Why do things suck so much?

Student Perspectives on Cyber Security in Undergraduate Education

Emma Plankey

December 14th, 2016

Abstract

The current state of cyber security is bleak. From consistent failures in the private sector, to the drastic failures of the U.S. Government, there is little doubt that cyber security has been under prioritized in program design. Experts in the security field agree that undergraduate computer science programs are doing students a disservice by not integrating security into their curricula. On the other hand, many computer science professors believe that students are spread thin between different foundational topics, and that security should ultimately be considered a lower priority than other skills. Students' point of view on this topic is largely unexplored. This paper seeks to understand students' viewpoints on the importance of cyber security within their education, in the context of the mixed messages they receive from professors and professionals.

Table of Contents

1. Introduction

Over the past thirty years, the tech industry has more than doubled in amount of individuals employed [7]. Though this growth means great things for the job prospects of computer science majors, it also indicates growth in the tech industry's potential for error.

It is a common adage in introductory computer science classes that the main flaw of computers is that they follow their instructions exactly. They are unable to discern the "good" instructions from the "bad". This principle is also foundational to many types of cyberattacks. For example, an SQL database query will execute as written, even if part of the query is given in the form of "bad" user input; that is, unless the program includes "good" instructions to sanitize said user input.

It is human error that is to be blamed for susceptibility to cybercrime. Too little is being done to systematically identify and prevent human error from continuing to occur. This paper seeks to explore how security experts wish this were being addressed, as well as the challenges institutions face with potential systemic change.

For this paper, I surveyed computer science students at Tufts University currently in core classes for the computer science major. There were 77 total respondents. For the full survey, please refer to Appendix a.

2. To the Community

The security industry continues to see the same vulnerabilities it has seen for years [14], as there have been no radical changes to prevent these from happening. As of April 2016, only three of the top fifty computer science programs in the nation required at least one cyber security class for graduation [11].

ABET accreditation requirements set an industry standard for expected outcomes of undergraduate programs. Until very recently, ABET has not required accredited undergraduate computer science programs to include any form of knowledge about security [16]. The most recent set of standards require "The program must enable students to attain, by the time of graduation… An understanding of professional, ethical, legal, security and social issues and responsibilities" [6]. With this addition, it is clear that the tech industry has begun to recognize the root of its security failures: a lack of adequate education in undergraduate programs. In order for this change to have a tangible effect, universities must take it seriously.

2.a. To the Tufts Community

It is the responsibility of every institution of higher learning to consider the ways in which it contributes to systemic failures. The computer science department's curriculum is an extension of its values, and currently, security is not high on its list. Reform of accreditation requirements will result in change on a national scale, and Tufts' Computer Science Department must rise to the occasion and hold itself to a standard of excellence as it does with the instruction of other topics. On the department website, Tufts computer science cites the ABET accreditation requirements as being the goals of its curriculum for the School of Engineering, but not for the

School of Arts and Sciences, despite the fact that the core classes for these programs are identical. Our program must take tangible steps to prioritize security.

3. Differing Perspectives

3.a. Professional Perspectives

It is obvious that specialists will advocate for the prioritization of their area of expertise However, to write off the concerns of security professionals as self-aggrandizing would be a dire mistake. These individuals protect all manner of sensitive information: our online identities, our medical records, and our national secrets. If these are not properly protected, lives can be destroyed and democracy can be compromised.

Security is being discussed too little too late for students to build good mindsets and practices before entering the professional world [16]. Earlier this semester, I conducted an optional recitation on topics in cyber security for students in the introductory computer science course. One attendee, when asked whether they would like to see an increased emphasis on security in the curriculum, responded: “No, in the real world it is the job of the security team to make sure my program is secure”. By not valuing security in our curriculum, Tufts is building the mindset of security teams as being a program’s janitorial staff: people who come in after a system has been built, and clean up the mistakes that others’ have made. Tufts is building the mindset of security as being intrinsically separate from a program’s core architecture. This mindset was referenced by Sandra Carelli, a Product Manager at RSA, in an email interview. When asked what impact increased emphasis on security in undergraduate programs would have on the tech industry, she responded, “Then new grads would be less likely to show up at

development jobs and create code with lots of security bugs. This would be less about preparing students for security careers than about improving the security and quality of software and systems" [2]. Increasing emphasis on security in our curriculum would not and should not build specialists out of all of our students, but rather give them an understanding of how to consider security as one aspect of program design.

