

**2012**

**COURSE OUTLINE FOR  
COSC 424**

**Secure Software**

**Course Supervisors:**

**Ray Hunt**

(Room 314, CSSE, ext 6347, [ray.hunt@canterbury.ac.nz](mailto:ray.hunt@canterbury.ac.nz))

**Course Lecturer:**

**To be advised**

## **1. OBJECTIVES OF THE COURSE**

The recommended preparation for this course is SENG365 (Web Computing Architectures) but other very useful background material is covered in COSC221 (Introduction to Computer Architecture and Operating Systems), ENCE361 (Embedded Systems) and COSC362 (Data and Network Security).

This course is also designed to be taken in conjunction with COSC407 Wireless and Mobile Network Security, COSC425 Computer and Network Forensics A and COSC430 Information Security and Access Management, COSC429 Cybersecurity. It is one of the papers which can be taken for the Post Graduate Diploma in Computer Security and Forensics as well as for Honours or Masters (Part 1).

This course is concerned with designing and implementing secure operating system and application software, that is, software that is not vulnerable to malicious attacks. This course particularly addresses web server security and the methodologies required to ensure web servers are not vulnerable to security breaches.

This course has relevance to Software Engineers as well as Computer Security specialists. Much of this course will take place in the security and forensics laboratory (Lab 3) where students will build a secure web services platform and subject such a platform to a range of security tests. Thus this course has a strong applied focus.

By the end of the course, students should be familiar with why security is important, what types of vulnerabilities can be present in software, how they can be exploited, and how to go about developing software that is sufficiently secure. The course involves significant practical work as well as written assignments. The course explains some of the most common security issues involved in the development of software, including secure coding practices, secure database access, secure data communications, security of web applications, use of encryption techniques and security testing and evaluation.

Design of secure web service infrastructure including topics such as: principles of least privilege, types of vulnerabilities and how they arise, security requirements definition, system specification, security procedure definition and security management and audit, threat and vulnerability analysis, information leakage, integrity violation, exploitation of vulnerabilities, e.g. buffer overflows, design and implementation of secure web servers and their applications, use of encryption and authentication, automation and testing, Denial of Service (or - how to build a secure web-based infrastructure).

## **2. ASSESSMENT**

This course will be assessed with a laboratory report, an assignment and an examination. The laboratory report will specifically follow a set of secure software testing experiments which will be carried out in Lab 3 at times to be arranged. With respect to the assignment, careful attention needs to be paid to the design, structure, language and grammar of the assignment.

Laboratory report due Friday 17 August 2012 (25% of course)

Assignment due Friday 5 October 2012 (25% of course)

Examination (2 hours) date to be advised (50% of course)

### **3. SYNOPSIS**

#### **Term 3**

- How application security impacts other aspects of security
  - case study of secure web-based infrastructure
    - network security
    - host security
    - database security
- Secure application design
  - threat modeling and vulnerability assessment
  - design principles
- Types of software security vulnerabilities
  - web-based
  - client-based
  - server-based
- Programming a secure application
  - authentication
  - authorisation
  - session management
  - data validation
  - interpreter injection
  - canoncalization
  - error handling auditing and logging

#### **Term 4**

- Programming a secure application (contd.)
  - buffer overflows
  - administrative interfaces
  - cryptography
  - configuration and maintenance
  - web services
  - ajax and other rich client technologies
  - handling e-commerce payments
- Testing applications for security vulnerabilities
  - Software quality assurance
  - source code review
  - automated testing
  - manual testing