Teaching Spam and Spyware at the University of C@1g4ry

John Aycock
Department of Computer Science
University of Calgary
2500 University Drive N.W.
Calgary, Alberta, Canada T2N 1N4
aycock@cpsc.ucalgary.ca

ABSTRACT

The University of Calgary is attacking the problems of spam and spyware from the angle of education. "Spam and Spyware" is a computer science course offered at both the undergraduate and graduate level, that objectively examines the legal, ethical, and technical aspects of spam and spyware and their countermeasures. We believe that this is the only course of its kind in the world. Furthermore, students are given hands-on experience in a secure laboratory, developing software for spamming, spyware, and defenses against them. This paper documents our course and its rationale.

1. INTRODUCTION

The University of Calgary's Department of Computer Science offered a new course in the fall of 2005 which attracted some notice – and controversy – in the computing community [10] and the media (e.g., [3, 4, 9]): Spam and Spyware. To the best of our knowledge, this course was the only one of its kind in the world at the time we planned it in 2004; recent searches indicate that this is still the case.

Spam and Spyware is a 13-week computer science course, with 150 minutes of lecture time per week, offered at both the 4th-year (senior) undergraduate level and the graduate level. A hands-on approach is taken, and students do assignments in a secure laboratory where they create software for spamming, anti-spam, spyware, and anti-spyware. Law and ethics are a major, integral part of the course, and all material is treated in an objective way.

In the remainder of this paper, we discuss the rationale for the course, course admission, what we taught, the secure laboratory and programming assignments, and our future plans for the course.

2. WHY? AN APOLOGIA

The prospect of teaching students about spam and spyware makes some people uneasy, arguing that this is a risky proposition. We would qualify that: it is risky unless done correctly. Potential spammers need not spend four years in university, hoping to enroll in a spam course, when they can spend four minutes with a web search engine and find hordes of bulk mailing software. A similar argument applies for spyware. On the other hand, finding detailed, organized information about the entire area of spam and spyware, including defensive techniques, is much more difficult. By providing this knowledge to students, we are giving them a solid base upon which to construct better defenses. Universities have a responsibility to society to educate people, so that society can benefit from an educated populus. Given that spam and spyware are frequently touted as major problems for our computer-dependent society, universities should be lining up to teach students about spam and spyware.

Another issue is our choice of teaching methodology. *Spam and Spyware* is a hands-on course, where students write spyware and spamming software, along with their countermeasures, in a secure laboratory. That students learn about both offensive and defensive techniques is in keeping with the balanced, objective view the course uses in its coverage of the area. The hands-on, learning-by-doing aspect is a technique used successfully for many other non-security courses; as students should be given the best education possible in computer security, it would be nonsensical to avoid effective teaching methods so long as the material can be taught safely. Research has shown that students enjoy a significant increase in their knowledge when hands-on work is used to complement traditional lectures [7], and in general such "active learning" is best for students [5].

Finally, why combine spam and spyware into one course? In our case, part of the reason was historical. We have a course on computer viruses and malicious software [2], which has lots of material already; neither spam nor spyware would fit into the existing course. Ultimately, there is an overlap between spam and spyware, because at a high level they are all about information. The information may be stolen (spyware), the information may be volunteered (selling via spam), perhaps surrendered under false pretenses (phishing spam), but the common denominator is information.

3. COURSE ADMISSION

One concern was whether or not anyone could "sit in" on the course. This was not the case. Undergraduate students had to meet certain criteria in order to be admitted to *Spam and Spyware*, in addition to traditional course prerequisites:

- They needed a grade point average of 3.0/4.0 or higher.
- Students had to be Computer Science students, at the 4th-year level or above in their program of study.
- Students submitted a one-page essay, explaining why they wanted to take the course and what their learning expectations were. The essays were evaluated by a committee.

Graduate students were subject to similar criteria, although a personal interview and consultation with the student's graduate supervisor were substituted for the admission essay.

No formal (or informal) auditing of the course was permitted, and the identity of students attending the lectures was verified by the instructor. Due to the overhead of this requirement, and lab space restrictions, the course enrollment had a maximum of 16 students.

CEAS 2006 - Third Conference on Email and Anti-Spam, July 27-28, 2006, Mountain View, California USA

4. ANNOTATED COURSE SYLLABUS

The material we taught students is summarized in this section. The course syllabus is presented below in its original form, annotated with explanatory comments that are denoted using a grey background. The course material was not found in a single place; we pieced it together from almost 200 reference sources.[1] None of these topics were given a superficial treatment, and detailed examples and explanations were given for each. (Spammers and phishers were especially cooperative in supplying timely examples.) The approximate amount of lecture time spent on each high-level topic is shown as a percentage.