When asked what impact increased emphasis on security would have on the tech industry, Nick Davis, a software engineer at Rapid7, responded, "I think security education for undergraduates students… should be *required*. As Gary McGraw (CTO of Cigital) has said many times, you need to build security in from the beginning of the software development lifecycle. Not only is it cheaper and faster that way, but it is also more effective" [8]. There is a lot at stake for tech companies that deal with sensitive information. From companies that deal in social media such as Yahoo [10], to companies that design medical devices [15], clients have an expectation that their information is as secure as possible. To quote one of Davis's colleagues at Rapid7, security analyst Katie Ledoux, "[Security's] *not* required? That seems insane" [12].

3.b. Professorial Perspectives

As the Tufts computer science program continues to grow at an unpredictable rate, it faces the challenge of finding enough instructors and enough seats for students in its existing core classes. Curriculum redesign is secondary to making sure existing students graduate on time. Department Chair Kathleen Fisher, in a recent interview with The Tufts Daily, commented on the need to cap the number of students in classes, "One of the challenges of having these caps on the classes is that the department actually spends quite a bit of energy and resources managing all of these caps

and trying to make sure that the right students get in so that people can graduate," [1]. With the influx of students pursuing computer science related majors, or looking to add it as a skill on their resumés, the overhead of managing the department is increasing as well. There is a paucity of lecturers applying to teach computer science. According to Fisher, "For people who are kind of on the fence, the lure of the better pay is pulling more of them into industry. The fact that jobs are so prevalent in computer science right now is causing more people to not go to graduate school but just go into industry," [1]. Programs cannot expand without the proper manpower.

Professor Ming Chow teaches Intro to Computer Security at Tufts University. The enrollment for the class has close-to-tripled over the past five years [4]. Despite his passion for security, Chow acknowledges potential tradeoffs of requiring it in our curriculum.

> The problem is, there is a cost. There's a big cost in which the instructors have to do a lot more. Let's say if a professor is teaching a 120 person course and you tell them to do security… it's going to be more burden of work on them… But on the flip side of it, that's what they do in most courses in civil and environmental engineering anyway. There's a code of ethics that goes onto each and every course. [3]

Curriculum revision, let alone overhaul, would put a great deal of pressure on a department that is already spread thin. That does not mean it is not worth pursuing, but it is not an easy challenge to approach.

Chow is working on introducing a new course next spring titled Cyber Security and Cyber Warfare. In 2015, he commented to CNN that "Politicians have little knowledge of tech and encryption. Technologists have little understanding of policy. Want to get it right? Every stakeholder needs to be sitting at the same table. The consequences of not getting it right is that

no one wins" [13]. The new course aims to do just that. Increasing the number of security-related electives at Tufts is one way to push back against enrollment caps without over-saturating existing syllabi.

3.c. Student Perspectives

Students want to see an increased emphasis on security in their core classes, and they want it to be introduced early on in the program. In a survey of 77 Tufts students studying computer science, 52% of respondents reported that they think it would be appropriate to introduce security in one of the first two core classes, Intro to Computer Science or Data Structures.

