for protecting sensitive cardholder data based on PCI guidance
- Recognize the role memory plays in the security of cardholder’s personal information
- Utilize PCI-guided best practices to avoid common mistakes and ultimately develop more secure software

Self-Paced Learning  Approximate Time: 60 Minutes

<table>
<thead>
<tr>
<th>PDH</th>
<th>CEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.10</td>
</tr>
</tbody>
</table>

Course Cost:
- Member: $
- Non-Member $
FOUNDATIONS OF MOBILE SECURITY

Course Description:
Mobile applications enable millions of users to be more productive, have more fun, and interact with their world in more ways than ever before. Mobile applications also expose users and their devices to a whole host of not-so traditional issues such as location disclosure, information theft, and call fraud. Approaches to securing mobile applications build upon many of the techniques used in more traditional software development; however, properly accounting for new threats and attacks requires a thorough understanding of mobile platforms.

Learning Objectives:
• Differentiate between traditional and mobile computing environments at a foundational level
• Generally compare common mobile application architectures
• Comfortably discuss the risk landscape affecting mobile architectures and approaches for mitigating those risks
• Identify benefits and limitations of common mitigations present on mobile platforms

Course Outline:
Understanding Mobile Architectures
- The Basic Mobile Ecosystem
  • Defining Mobile Devices
  • Actors of the Mobile Ecosystem
  • Cellular Carriers and Networks
  • Data and Functionality on Mobile Devices
- The Mobile Application Stack
- Approaches for Loading Applications
- Learn it! Attack Surface
- Mobile Device Attack Surface
- Common Mobile Platforms
- Mobile Application Architectures

The Mobile Application Risk Landscape
- Software Assurance vs. Software Attestation
- Risks Affecting Mobile Platforms
  • Physical Risks to Mobile Devices
  • User-approved Security Permissions
  • Buffer Overflow Issues
  • Injection Problems
  • Common Web Security Issues
  • Information Disclosure
  • Malware Attacks
  • Time-of-Check/Time-of-Use Problems
  • Unprotected APIs
  • Reverse Engineering
- Common Security Mechanisms on Mobile Platforms
  • Sandboxing
  • Code Signing
  • Application Permissions
- Defenses against Buffer Overflow Exploitation
- Protection against Physical Access
- The Mobile Application Stack versus Risks
- Managing Risks in the Mobile SDLC

Self-Paced Learning | Approximate Time: 60 Minutes
PDH | 1.0
CEU | 0.10
Course Cost:
Member: $ 
Non-Member $
Foundations of iOS Security

Course Description:

Apple’s iOS platform provides a rich set of features for creating versatile mobile applications. However, these applications are prone to security risks that are either iOS-specific or common to other types of mainstream software. The Foundations of iOS Security course gives an overview of the architecture of the iOS operating system and the security issues that affect iOS applications.

Learning Objectives:

- Explain the iOS platform architecture with emphasis on security
- Describe the application of the iOS security model
- Describe key aspects of iOS applications that are relevant to security
- Begin to identify common iOS application security risks

Course Outline:

The iOS Application

- iOS Architecture
  - Objective-C Runtime
  - Application
  - Cocoa Touch
  - Media
  - Core Services
  - Core OS
- Types of iOS Applications
  - HTML Container
  - Native Objective-C
  - Hybrid
- Application Package and Components
- Application Storage
  - File System
  - iCloud
  - Keychain
  - Server-Side Data Store
- Inter-Process Communication (IPC)

iOS Security Model and Services

- Platform Security Model
  - Encryption for System and File Protection
  - Mitigations against Memory Corruption Attacks
  - Application Sandboxing
  - Code Signing and Application Vetting Process
- Application-level Security Services
  - Data Protection API
  - KeyChain Services
  - Cryptography and Certificate Management
  - Additional Security Controls

Common iOS Application Risks

- Insecure Handling of URL Schemes
- Information Leakage and Insecure Data Storage
- Weak Authentication
- Improper Input Validation and Data Representation
- Memory Corruption Issues
- Lack of Server-Side Controls
- Weak Cryptography
- Web Security Issues
- Reverse Engineering
- Jailbreaking

Addressing Mobile Application Security Risks

Self-Paced Learning

<table>
<thead>
<tr>
<th>PDH</th>
<th>CEU</th>
<th>Course Cost:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.75</td>
<td>0.075</td>
<td>Member: $</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Non-Member $</td>
</tr>
</tbody>
</table>

Approximate Time: 60 Minutes

PDH 0.75
CEU 0.075
Course Cost:
Member: $
Non-Member $
Course Description:
The Google Android platform provides a rich set of features for creating versatile mobile applications. However, these applications are prone to security risks that are Android-specific or common to other types of mainstream software. The Foundations of Android Security course gives an overview of the architecture of the Android platform and the security-relevant aspects of it.