- Introduction *(3%)*
  - Laboratory protocol

    The laboratory protocol governed behavior in the laboratory, and was an integral part of our secure environment. The direct result of failing to abide by the protocol was a failing grade in the course.

    The laboratory protocol applied to not only the students, but to the course instructor and the technical staff maintaining the laboratory too. This is consistent with our philosophy of treating the laboratory like a biohazard area – in fact, the laboratory protocol was initially based on biohazard protocols [6].
  - Legal agreement

    Students had to sign a legal agreement in order to take *Spam and Spyware*; they were notified of this requirement in advance, and students were given sufficient time at the beginning of the course to review the agreement with their own legal counsel if they so chose.

    The legal agreement bolstered the laboratory protocol by imposing a contractual legal obligation on the students to abide by the protocol. It also governed the usage of the course material, specified liability and indemnity, and listed penalties (academic and legal) for violation of the agreement.
  - Professionalism

    Potentially offensive material was encountered periodically in *Spam and Spyware*, like racy images on web sites and sexually-explicit language in spam corpora; this was the cost of using real examples. We set the ground rules early on, discussing professionalism and the avoidance of sexual harassment so that students were forewarned and could deal with this material appropriately.
- Definitions of spam and spyware *(3%)*

  Working definitions of what constitutes spam and spyware were necessary to begin with. Students created these definitions themselves, guided by the instructor, to see firsthand how difficult a task it is. In the end, several "official" definitions were supplied that reflected the diversity of opinion in the field.
- Ethics *(8%)*
  - General ethical theories
  - Recognizing ethical problems
  - Ethical decision-making processes
  - Sample ethical problems
  - Professional codes of ethics and conduct
    - ACM
    - IEEE
    - Canadian Marketing Association

  We made the conservative assumption that students have had little or no training in ethics. Starting at general ethical theories, we progressed into more specialized codes of ethics and conduct. Heuristics were given for recognizing when ethical problems exist, along with ways to analyze ethical problems; example problems were spam- and spyware-related. Students also completed a written ethics assignment.
- Spam and spyware law *(11%)*
  - Canada
  - Australia
  - United States

  The law in this area changes rapidly, so we examined both existing legislation and the leading legislative contenders that are likely to become law. Any prosecutions under existing laws were noted, and a lawyer was brought in as a guest speaker who provided additional information.

  Law and ethics were deliberately presented early in the course, before any programming assignments. This was part of creating a secure laboratory environment, by drawing students' attention to the potential ethical and legal ramifications of their actions.
- Spyware *(23%)*
  - Spyware and adware history
  - Anti-virus and anti-spyware vendors' classification criteria

    Of course, there is no universal agreement on how to classify spyware – especially when the opinion of an alleged spyware producer is considered! We also looked at anti-virus/anti-spyware vendor appeal processes, and the potential legal ramifications of (mis)classification.
  - Why spyware exists

    What are the motivations behind spyware? This was a treatise on the value of information and nefarious uses for it: marketing data, affiliate programs, identity theft, extortion, and espionage.
  - How does spyware get on a machine?
    - Drive-by downloads

      We took a broad view of drive-by downloads which included both exploiting technical vulnerabilities (e.g., buffer overflows in a web browser) and embedding executable objects into web pages. This naturally led into code signing and its attendant problems.
    - Voluntary installation, bundling, and EULAs
  - Spyware capabilities and countermeasures
    - Keyloggers

      Many forms of keylogger were considered: hardware keyloggers, in-kernel keyloggers, and user-space hooking. The implementation of hooking in Mac OS X and Windows was shown (students learned about X Windows hooking implementation in a programming assignment).

[1] Zdziarski's *Ending Spam* [12] may be useful for the spam part of the course, but it was unfortunately published too late for adoption as a textbook.

    - General keylogging defenses

      For example, virtual keyboards, one-time passwords, and two-factor authentication. Attacks against the defenses themselves were also presented.
    - Startup hooks, in-kernel interception, and browser helper objects

      Details were given for both Unix and Windows.
    - Defenses against startup hooks, including automatic identification of bundled software
    - Avoiding uninstallation, self-monitoring techniques, and defenses
    - Sending out information, defenses via hostname lookup and egress filtering, and the more general problem of covert channels
    - Hiding and forms of obfuscation
    - Static and dynamic anti-spyware methods
    - Advanced hiding via rootkits, and rootkit detection

- Phishing *(15%)*
  - Definition and history
  - Social engineering
  - Specialized forms of phishing

    This included discussion of "context-aware" attacks, attacks on one-time password systems, spear phishing, personalized attacks, and aggregation attacks.
  - URL tricks

