Course Information

Instructor: Prof. Kevin Fu
Room CS230, 545-4006, kevinfu@cs.umass.edu
Office Hours Mondays: 10:20–11:20AM or by appointment
(Please CC calendar requests to quinns@cs.umass.edu)

Staff email list: cs660-staff@cs.umass.edu
Student email list: cmpsci-660-01-fal12@courses.umass.edu
Web page: http://spqr.cs.umass.edu/courses/cs660fa12/

1 Overview

This graduate-level course covers key concepts in information assurance (IA) via cutting-edge and seminal research papers. The course’s two goals are (1) to prepare students to conduct successful, publishable research in security and privacy and (2) to teach students how to build stronger systems by thinking about attacks and defenses. Topics include threat modeling, risk assessment, famous missteps, the economics of security, and web security among others. Students will be evaluated based on several homeworks, five in-class quizzes, a small-team project, and class participation.

Intended audience. This 3-credit course is intended for graduate students who plan to pursue a research career in computer security. This is a graduate-level course. Homework assignments will pose open-ended questions, and you will be asked to produce somewhat free-form responses evaluating the importance of published research.

If you are looking to learn about the fundamentals and cultural norms of the vibrant security community for publishable research, this course is for you! If you are seeking to primarily learn operational skills (e.g., how to configure a firewall or SSH server), you probably will not enjoy this course.

Prerequisites. CS460 (Computer and Network Security) or CS466 (Applied Cryptography).

Time and location. Lectures are held in Room 142 in the Computer Science Building on Mondays and Wednesdays from 9:05AM to 10:20AM. A schedule of topics is posted on the Web site.

Contact. Be sure to check your UMass email address regularly. If you check another email address, set up your UMass account to forward your mail appropriately.

2 Textbook and reading

There is no textbook for this course. We will assign regular readings from research literature on computer security. However, the following books may help the budding researcher: Security Engineering by Ross Anderson, Principles of Computer System Design by Saltzer and Kaashoek.
3 Grades and methods of evaluation

Grading is weighted as follows:

- Five quizzes 35%
- Homework/labs 30%
- Final project 25%
- Class participation 10%

Passing the class is not possible without completing the final project, regardless of your other grades.

3.1 Quizzes

There will be five quizzes during the semester. Quizzes will take place at the beginning of a lecture, roughly at the conclusion of a topic area; i.e., the class meeting following the conclusion of a topic. The intent of each quiz is to test your understanding of the immediately covered material from the readings and lectures. A quiz is not intended to be a comprehensive exam, but you may need to understand past material to answer quiz questions that build on past material.

Quizzes are closed book. There are no make-up quizzes, but we will instead drop your lowest quiz score. Plan your strategy accordingly and budget your time for unexpected absences.

UMass policy is that students are required to notify instructors at least one week in advance that they will miss class in order to observe a religious holiday; in this case we will offer a makeup quiz.

3.2 Homework

Individual homework will consist of both technical essays responding to research papers, and hands-on homework assignments related to technical problem solving and measurement. Due dates will appear on the Web site.

Essays are one-page responses to a technical paper. The essay should follow strategies for effective technical writing. The essays will be graded on both the quality of writing as well as the effectiveness of the technical argument. During lecture, the instructor may add additional twists such as focusing on a particular technical element of the paper. Please use the following guidelines based on the work of Alex J. Halderman. In each response to a required paper:

1. In the first paragraph:
   (a) State the problem that the paper tries to solve; and
   (b) Summarize the main contributions.

2. In one or more additional paragraphs:
   (a) Evaluate the paper’s strengths and weaknesses;
   (b) Discuss something you would have done differently if you wrote the paper; and
   (c) Suggest at least two interesting open problems on related topics.
3. List any areas you had trouble understanding. I’ll try to explain them in class.

Your most important task is to demonstrate that you’ve read the paper and thought carefully about the topic. Your responses should be no longer than about 400 words per paper. Paper responses are due before the start of lecture. At 9:05 AM, a homework assignment will be considered late. Email your PDF responses (no Word docs allowed) to cs660-staff@cs.umass.edu. Please use the subject line [cs660 response].

Absolutely no collaboration is allowed on the essays; see the plagiarism policy later in this document to avoid failing the course.

Here is an example of an exemplary homework response donated by Andrew Hall. He is responding to the “DieHarder: Securing the Heap” paper by Novark and Berger at ACM CCS 2010.

The paper describes a new memory allocator, DieHarder, designed to mitigate the threat of heap-based overflows as attack vectors against memory bugs in code. First, it offers a broad overview of the mechanism of a heap-overflow exploit, and the ways in which such exploits rely on the design of the memory allocator used. In doing so, the authors also provide a security-focused summary of the designs of the major memory allocators in use today. They continue by noting the major design changes implemented in DieHarder versus DieHard, in particular the use of sparse page layout to greatly increase the probability of “guard pages” existing between allocated objects. In addition, DieHarder makes use of OpenBSD’s “destroy-on-free” mechanism to further complicate exploitation for attackers restricted in the number of simultaneous live objects they control.

