ABSTRACT

In "A Hands-on Introduction on Computer Security" I present 3 1-hour activities intended for undergraduates with no computer science or programming experience. Through group design challenges mediated by the instructor, the students design their own procedures for ensuring privacy and authentication goals based on public key encryption technology. They build up conceptual knowledge from basic encryption to advanced digital certificates and apply these ideas to practical, every-day browsing. The lessons were designed for this audience based on principles of constructivist education and situated learning, while particular attention was paid to affective student experiences and welcoming under-represented groups into computing. This essay reflects my reasoning and justification of the activities and assessments I chose to include in this lesson.

CONTENTS

MOTIVATION AND GOALS

I seek to teach internet security to undergraduates with no prior computing experience. This is an unusual pedagogical approach, and warrants some motivation. Usually, computer security is a topic that only advanced computer science students experience. However, I believe for two reasons that this is a topic that deserves a broader audience: people need a better understanding of security, and security is fun.

First and foremost, I think security is a subject that deserves a more generally educated citizenry. E-commerce is everywhere – you can even log into your bank from your mobile phone these days! Shouldn't users have a better understanding of how their information is kept private other than blind faith in software engineers and a tiny lock icon at the bottom of the screen?

For example, any user of an e-commerce site for banking or purchasing may see a "security" guarantee like the following.

← VeriSign Secured Logo

The natural questions here are: *How is this site secured? Who is VeriSign? Why should I believe them?* My lesson seeks to answer these questions and more in an authentic and application-oriented context.

Second, I think security is just plain fun. Designing a good protocol is like playing a spy game. You have to anticipate an opponent's moves, think outside-the-box, and use interdisciplinary skills from mathematics and psychology. I find this fun highly motivating, and I think most students will too. Thus, this lesson may be well-positioned as a gateway to welcome students into computing in a way that suits collaborative or competitive styles that are normally turned off by the solitary pursuits of introductory programming.

These two motivating factors lead me to my lesson goals. The intention of this lesson is to elucidate the technical and logistical principles that enable *trustworthy* communications over the internet, with a concentration on *identity verification* (who am I exchanging data with?) and *privacy* (can anyone else see my data?). I hope to help students understand the *why* and *how*, with an emphasis on design considerations and practical applications.

A major goal is to avoid the implementation details (how encryption happens) and focus on the reasoning behind the system (what assumptions are made about the keys that can unlock this system?). To that end I have formatted the lesson as a series of design challenges that allow students to use technological tools like encryption and key generation "out of the box", so they can focus on designing the process of passing data through the internet from start to finish and understanding the assumptions and limitations.

A second goal is to make the learning process challenging, creative fun. The team-based competitions and collaborations should hopefully inspire camaraderie and joviality that can make the learning effortless and desired rather than forced. I want to inspire students to learn from each other and ask good questions that *they* want the answers to. Ultimately, I hope this fun can be a catalyst for motivating further interest and pursuit of computing, especially for under-represented groups.

COGNITIVE AND AFFECTIVE RATIONALE

Several themes mark my lesson as distinct from typical security approaches: its emphasis on design considerations, its focus on solving authentic problems, its constructivist, hands-on approach to learning, its almost-exclusive reliance on group work, and its casting of the instructor as guide rather than leader. This section explains the reasoning behind these choices along both cognitive and affective dimensions. The affective dimension is especially important, given my focus on novices. At the end of my lesson, I want students to leave with improved self-efficacy regarding internet ("I *can* understand what is going on behind the scenes") as well as curiosity ("I *want to* understand"). Bonney et al.[1] name these as among the top attitudinal factors affecting STEM education. To succeed at inspiring knowledge and curiosity, I have carefully designed both the lesson plan and its live execution, bringing key concepts and lessons from educational research to bear in both cases.

Why the emphasis on design?

Within computer science, encryption and authentication remain challenging and active problems that require applying rigorous theoretical knowledge (discrete mathematics, encryption algorithms) with contextual design concerns (What algorithm should I use? How should my certificate template look?). Most disciplinary education focuses on the former, so my lesson's emphasis on the latter should be well-received and help create more capable, application-oriented students.

---

[1] C. R. Bonney, T. M. Kempler, A. Zusho, B. P. Coppola, P. R. Pintrich, "Student learning in science classrooms: What role does motivation play?," in S. Alsop (ed.) *Beyond Cartesian Dualism: Encountering affect in the teaching and learning of science*, Springer, 2005, pp. 83-97.