    A variety of tricks involving URLs were presented: the outright hiding of URLs, URL encodings, and URL obfuscation. The latter category included typosquatting, homograph attacks, and login URLs.
  - Session fixation and preset session attacks, browser proxy settings, and cross-site scripting
  - Pharming methods
  - Infrastructure for phishing
  - Anti-phishing techniques

    A wide spectrum of anti-phishing methods were examined, starting with user education and safe computing policies. Client-side, automatic assessment of web pages covered standalone algorithms up to methods requiring aggregated data and scale. Finally, the monitoring of outgoing traffic for sensitive information was studied. The use of anti-spam techniques against phishing was mentioned at this point, but a detailed discussion was deferred to the anti-spam section.
  - Countermeasures to pharming and specific attacks, like cross-site scripting

- Fraud *(4%)*
  - Advance fee fraud, a.k.a. 419 scams
  - Sweepstakes scams
  - Recovery scams
  - 900 and 809 scams
  - Overpayment scams
  - Money-laundering

  Various forms of fraud were presented, so that students are able to identify classic scams, and have a general notion of the type of scheme they should be wary of.

- Email *(7%)*
  - Mail system architecture
  - Mail routing and the DNS
  - SMTP transactions
  - Mail envelopes and headers

  A detailed knowledge of how mail is sent and received on the Internet and the anatomy of email messages is a necessary prerequisite for any meaningful discussion of spam and anti-spam techniques.

- Spam *(12%)*
  - Product primer

    A brief explanation of pharmaceuticals hawked by spammers and the different forms of "opt-in" was used to familiarize students with the language used in spam and its meaning.
  - Amassing email addresses

    A variety of methods for acquiring addresses, including dictionary attacks, enumeration attacks, and harvesting.
  - Anti-harvesting techniques and harvester counterattacks
  - Cleaning and verifying email lists
  - Bulk email software techniques, open relays, open proxies, and zombies

    Real examples of bulk mailing software, and utilities for address harvesting and mailing list management were used throughout. Additionally, one of the assignments required the use of an open proxy in the secure laboratory.
  - CGI hijacking

    In this first offering of *Spam and Spyware*, BGP hijacking was left as a research question for students on the take-home final exam.
  - "Bulletproof hosting," "pink contracts"
  - Web sites for spammers, and "jump pages"

- Anti-spam *(15%)*
  - Manual spam tracking methods

    While not the primary focus of the course, students were introduced to the methods, terminology, and the major players in the area of manual spam tracking.
  - Rate limiting and bounce rate monitoring
  - Blacklisting and RBL mechanics

    Students were introduced to RBLs much earlier in the course: the usage of RBLs appeared as a question on a written ethics assignment.
  - Whitelisting
  - Greylisting
  - Tarpits
  - Challenge-response systems, CAPTCHAs, and their problems
  - Proof-of-work systems
  - Sender Policy Framework
  - DomainKeys
  - Filtering: simple filtering, heuristic filtering, checksum-based filtering, and statistical filtering
  - Filter-evasion methods used by spammers

Other orderings of the material are possible. The order of law and ethics is arbitrary, so long as it is covered prior to students being in the secure laboratory. Spam, spyware, and phishing could also be reordered without great difficulty, although there are some minor dependencies between the topics that would need to be resolved.

5. SECURE LABORATORY

As previously mentioned, our secure laboratory environment was created in part by the laboratory protocol, legal agreement, and law and ethics lecture content. In this section, we present the physical and technical aspects of our laboratory's security, along with the laboratory's setup for support of assignments.

In many ways, this laboratory setup was an application of the expertise gained by establishing a computer virus laboratory for another course [1], suitably adjusted for working with spam and spyware. For example, the "spam lab" needed no extraordinary precautions for handling self-replicating code. Just like real biohazard labs have different classifications, we would characterize our spam lab as medium-security, compared to the high-security virus laboratory. Despite the lower security requirement, we subjected the specification of this laboratory to external review.

Physically, the spam lab was located in a separate locked room. The door was re-keyed to ensure that no unauthorized keys were in circulation, and keys were issued to the students in the course. The lab door also had a hydraulic door closer installed.

The lab contained eight student computers, with padlocked cases, and a locked server cabinet containing the lab's server and tape drive backup. All computers were x86 machines running Linux, hardened as much or more than our standard Linux installation (the server, in particular, had all extraneous services shut down, and no remote logins to the server were permitted). Unnecessary computer I/O, like USB ports and CD-ROM drives, were disabled in the BIOS and also physically unplugged when possible. Finally, the spam lab's network was isolated.

Each student computer ran an SMTP server performing local mail delivery, effectively giving students eight different targets to send mail to. For debugging purposes, file access permissions were set so that students could view the SMTP log files. Students worked in groups of two, and two accounts were set up for each group: a "spam" account and a "test" account. The intent was that the spam account would be for development, and the test account would be the victim, but in practice few groups used the test account.