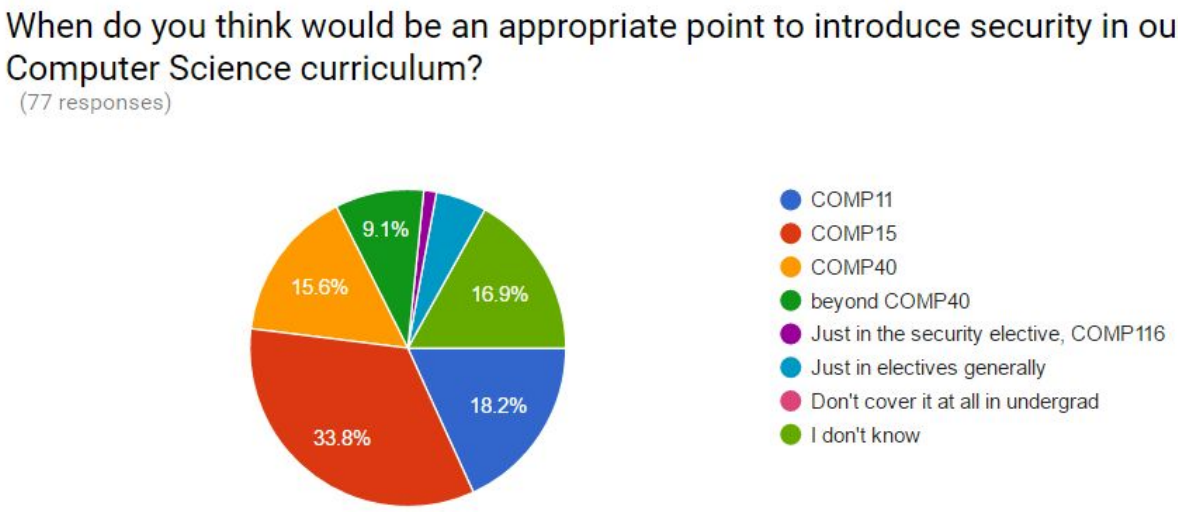


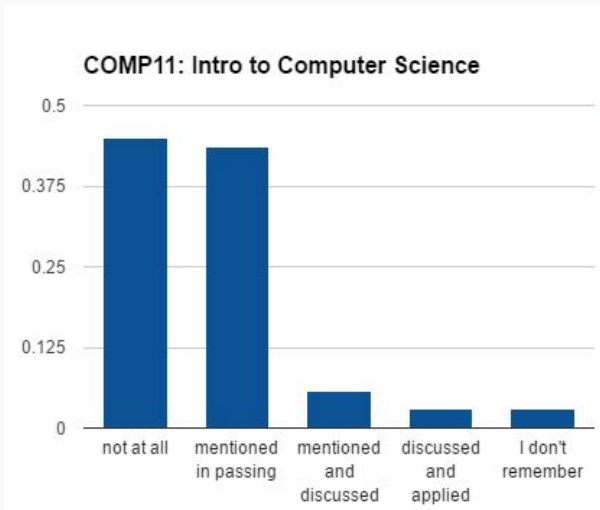
Figure 1:When students want security to be introduced

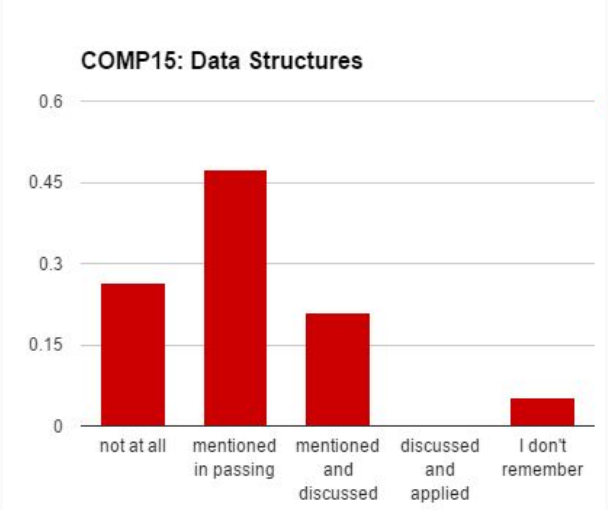
In fact, only one student responded that security should be relegated to the Intro to Computer Security elective, as it is now. That student went on to clarify later in the survey,

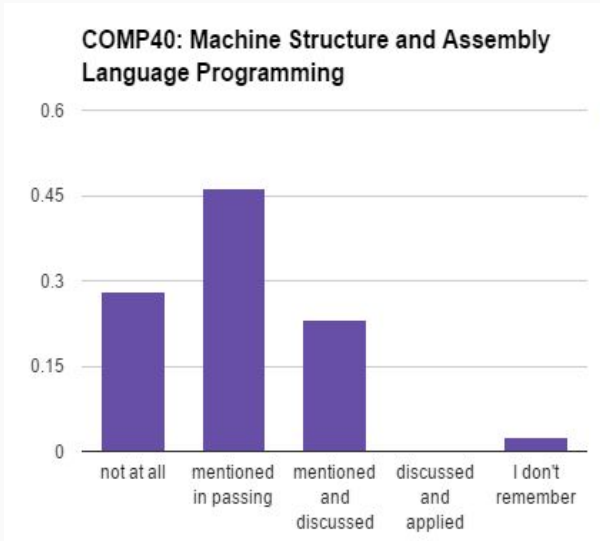
> My response 'Just in the security elective' doesn't accurately capture my feelings. I don't think it is right to shoehorn security in as a major focus in any of the existing core credits

