CS342 Course Information

[9/08/06] Modified Lyn's office hours

1 Course/Instructor Information

Lectures:	SCI E111, Tue/Fri 1:30–2:40pm
Web Site:	`http://cs.wellesley.edu/~cs342`
First Class:	`CS342-F06`
Professors:	

Name:	Daniel Bilar (call me "44616e" or "Dan", whichever is easier)	Franklyn Turbak (call me "Lyn")
Office:	SCI E108	SCI E126
Phone:	x3093	x3049
E-mail:	`dbilar@wellesley.edu`	`fturbak@wellesley.edu`
Office Hours:	T 3-5pm, R 3-5pm	M 11am-1pm, W 4-6pm, F 4-6pm

Appointments can be made for times outside our usual office hours. We will sometimes need to cancel or shift office hours to attend a meeting or a talk. We will post a message to `CS342-F06` to announce a change in office hours.

You should plan on reading the CS342-F06 conference on a regular basis for announcements, questions about assignments, discussions of course material, etc. It is strongly recommended that you add it to your FirstClass desktop.

2 Course Overview

This course is an introduction to computer security, which traditionally covers the protection of confidentiality, integrity, and availability of computational resources. Topics covered in the course are private-key and public-key cryptography, security protocols, authentication, access control, information flow, network security, intrusion detection, computer forensics, viruses and worms, buffer overflows and other software vulnerabilities, Web security, mobile code security, electronic voting, privacy issues, legal issues, and ethical issues. Assignments include paper-and-pencil exercises as well as hands-on activities with security exploits and tools in a Linux environment. Participants will independently research, present, and lead discussions on security-related topics.

3 Big Ideas

New vulnerabilities and exploits are reported every day. Unless you are a full-time professional specializing in computer security, it is hard to keep track of all of them. So the course will focus on the following **big ideas**, principles that remain the same even though the day-to-day details may differ:

> *Vulnerabilities Stem from Violated Assumptions*: Insecurities almost always arise from faulty, unchecked assumptions made in system design. For example:

> You think you are updating personal information for your online banking account but are actually giving your bank information to a phishing site.
>
> Alice thinks she is communicating privately with Bob, but is actually communicating with Carol (or Carol is listening to the communication).
>
> A firewall program may fail when inundated with an unexpected level of network traffic.
>
> In a buffer overflow attacks, what is assumed to be a return address to a procedure is actually the address of hostile code inserted by a program input that exceeds the maximum assumed length.

Security is a System: Although technological tools (such as cryptography) are important for security, using these tools does not guarantee security. Any such tools are embedded in a larger system that involves potential insecurities due to people, improperly used technology, and the way the pieces of the system are organized. There are also fundamental trade-offs between security and convenience/usability that must be balanced.

Understanding Security Requires Wearing Different Hats Evaluating the security of a system requires viewing it from many different perspectives, such as those of the designers the users, and potential attackers. Also, anthropocentric thinking can be a liability when it comes to understanding the scales of resources involved in attacks (e.g., incredibly short processing times, incredibly large bandwidths).

4 Goals

(Major) Understand and critically evaluate computer security systems from different perspectives.

(Major) Gain basic knowledge of history, key technology, and critical issues relevant to computer security systems.

(Major) Exhibit adeptness and initiative in dealing with unknowns.

(Minor) Become familiar with practical security tools and techniques in hands-on context.

(Minor) Show creative and interesting synthesis in final project.

5 Philosophy

We are a community of learning in which exploration/curiosity/experimentation are encouraged.

The instructors are not gurus who know all but guides who are also learning in the community.

There is no such thing as a dumb question! (After all, questions expose hidden assumptions.)

6 Expectations

You should attend every class.

You should do the assigned readings before discussion classes. (We will announce which classes will have a discussion component).

You should be willing to find resources and learn technology/tools largely on your own or with classmates.

Ask questions (of instructors/classmates) when you get stuck.

7 Learning Strategies

Be playful. Experiment!

Use online security resources. (See CS342 web page for some starting points.)

Collaborative learning is encouraged.

Start assignments and the final project early!

8 Reading Materials

There is no textbook for this course. However, we will often post reading materials online or ask that you read chapters from books in the *Security Library*: the bookshelf in the back of the Networks/Security Lab library (SCI 121B). Because we will all be sharing the books from the Security Library, we require you to use the clipboard magnetically attached to the bookshelf to sign out any book you take from the room. This way, others can track down the book if they need it.

9 Course Work

There will be six problem sets, most of which will have both paper-and-pencil and hands-on (system administration and programming) components. You will typically have a week and a half or two weeks to work on a problem set. See the course schedule for the tentative schedule of problem sets. *The hands-on components of your assignments may take significant time; start them early!*

Students will work in pairs to administer their own Linux machine in the Networks/Security lab (SCI 121B). Problem sets will typically include a problem involving defending the security of this machine or attacking another machine.

The course culminates in a final project on a computer security topic that you choose. The project has both a written component (a paper whose final version is due Thu. Dec. 21) and an presentation components (a 30-minute talk to the class during one of the final four class sessions). See the course schedule for the schedule of project milestones. Start thinking of project ideas early and bounce them off the instructors.

There are no exams in this course.

Your course work will contribute to your grade as follows:

Problem sets	60%
Final Project	30%
Class participation	10%

10 Collaboration Policy

We believe that collaboration fosters a healthy and enjoyable educational environment. For this reason, the course is designed to encourage you to work with others on many assignments. However, there are some things you must do on your own.

There are two kinds of problems on problem sets:

1. *Individual problems*: An individual problem **must** be completed by each individual student without collaborating with anyone else. It is effectively a "take-home quiz".