Learning Objectives:
• Comprehend the Android platform and the Android application architecture
• Comprehend the Android security model
• Describe key aspects of Android applications that are relevant to security
• Recognize common Android application security risks

Course Outline:
The Android Platform
• The Android Architecture
  • Applications
  • Application Framework
  • Native Libraries
  • Android Runtime
  • Linux Kernel
• The Android Package File
• Android Application Types
• Android Application Components
The Android Platform (continued)
• Inter-Process Communication
• Android Storage Options
The Android Security Model
• Android System and Kernel-Level Security
  • Sandboxing
  • File Permissions
  • File System Encryption
  • Memory Management
• Android Application Permissions
• Application Permission Levels
The Android Security Model (continued)
• Application Signing
• Device Rooting
Common Risks for Android Applications
• Android-Specific Risks
  • Permission Re-Delegation
  • Over-Privileged Applications
  • Unauthorized Intent Recipients
  • Intent Forging
  • Sticky Broadcasts
  • Insecure File Storage
  • Storing User Credentials on Client Side
• Common Application Security Risks
  • Insecure Network Communications
  • Information Disclosure
  • Lack of Strong Authentication
  • Injection Issues
  • Buffer Overflows
  • Lack of Server-Side Controls
  • Weak Cryptography
• Addressing Android Application Security Risks

Self-Paced Learning

<table>
<thead>
<tr>
<th>PDH</th>
<th>CEU</th>
<th>Course Cost:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.75</td>
<td>0.075</td>
<td>Member: $</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Non-Member $</td>
</tr>
</tbody>
</table>

Approximate Time: 60 Minutes
PDH 0.75
CEU 0.075
Course Cost:
Member: $
Non-Member $
Course Description:

iOS applications are affected by issues inherent to the iOS platform and issues shared with other types of mainstream software. Developers must be familiar with both types of risks to design and develop applications resilient to common attacks. Building on the Foundations of iOS Security course, this course teaches effective techniques for addressing common risks in iOS applications.

Learning Objectives:

- Identify security risks relevant to iOS applications
- Leverage native iOS controls and features in a secure manner
- Apply defensive programming techniques to mitigate common security risks in iOS applications

Course Outline:

Preventing Injection Issues
- Understanding Dataflow
- Defining Sources and Sinks
- Performing Input Validation
- White-listing, Black-listing, and Rostering
- Encoding Output
- Simulation: SQL Injection
- Mitigating SQL Injection

Preventing Data Leakage

Leveraging Channel Security

Secure Data Storage
- Performing Password-Based Encryption
  - Key Generation
  - Overall Encryption Process
  - Generating Entropy
  - Encryption Function Implementation

Selecting a Data Protection Approach
- Performing Strong Authentication
  - Weak Authentication
  - Writing Secure Code
  - Authentication Factors

Enforcing Server-Side Controls

Securing URL Schemes

Using Cryptography Securely

Mitigating Memory Corruption Issues

Defending Against Reverse Engineering

Device Interrogation

Self-Paced Learning | Approximate Time: 60 Minutes
--- | ---
PDH | 0.75
CEU | 0.075

Course Cost:
- Member: $24
- Non-Member: $24
Course Description:

Android applications are affected by issues inherent to the Android platform and by issues common to other types of mainstream software. Developers must be familiar with both types of risks to design and develop applications resilient to common attacks. Building on the Foundations of Android Security course, this course teaches effective techniques for addressing common risks in Android applications.

Learning Objectives:

- Identify security risks relevant to Android applications
- Apply defensive programming techniques to mitigate Android specific risks
- Apply defensive programming techniques to mitigate general software security risks in Android applications

Course Outline:

Addressing Android-Specific Risks
- Avoiding Permission Re-Delegation
- Requesting Permissions Following Least Privilege
- Controlling Intent Recipients
- Prevent Intent Forging
  - Identifying Vulnerable Code
  - Writing Secure Code
- Avoiding Sticky Broadcasts
- Using Internal File Storage Securely
- Using External File Storage Securely
- Preventing Information Leakage

Addressing Common Application Security Risks
- Leveraging Channel Security
- Avoiding Storage of Sensitive Information on the Client
- Performing Strong Authentication
- Preventing Injection Attacks
  - Understanding Dataflow
  - Performing Input Validation
  - White-listing, Black-listing, and Rostering
  - Encoding Output
  - Simulation: SQL Injection
  - Mitigating SQL Injection
- Enforcing Server-Side Controls
- Performing Password-Based Encryption
  - Key Generation
  - Encryption
  - Decryption
- Mitigating Memory Corruption Issues
- Using Cryptography Securely

Device Interrogation
EMBEDDED SYSTEMS & IoT COURSES
EMBEDDED SYSTEM

Course Description:

Embedded software is found in most electronic devices designed today. Embedded software controls our cell phones, microwaves, network routers, automobiles, and industrial controls. Each of these embedded systems is unique and highly customized to the specific application. As a result, embedded systems development is a widely varying field that can take years to master.