Why a situated "authentic challenge" format?
Existing research suggests situated learning as good practice for application-oriented education. As Brown et al. [2] observe, "Math word problems … are generally encoded in a syntax and diction that is common only to other math problems." Many introductory security problems are encoded in a similarly detached syntax that plays out imaginary interaction between "Alice" and "Bob" [3]. As a beginner I found these examples unmotivating and difficult to grasp, since communication usually happens between an individual and a business, not two random people. As an instructor I hope to encourage creative thinking and ownership, not stock answers to hypothetical problems. Each design challenge as well as the case study and assessment ask students to answer questions very similar to the ones practioner's would pursue.

Why a hands-on constructivist format?
Computer science (esp. security sub-disciplines) can often appear to outsiders as impenetrable. Entry-level understanding is often blocked by unknown jargon and explanations that use concepts favoring insider knowledge. Many online tutorials do not present material cleanly and compellingly for a novice. For example, neither Verisign's own tutorial on digital certificates nor Wikipedia's article escape undefined jargon such as "hash function", "public key infrastructure", or "web of trust" within the first few paragraphs [4] [5]. Non-experts deserve an accessible introduction to web security, both for their edification and to stimulate interest in computer security. My lesson strives to adopt a jargon-limited approach that uses analogies (digital certificate = driver's license) to bridge gaps in understanding.

Making concepts accessible by constructing rather than delivering jargon has affective, experential benefits as well. For example, Aschebacher et al.[6] quote one student (Diana) whose strong academic self-image was shattered by a miserable science experience. Reflecting on an incomprehensible chemistry lesson, Diana remarks "'Is this lady speaking the same language that I speak?! Some of these words are just clueless." To avoid this jargon-laden turn-off, I intentionally designed my lesson to build concepts from the group up and only afterwards assign them labels like "digital certificate".

Why team-based activities?
Turning the design exercise into a team experience adds collaborative and competitive elements that can inspire team camaraderie and reinforce learning. Also, team contexts reinforce the "authentic" experiences students are likely to find if they go on to make security decisions or recommendations in industry.

Why an instructor-as-guide approach?

My current lesson includes several diverse self-driven activities (e.g. group challenge, case study) and I don't want to set students lose on a problem they do not understand. Any prolonged feeling of being

---

[2] A. S. Brown, A. Collins, & P. Duguid (1989), "Situated cognition and the culture of learning," *Educational Researcher* 18(1): 32-41.

[3] http://en.wikipedia.org/wiki/Alice_and_Bob
[4] http://www.verisign.com.au/repository/tutorial/digital/intro1.shtml

[5] http://en.wikipedia.org/wiki/Public_key_certificate

[6] Pamela R. Aschbacher, Erika Li, Ellen J. Roth, "Is science me? High school students' identities, participation and aspirations in science, engineering, and medicine," *Journal of Research in Science Teaching*, in press, 2010.

lost could immediately confirm for a student that "computer science is hard" and that "I'm not capable of succeeding here". Additionally, I want to create an atmosphere where students can approach me and ask questions without fear of looking stupid, since I cannot possibly avoid all confusion. Deliberately leaving time for the instructor to circulate and answer specific questions will be critical, and I have left ample wandering time within each design challenge. A mandatory recap period at the end of each design challenge should also reinforce concepts and provide an instructor timely indications of students who are falling behind so interventions can be made.

ASSESSMENT REFLECTION

This lesson has two primary cognitive goals: to provide students with a situated learning experience to think like an Internet security expert, and to impart practical knowledge about browsing the web securely. This assessment procedure intends to both reinforce and extend learning of both objectives and provide an indication to the instructor of the level of mastery.

Two activities will be assessed. First, students will undergo a summative team design challenge in which they concoct governing policy for a Certificate Authority. They will give a ten minute presentation of their recommendations to the entire class, and this presentation's content and justification will be the basis of assessment. Second, students will individually complete a take-home exercise in which they use Firefox or another modern browser to evaluate an unknown digital certificate, list major assumptions involved in accepting the certificate, and demonstrate that they can use the browser settings to decide whether to trust or deny its authority. Each will submit a screencast of their process (at most 5 minutes in length) for evaluation.

1) *Team Design Challenge: Presentation on Certificate Authority Policies*

This exercise is highly situated and contributes to students "thinking like security professionals". They must develop creative answers to open-questions such as "how will you revoke certificates?" or "how do you verify a company's identity?". The presentation of a carefully justified proposal is similar to the kind of self-verification that might exist at a real corporate authority like VeriSign, Inc. The required questions will both revisit concepts built up in previous lessons (e.g. key size as indicator of time until breaking) and extend new ones (e.g. revocation), thereby fulfilling the learning goals of the assessment process.