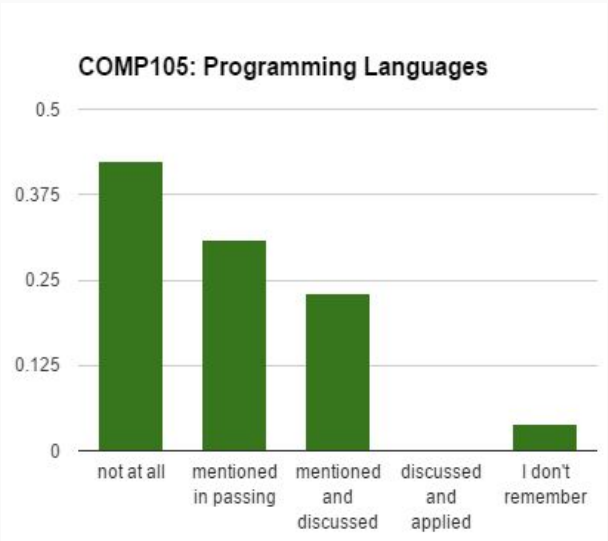
(i.e. 15/40/etc), but I do think that security should be made a required credit for graduation. It's not right to leave an undergraduate program without having learned about best security practices in depth.

The responses suggest students want to learn more about security, but do not necessarily have access or impetus to do so. Several of the students offered additional comments that they wish the security elective were required. It is also important to consider is how students perceive we are currently covering security. When asked about how much security is discussed on a class-by-class basis, they responded:









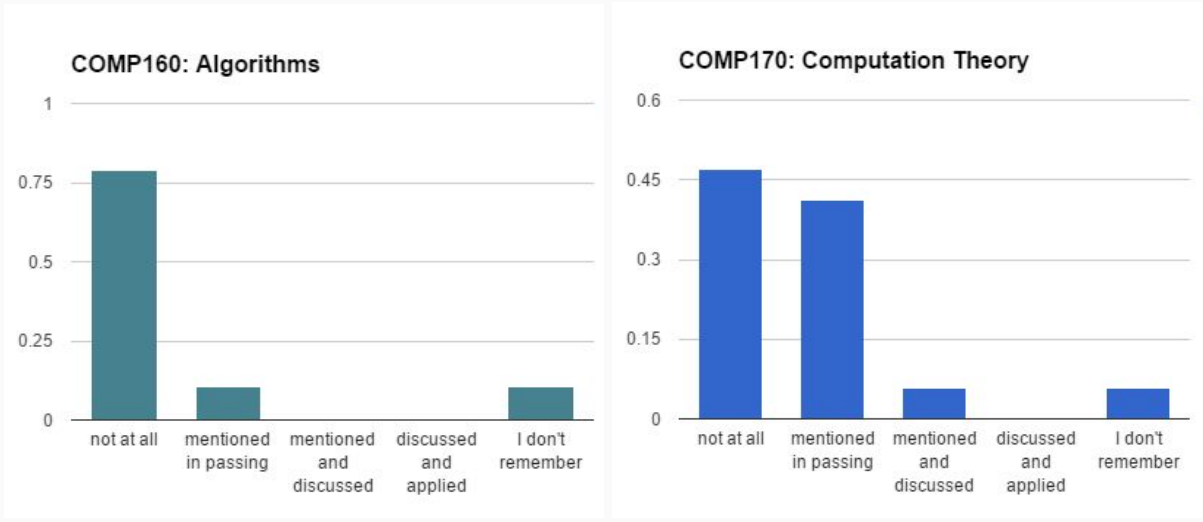


Figure 2: How students perceive security is covered in core classes

Across these core classes, only two students replied that they had ever applied something related to security for their core coursework.