The majority of the paper was spent reviewing the basic mechanisms of heap-based vulnerabilities and the allocation techniques that enable them, which I found extremely helpful in comprehending the presented results. Additionally, the paper presented benchmarks illustrating the relative performance of the various mentioned allocators, establishing the essential practicality of the system (as opposed to the more extreme Archipelago). Ultimately, I believe the authors presented a convincing case that the present number of heap-based attacks “in the wild” and the plethora of “stop-gap” measures employed to counter them are symptomatic of fundamental design flaws (from a security perspective) in the major memory allocators in use today, and that DieHarder provides a practical example of a memory allocator designed with security in mind from the beginning.

3.3 Final project

A major part of this course is a team project. We will assign you team partners. Each team will have 3–4 members. One goal of this course is to expose you to the joys and difficulties working in teams, which is the reality of 96.4% of all academic and industrial projects. You will be responsible for organizing team meetings around your many schedule constraints. Effective teamwork is essential. We will spend class time discussing how to be a good team partner.

There are two options for project topics:

1. Choose a security-related paper from the last year and reproduce its results (e.g., run your own experiment or run your own simulation). Acceptable venues from which to draw papers include the USENIX Security, IEEE Symposium on Security & Privacy (aka Oakland),
ACM CCS, or NDSS. Occasionally there are also security papers at OSDI, NSDI, SOSP, and SIGCOMM. Check with the instructor if you have a passion for a different venue and seek permission to use a paper for your project.

2. Propose a fresh project that thoroughly explores a concept in security or privacy. We will devote part of a class period to brainstorming project ideas; you may think of your own or select one from the list we come up with as a class. This choice is much more open ended. The advantage is that there is more room for creativity, but the danger is that the problem is very open ended and could result in disaster if the project does not work out in the end.

No two teams may choose the same project. Don’t worry—there are enough to go around.

Project milestones will be spaced throughout the semester to make sure everyone keeps up. A rough schedule follows:

**First weeks of class:** Class-wide brainstorming on potential topics. Each team emits a project proposal.

**Within two weeks of the proposal:** Each team emits a detailed project plan.

**Body of semester:** Teams work independently on projects, emitting one mid-semester status report.

**Several weeks before the end of class:** Each team reports a summary of results to the instructor.

**After that:** Each team turns in a final writeup analyzing the results and putting the project into context.

Feel free to solicit faculty members to informally co-supervise your class project. The best possible outcome of a class project is a publishable research artifact. You may not, however, receive credit for the same project twice, e.g. by undertaking an independent study for the same outcome.

A list of project ideas will appear on the course Web site. There are four components to your project grade: a project proposal, a midterm status report, a final project report, and a project presentation. The due dates for each of these milestones appear on the course Web page. You cannot pass the project unless each is completed.

Communicate with your teammates! Lack of communication could result in a dysfunctional team that risks failing the class. If you have tried repeatedly to communicate with an unresponsive team member, contact the course staff before the problem becomes unmanageable.

**Project proposal due September 26 (20%).** Your proposal should explicitly state the problem your project will address, your project’s goal and motivation, related work, the methodology and plan for your project, and the resources needed to carry out your project. Be sure to structure your plan as a set of incremental milestones and include a schedule for meeting them. Part of the grade will involve peer review; we will pseudorandomly assign another team to provide a constructive critique of your proposal.
Status report due November 7 (30%). Your status report should contain enough data and analysis to show that your project is on the right track. You should append a copy of your original proposal with instructor comments, along with any surprising results or changes in direction, schedule, etc. You should also have a refined version of the problem statement and goals, as well as a more developed related work section. Part of the grade will involve peer review; a different team will provide a constructive critique of your status report.

Final report and Presentation due December 5 (50%). A final report describes your research problem, contributions, results, and analysis. You will present your research problem, analysis, and results in a brief presentation. The presentation may include a system demo if appropriate. The final report must include a paragraph explaining, for each team member, their contributions and duties in the project. Part of the grade will involve peer review; students will provide a constructive critique of presentations.

Peer Rating of Team Members (weighting factor) To encourage team members to contribute to the success of the project, individual grades will take into account peer ratings from each team member. Ratings are excellent, very good, satisfactory, ordinary, marginal, deficient, unsatisfactory, superficial, and no show. The course staff will use the peer feedback as a weighting factor for individual grades for the team project. We provide the following examples of weighting factors. A student receiving all “excellent” or “very good” ratings would receive a 100% weighting factor for the team grade. A student receiving all “ordinary” ratings would receive a 75%. A student receiving all “deficient” ratings would receive a 50%. Universal ratings of superficial or no show would result in a zero for the team project.