2. *Group problems*: In a group problem, you may work with a partner in a two-person team. The two team members can (in fact, must; see below) work closely together on the assignment and turn in a single hard- and soft-copy of the assignment for the team. The grade received on such a submission will be given to both team members.

   All problems that involve system administration are group problems. In these problems you **must** work with your administration buddy.

   There are also some problems that do not involve system administration in which you *may* (but are not required) to work with a partner. In this case, the partner need not be your administration buddy. Based on past experience, we think that working with a partner can often significantly decrease the amount of time you spend on an assignment, because you are more likely to avoid silly errors and blind alleys. On the other hand, certain individual may take more time on an assignment than they would alone. In this case there are still benefits to working with a partner, but they may be outweighed by the time cost.

   All work on group problems must be a true collaboration in which each member of the team will carry her own weight. It is *not* acceptable for two team members to split the group problems of an assignment between them and work on them independently. Instead, the two team members must actively work together on all parts of the assignment. In particular, almost all group programming assignments should be done with the two team members working at the same computer. It is strongly recommended that both team members share the responsibility of "driving" (i.e., typing at the keyboard), swapping every so often.

   The fact that team members have to work closely together means that you need to carefully consider a potential partner's schedule before forming a team. You cannot be a team if you cannot find large chunks of time to spend together!

Unless otherwise instructed, teams are allowed to discuss the group problems (but **never** the individual problems) on problem sets with other teams and exchange ideas about how to solve them. However, there is a thin line between collaboration and plagiarizing the work of others. Therefore, we require that each (one-person or two-person) team must compose its own solution to each assignment. In particular, while you may discuss strategies for approaching the programming assignments with other teams and may receive debugging help from them, **each team is required to write all of its own code**. It is **unacceptable** (1) to write a program with another team and turn in two copies of the same program or (2) to copy code written by other teams. Such incidents will be interpreted as violations of the Honor Code.

In keeping with the standards of the scientific community, you must give credit where credit is due. If you make use of an idea that was developed by (or jointly with) others, please reference them appropriately in your work. E.g., if person X gets a key idea for solving a problem from person Y, person X's solution should begin with a note that says "I worked with Y on this problem" and should say "The main idea (due to Y) is ..." in the appropriate places. It is unacceptable for students to work together but not to acknowledge each other in their write-ups.

When working on homework problems, unless you are otherwise instructed, you may consult class materials, reference books, and online resources, for hints, techniques, and even solutions. However, you **must** appropriately cite any sources that contribute to your solution. Using the work of others without appropriate citation is considered a violation of the Honor Code.

11 Ethics

Note: this section is adapted from the MIT 6.857 Course Information handout.

This is a course on Computer Security. Although the course is primarily concerned with techniques that are designed to increase the security of networks and computer systems, a proper understanding of those systems requires that you be versed in their vulnerabilities and failings as well.

Nevertheless, unless you have explicit written authorization from the owner and operators of a computer network or system, you should never attempt to penetrate that system or adversely affect that system's operation. Such actions are a violation of Wellesley policy and, in some cases, violations of State and Federal law. Likewise, you should refrain from writing computer viruses, worms, self-reproducing code, or other kinds of potentially damaging software for this course unless you have explicit, written approval for the specific type of software that you wish to create. These kinds of programs are notoriously difficult to control and their release (intentional or otherwise) can result in substantial civil and criminal penalties.

We recommend that you read and review the following documents on the ethical use of information technology:

- The Responsible Use of Information and Technology Resources at Wellesley College (`http://www.wellesley.edu/acceptuse.html`).
- The ACM Code of Ethics and Professional Conduct (`http://www.acm.org/constitution/code.html`).

11.1 Late Homework Policy

All problem sets will be due at the advertised time (typically 6pm on a Tuesday or Friday).

We realize that it is not always possible to turn in problem sets on time. On the other hand, turning in one problem set late can make it more difficult to turn in the next problem set on time. To address this issue, we have decided to use *Lateness Coupons* this semester. A problem set can be turned in $24 \cdot n$ hours late if it is accompanied by n Lateness Coupons. If you work with a partner, each of you needs to attach one Lateness Coupon per person per day late.

At the end of this handout, you will find six Lateness Coupons (one for each assignments) that you can use throughout the term. Use them wisely: you only get ten, and they are not copyable or transferable between students. You also cannot use lateness coupons on any part of your final project.

In extenuating circumstances (e.g.,, sickness, personal crisis, family problems), you may request an extension without penalty. Such extensions are more likely to be granted if they are made before the due date.

11.2 Problem Set Header Sheets

We would like to get a sense for how much time it takes you to do your CS251 problem sets. We use this information to design problem sets later in the semester, as well as for future semesters.

Please keep track of the time you spend on each problem of your problem sets, and include this information on the problem set header sheets that we will provide at the end of each problem set. (Two time columns will be provided for the case of students working together on an assignment.) Turn in this header sheet as the first page of your hardcopy submission. Assignments will typically have both group problems and individual problems; a separate header sheet will be provided for group problems and individual problems.

12 Students With Special Needs

Students with disabilities (including "hidden" ones, like learning disabilities) who need disability-related accommodations are encouraged to work with Barbara Boger, Director of Programs, the Learning and Teaching Center (for learning or attention disabilities), or Jim Wice, Director of Disability Services (for physical disabilities), to arrange accommodations. Their offices are in Clapp Library.

CS342 Lateness Coupon #1
CS342 Lateness Coupon #2
CS342 Lateness Coupon #3
CS342 Lateness Coupon #4
CS342 Lateness Coupon #5
CS342 Lateness Coupon #6