The server machine supplied a DNS server, as well as an open SMTP relay and an open proxy server (SOCKS version 4 [8]). All spam/test accounts were served via NFS, and the server machine also supplied a common directory into which everyone could read and write.

All student computers had an A record in the DNS, and some extra DNS entries were added to allow testing a variety of mail delivery cases. The name `alias` was configured as a CNAME for the machine `notthere` which, as the name suggests, was not there. Instead, `notthere` had an MX record pointing to `localhost`. Students' spamming software could thus attempt mail delivery to hostnames using three different kinds of DNS record.

Following *Spam and Spyware*, all computers had their hard drives reformatted, and all backup tapes were erased.

6. ASSIGNMENTS

Spam and Spyware had five assignments: one written ethics assignment already mentioned, and four programming assignments in the secure laboratory. To maintain a balanced outlook, the assignments were paired so that one assignment involved an offensive technique, the next assignment a defensive one. The assignments given during the first offering of *Spam and Spyware* are below.

Assignment 1: spyware/offensive. Writing spyware that installed a startup hook, changed the browser start page, and performed keylogging. Keylogging was directed at the capture of the username and password used in the web browser to access a fictitious bank's web site.

Assignment 2: spyware/defensive. Students exchanged their spyware from the previous assignment. They then developed anti-spyware software that accurately detected, identified, and removed all spyware samples.

Assignment 3: spam/offensive. Writing bulk mailing software that delivered messages directly to an SMTP server, optionally routing through an open proxy. Because laboratory constraints precluded us from sending a message to multiple recipients in any meaningful way, students instead sent multiple messages to one recipient. Spam and ham corpora (a subset of SpamAssassin's public corpora [11]) were supplied for the students to transmit.

Assignment 4: spam/defensive. Once the email was delivered to some lucky recipient, students developed a spam filter that sorted the recipient's mailbox into spam and ham messages as accurately as possible.

The specifics of assignments will naturally change for future offerings of the course, but this set of assignments is a representative sample.

7. FUTURE PLANS AND CONCLUSION

Our *Spam and Spyware* course is making the transition from a one-off course into a permanent offering, part of a concentration in computer security for our computer science students. We will be adding even more material into the course in the future, including technical material, recent developments, and the examination of more cross-disciplinary topics like marketing. We will also be making some enhancements to the laboratory based on feedback from the students, and the assignments will be modified and have their scope extended. (This latter comment does not indicate deficiencies in the current assignments, but reflects the natural evolution of the course material and our increased experience teaching the course.)

Spam and spyware can be taught safely and effectively. Moreover, given the current magnitude of these problems to our computer-reliant society, educational institutions have a duty to offer courses like *Spam and Spyware*. Education is part of the solution to spam and spyware, but not just end-user education – the next generation of computer scientists must be taught about these problems and how to deal with them.

8. ACKNOWLEDGMENTS

Thanks to the support staff who set up and maintained the laboratory: Jennifer Federl, Darcy Grant, Brian Scowcroft, Beverley Shevchenko, Mike Slywka, Erik Williamson. Alex Shipp and John Graham-Cumming reviewed the laboratory setup, and Greg Hagen and Matthew Prince gave guest lectures. Also thanks to Ken Barker and the students in the course. Shannon Jaeger made helpful comments on a draft of this paper. The author's research is supported in part by the Natural Sciences and Engineering Research Council of Canada.

9. REFERENCES

[1] J. Aycock and K. Barker. Creating a secure virus laboratory. In *13th Annual EICAR Conference*, 2004. 13pp.

[2] J. Aycock and K. Barker. Viruses 101. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, pages 152–156, 2005.

[3] CTV News. Getting a university credit for learning to crash your computer. Originally aired 12 February 2005.

[4] P. Green. University offering e-mail spam 101. *Calgary Herald*, 9 February 2005.

[5] B. Gross Davis. *Tools for Teaching*. Jossey-Bass, 1993.

[6] Health Canada. Laboratory biosafety guidelines, 3rd edition, 2004.

[7] A. R. Korwin and R. E. Jones. Do hands-on, technology-based activities enhance learning by reinforcing cognitive knowledge and retention? *Journal of Technology Education*, 1(2), Spring 1990.

[8] Y.-D. Lee. SOCKS: A protocol for TCP proxy across firewalls.

[9] J. P. Mello, Jr. Malware 101: University offers course on spyware. *TechNewsWorld*, 10 February 2005.

[10] Slashdot. University of Calgary to offer course on spam, 5 February 2005.

[11] SpamAssassin. Public mail corpus. http://spamassassin.apache.org/publiccorpus/.

[12] J. A. Zdziarski. *Ending Spam*. No Starch Press, 2005.