2) *Take-Home Exercise: Screencast Explanation of Trustworthiness*

This exercise fulfills the practical usage goals of the lesson. Students should be able to interact with a modern browser to diagnose a digital certificate for a real-world website and adjust security policies to taste. The screencast format encourages metacognition (e.g. "think out loud about your process and justify it") and provides the instructor with both visual and auditory information about what the student can do in a practical context. Successful completion of the exercise should be a summative presentation of what the individual knows about certificates and reinforce concepts learned throughout the unit (e.g. key size is related to expiration date, browsers have an internally maintained list of trusted third parties, etc). Students are also encouraged to use provide at least two things they still wonder about in their narrative (e.g. "where does the browser's list come from?"), which may encourage further learning via reflection.

REFLECTION ON PRACTICE SESSION

I delivered my lesson to six Olin undergraduates (3 first-years, 2 juniors, 1 senior). I had a blast seeing my hard work come to fruition. Overall, students in brief oral feedback at the end of the lesson said they

were "excited", "involved", and "not bored" (which I'm told is a high complement coming from the particular speaker). The experience motivated several opportunities for reflection, especially regarding student inventiveness and delivering clear instructions.

I was amazed by the level of inventiveness and creativity my students displayed. Students were excited to play with the applets and came up with many bits of crucial knowledge on their own, such as "changing any digit in the encrypted text makes it impossible to decrypt" and "encryption is like any abstract function f(x), so it can be composed e.g. f(f(x))." This was a big victory for me as a lesson designer, since my tools taught them more than I probably could have in a lecture format. I could tell they enjoyed the open-ended challenges, they eagerly came up to me and bragged, "I bet you can't hack this now… try it!". While the open format sometimes meant different students came to different conclusions, I think it was overall a great success.

The biggest problem I found in delivering a good design challenge was giving clear, accurate instructions and keeping up with each separate team's progress simultaneously. I think the best way to debug the instructions problem might be to show a list of instructions to a potential student and ask "please explain back to me what is required of you". This short litmus test can help debug clarity without requiring going through an entire practical lesson. Additionally, the detailed student handout packet I've prepared should provide much clearer instructions than my verbal ones on the practice day.

Overall, the practice teaching experience was a big confidence boost that my constructivist approach was successful for my audience and can be a fun way to learn about security. I think my instructor strategy of bouncing between groups to ask/answer questions worked well for the group size (6) but might be difficult to pull off in a group larger than 10 or so. I would recommend that TAs or other resources be enlisted if an instructor pursues a larger group.

REFLECTION ON DIVERSITY

Computer science and engineering remains an especially homogenous field, with slim participation from both women and minorities as compared to the general population. Because my lesson is targeted at novices and would most likely be given as a stand-alone special unit or as part of an introduction to computing series for non-majors, it may reach an audience with much more diversity than is typical for computer science. It thus has the potential for welcoming underrepresented groups into the discipline, and I hope to fulfill that promise.

A huge advantage for my lesson is its hands-on, interactive nature. I want to harness students' natural curiosity as a vehicle for motivating interest in computing. I hope the vast room for experimentation creates a level playing field for all students in which everyone struggles at first but slowly builds confidence in their abilities to discover and invent. This level playing field will perhaps be more welcoming to minorities and underperforming students in general than the established achievement structure of traditional classrooms. Perhaps the biggest win for my approach is its team focus, as collaborative learning is widely believed to support underrepresented groups.

One potential criticism of my current lesson is the "competition" focus, where Attackers act in an adversarial role against the Bankers and Customers. Some might argue that girls would be more receptive to a collaborative model. I would argue that the need for contextualization of the adversary's role in security (e.g. what does it mean to be an Attacker, what different assumptions can they exploit) is essential to developing deep understanding, so I will stick with this format.

A more nuanced issue arises in team roles and dynamics. Given that multiple students will be asked to work as a team at one computer workstation, the possibility that one student (likely female) will do the secretarial recording while another (likely male) will do the "actual work" does exist. However, most of

the valuable conceptual work is done in the planning stages, while the individual who types in the message is merely implementing the plan (albeit in a way that is much more reinforcing than just watching). I did not observe gendered role segmentation in my practice lesson, but perhaps with less-motivated self-starters this would be more apparent. In any case, the high number of iterations in the task at hand should offer every student a chance to fulfill every role, and an instructor should encourage this uniform workload so all students can participate.

Leaving it up to the instructor to assign teams is probably the safest play, though I encourage advanced thought about gender roles and perhaps specific planning to avoid all-except-one gender teams.

One huge challenge for this lesson is communicating instructions for each design task in a way that is both clear and efficient. I have written out detailed instructions for each phase, but this may not be the best for ESL students or those who do not read particularly well in general. Instead, supplementing written instructions with pictograms can illustrate what should happen in a clear, concise way that everyone can understand.