1. **Department, number and title of course:** Computer Science: CS 161, Computer Security

2. **Catalog Description:** (4 units) This course will cover the most important features of computer security, including topics such as communication security, operating systems security, database security, programming language security, application level security, privacy, and digital rights management. The course is designed for junior- and senior-level students, primarily those specializing in computer science.

3. **Prerequisites:** CS 61C, Machine Structures; Math 55 or CS 70, Discrete Mathematics

4. **Textbooks and/or other required material:**
   - Optional:

5. **Course objectives:** The focus of this course is on modern language- and cryptography-based approaches to security, instead of older verification-based approaches. Course content is closely coordinated with a substantial project. After completing this course, students will be able to analyze, design, and build secure systems of moderate complexity.

6. **Topics covered:**
   - Introduction to Computer Security: Basic concepts, threat models, common security goals.
   - Cryptography and cryptographic protocols, including encryption, authentication, message authentication codes, hash functions, one-way functions, public-key cryptography, secure channels, zero knowledge in practice, cryptographic protocols and their integration into distributed systems, and other applications.
   - Operating system security, including memory protection, access control, authorization, authenticating users, enforcement of security, security evaluation, trusted devices, digital rights management.
   - Network security, including firewalls, intrusion detection systems, viruses and worms, web security. Case studies, including DNS, IPSec.
   - Software security, including secure software engineering, defensive programming, buffer overruns and other implementation flaws.
   - Language-based security, including analysis of code for security errors, safe languages, and sandboxing techniques.
   - Advanced topics and case studies, for example: Privacy, mobile code, digital rights management and copy protection, trusted devices, denial of service and availability, network based attacks, security and the law, electronic voting, quantum cryptography, penetration analysis, ethics, full disclosure.

7. **Class/laboratory schedule:** Three hours lecture and one hour of discussion per week.

8. **Contribution of course meeting the professional component:** This course contributes to engineering topics. Students learn both principles and specific tools. Students complete group projects that provide experience on how to design secure systems and to analyze the security of existing systems. These projects also require students to learn specific tools and libraries that are useful to security engineers. It is approximately 50% science and 50% design. It provides a capstone design experience.

9. **Relationship of course to program objectives:** This course requires students to apply their fundamental knowledge of mathematics, science and engineering to analyze and
solve problems. Students identify, formulate, and solve challenging engineering problems. They apply modern skills, tools and techniques, and they need to communicate their ideas effectively with other students and the instructor for the class. Students work in teams and learn collaborative techniques. They learn of emerging applications.

10. **Prepared by**: Professor David Wagner; May 12, 2006.