Your report should follow the structure of a research paper. Your presentation should follow the structure of a research talk. We will discuss how this is done in class.

3.4 Class participation
Students can participate in class in several ways. At the beginning of class, students will have the opportunity to present a 5-minute talk on a research-worthy, intellectually-stimulating topic in computer security that is thematically related to the topic of the day (but not the assigned reading). The material could draw upon one of the optional papers, or a paper that you find. You may sign up for a time slot. Students can also engage in discussion during class and on the class email list. Quality rather than quantity counts most in this subjective evaluation. One can also gain class participation credit by signing up to “shepherd” other teams’ projects. That is, you can provide feedback on write-ups.

4 Policies
4.1 Lateness
Each student is granted one “penalty free” late pass for turning in a homework assignment. You need not provide any excuse. A free late means you may turn in the homework by 8:00 AM on the day of the next class without penalty. We will strictly enforce the deadline; we do not want to lingering on old assignments that delay new assignments. Homeworks will be accepted only as
emailed PDF files. The turn in date is when we receive the message, not when you send it. Any late homework beyond your one freebie will result in a zero grade. Late freebies may NOT be used for any of the term project assignments. A late final project assignment (i.e., the proposal, status report, or final report) will have a 20% grade reduction for each late weekday (7:59AM).

4.2 Incomplete grades will not be...

Because this is Prof. Fu’s last semester teaching at UMass Amherst, there is no opportunity to apply for an incomplete grade (i.e., finishing the course later). If a project or any other element of the course is not completed by the last day of class, the student will be assigned a grade based on the materials submitted. There is no opportunity for an extension from the instructor as I will no longer have a teaching position at UMass Amherst.

4.3 Ethics, Law, and University Policies

These guidelines on ethics come from Alex J. Halderman, a professor noted for his research in security analysis and electronic voting.

To defend a system you need to be able to think like an attacker, and that includes understanding techniques that can be used to compromise security. However, using those techniques in the real world may violate the law and the university’s computing practices, or may be unethical. You must respect the privacy and property rights of others at all times, or else you will fail the course. Under some circumstances, even probing for weaknesses may result in severe penalties, up to and including civil fines, expulsion, and jail time.

Before engaging in any security analysis, carefully read the Computer Fraud and Abuse Act (CFAA),\(^1\) a federal statute that broadly criminalizes computer intrusions. This is just one of several laws that govern hacking. Understand what the law prohibits—you don’t want to end up like this guy.\(^2\) The EFF provides helpful advice on vulnerability reporting\(^3\) and other legal matters\(^4\). If in doubt, I can refer you to an attorney.

4.4 Collaboration and plagiarism

You may discuss material with others, but your writing must be your own. When in doubt, contact the instructors about whether a potential action would be considered plagiarism. The University maintains a detailed policy on cheating.\(^5\)

When discussing problems with others, excluding projects, do not show any of your written solutions to others, including code. Do not take notes about the solution other than to jot down publicly available references. Use only verbal communication.

Using someone else’s code or API is forbidden. You may use publicly available code (libraries and open source material) if code was published before we assigned the work. If you find code that trivially solves some problem we have assigned, we expect you’ll tell us where so that we learn the homework assignment is moot.

---

\(^1\)http://www.law.cornell.edu/uscode/18/1030.html  
\(^2\)http://en.wikipedia.org/wiki/Sarah_Palin_email_hack  
\(^3\)https://www.eff.org/issues/coders/vulnerability-reporting-faq  
\(^4\)https://www.eff.org/pages/grey-hat-guide  
\(^5\)http://www.umass.edu/dean_students/codeofconduct/acadhonesty/#B
If you do discuss material with anyone besides the instructors, acknowledge your collaborators in each write-up. If you obtain a key insight with help (e.g., through library work or a friend), acknowledge your source, briefly state the insight, and write up the solution on your own. In most of your write-ups, we expect to see citations.

We cannot emphasize enough that you MUST cite all your sources properly. You must remove any possibility of someone else's work from being misconstrued as yours. We consider the facilitation of plagiarism (giving your work to someone else) as plagiarism as well. If we detect two homework assignments that share text, both persons will be disciplined.

Never misrepresent someone else's work as your own. It must be absolutely clear what material is your original work. Plagiarism and other anti-intellectual behavior will be dealt with severely. Investigating plagiarism is a pleasant experience for neither instructor or student. Please help us by avoiding any questionable behavior.

Cheating is usually the result of other problems in school. Please come see us anytime if you are unable to keep up with the work for any reason and we will work something out. We want to see you succeed and will do everything we can to help you out!

Any activity considered by the instructor to be plagiarism or anti-intellectual activity will result in a zero for the assignment, your likely failure in the course, and a letter submitted to the University.