This course will cover some of the basic principles of writing software for embedded systems. The course will survey the issues and discusses the various techniques for dealing with them. In particular, the course discusses approaches to the appropriate use of the real-time operating systems upon which much embedded software is based. In addition to explaining what these systems do, this course provides guidance on how you can use them most effectively. The information in this course is not specific to any microprocessor or real-time operating system nor is it oriented towards any particular software design methodology. The principles are the same, regardless of which microprocessor and which real-time operating system and which software design methodology you use. We will concentrate on the principles that you can apply to almost any embedded system project.

The material in this course will provide the necessary information to understand the embedded systems development cycle and the specialized aspects of developing and testing software in this environment. This course will also discuss the design considerations unique to embedded systems. The course will cover the key methods and technologies for each phase of the development process: specification, partition, design, integration, validation, and maintenance and upgrade.

Part I – Module 1-6 (total of 6 modules)
1. Megatrends and driving forces in IT and their linkages to embedded systems
2. Overview of embedded systems including characteristics, attributes, programming languages, processor technologies, and design challenges
3. Overview of real-time systems including common characteristics, interaction with hardware devices, and embedded systems design process
4. Current examples of embedded systems, structure of an embedded program, best practices, and typical code examples
5. Block design of embedded systems, optimal mix of software and hardware required to build a system, guide to selection of processors and peripherals, signal processing, and buffer design
6. DSP fundamentals and architecture and additional material to deepen understanding of embedded system software development

Part II - Modules 7-11 (total of 5 modules)
7. Overview of Round Robin scheduling, Function Queues, brief introduction to Real-time Systems, and coding examples to strengthen understanding
8. Overview of Super Loop Architecture, additional characteristics of Real Time Operating systems (RTOS), and basic elements of Hardware Abstraction Layer (HAL)
9. Overview of task management, reentrancy, and use of semaphores
10. Additional details on scheduling and periodic tasks,
11. Rate monotonic analysis, and provide a Motor Control example

Part III Modules 12-14 (total of 3 modules)
12. Priority Inversion, fundamentals of optimization in embedded systems including: C code, importance of physical architecture, power, memory and cache
13. Impact of compilers in optimization, techniques to use compiler effectively including optimization levels, intrinsics, cross correlation, pragmas, data alignment
14. Using parallel ALU effectively including Partial Summation, Loop Unrolling, Software Pipelining, Multisampling, effective use of pointers, best practices
14.1. Case study to deepen understanding of embedded systems development
MULTI-CORE VIDEO SERIES

Course Description:

Multi-Core Lecture Series consists of 11 one-hour lectures by some of the world’s leading researchers in the field. This series is not a course and it consists of the presentation for those who are in the research field. This is more intended for research information sharing than educational training. Topics that are covered during these lectures are listed below. This series also includes an hour discussion of the lecturers.

Video Presentations:

- Automatic Parallelization by David Padu
- Autoparallelization for GPUs by Wen-Mei Hwu
- Dependences and Dependence Analysis by Utpal Banerjee
- Dynamic Parallelization by Rudolf Eigenmann
- Instruction Level Parallelization by Alexandru Nicolau
- Multigrain Parallelization and Power Reduction by Hironori Kasahara
- The Polyhedral Model by Paul Feautrier
- Vector Computation by David Kuck
- Vectorization by P. Sadayappan
- Vectorization/Parallelization in the IBM Compiler by Yaoqing Gao
- Vectorization/Parallelization in the Intel Compiler by Peng Tu
- Roundtable Discussion by all presenters

Self-Paced Learning | Approximate Time: 12 Hours
---|---
PDH | 12
CEU | 1.2

Series Cost:

- Member: $495
- Non-Member $695

Cost for Each Video:

- Member: $95
- Non-Member $125
Course Description:

Quartos are unique new short, tightly-focused topical courses, usually around thirty minutes, based on current articles in the Computer Society’s twelve peer-reviewed magazines. Ranging across the entire field of computer science and engineering, Quartos are written by the leading researchers and thought leaders, and give learners an opportunity to pick up professional education credit for learning about what’s happening in computer science and engineering today. In addition to an interactive learning experience, the course includes a digital copy of the original article on which it’s based, so learners can build their own reference library.

Courses:

- A Cloud Security Risk-Management Strategy
- Application-Screen Masking A Hybrid Approach
- Big Data Privacy in the Internet of Things
- Cyberhuman Security
- Denial and Deception in Cyber Defense
- Digital Data Grows into Big Data
- Inferring Mobile User Status with Usage Cues
- Internet of Things Making the Hype a Reality
- Internet of Things Perspectives
- Protecting Websites from Attack with Secure Delivery Networks

<table>
<thead>
<tr>
<th>Self-Paced Learning</th>
<th>Approximate Time: 1 Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDH</td>
<td>1.0</td>
</tr>
<tr>
<td>CEU</td>
<td>0.1</td>
</tr>
<tr>
<td>Cost for Each Video:</td>
<td>Member: $19</td>
</tr>
<tr>
<td></td>
<td>Non-Member $29</td>
</tr>
</tbody>
</table>