One can argue that students who do not know much about security might not recognize when topics they cover relate directly to it For example, the curriculum for Intro to Computer Science covers input validation. Because this topic is often discussed in terms of the user misunderstanding instructions rather than in the context of the user trying to manipulate and break a program, students might not recognize that they are, in fact, applying an important security feature to the programs they write for the class. Likewise, the Data Structures course covers public and private attributes of classes, and students are asked to consider the difference when implementing classes. Because this topic is discussed just in terms of architecture and good practice, rather than explicitly in terms of security, students might not recognize that their implementation decisions affect how secure their design is. These results suggest that when students do have access to topics in security, they don't always draw the connection between the topic and how it relates to security.

In a personal interview, Jeremy Colebrook-Soucie, a computer science student in the School of Engineering, expressed concern that mandating security in core classes could both detract from the depth in which courses cover their main topics, and not do security justice in the process. “Of all the required courses, it doesn’t really make sense to teach security in any of the Big Four [upper-level classes]… I don’t really think there’s a spot in the core curriculum right now to teach security, and I think that any attempt to insert it would be half-assed” [5]. As much as students are interested in security, they are also mindful of tradeoffs that must occur in the curriculum design process.

4. Fixing this Mess

Our technological world is relatively insecure. Hackers have shown that our identities [10], our elections [9], and even our toasters are subject to their influence. This is an issue that has been known about and discussed since L0pht testified before United States legislators in 1998 [14]. It is not news, and it is not solved by discussion ad nauseum. Now is the time for action, for repairing existing systems, and for holding ourselves to a higher standard.

4.a. For Universities

Educate students on the importance of security considerations when designing programs. A curriculum that prioritizes security will teach its students to prioritize security.

In the short term, talk about how the existing curriculum relates to security. For example, relate input validation to how it can be used to prevent command injection. Relate the worst-case runtime of an algorithm to how slow a nefarious user could force your program to run [16].

Relate pointers to the potential for a buffer overflow. In revising how existing topics are covered, security can be discussed and applied in most core classes while minimizing potential costs.

In the near future, expand the selection of security electives. Currently, three of the top ten computer science programs in the U.S. do not so much as offer a security elective [11]. At Tufts, with enrollment caps in courses and only one section of Intro to Computer Security being taught yearly [4], students have to compromise their desire to learn about security with the logistic nightmare of graduating on time. Because Tufts is hiring a new Cyber Security and Policy Bridge Professor this semester, there will be increased potential for course offerings relating to security. Take advantage of it.

In the long term, consider requiring students to take a course in security before they graduate. Weigh the costs and benefits. Consider the ethical implications of not doing so. Examine how much security can be integrated into existing syllabi, and how much material students might still be missing. There is no one-size-fits-all solution to this.

4.b. For the Industry

Recognize that changes in undergraduate education will not come immediately, and will not retroactively apply to those who have already graduated. Take responsibility and ownership of the actions and flaws of the programmers under your employ. Create professional development programs that are focused on improving the security of applications you create.

Recognize that, while not everyone can or should be a security expert, there are financial and PR benefits to making sure every programmer has a basic understanding of how to think about security. Make sure programmers can recognize security issues so that they know when to

reach out to the specialists for help. Make sure programmers do not treat the security team like janitorial staff that cleans up their repeated mistakes.

5. Conclusion

Cyberattacks will never be completely eliminated. So long as the internet is a free space where one's actions are neither constantly monitored nor controlled, there will be those who use this freedom to malicious ends. That being said, our education and employment system could be doing much more to maintain a high standard of security.

6. References

1. Burke, E. (2016, December 6). Computer science department works to meet growing student demand. *The Tufts Daily*, Retrieved from https://www.tuftsdaily.com/
2. Carielli, S. (2016, December 3). Email.
3. Chow, M. (2016, December 6). Personal Interview.
4. Chow, M. (2016). Courses @ Tufts. Retrieved from http://mchow01.github.io/courses/
5. Colebrook-Soucie, J. (2016, December 5). Personal Interview.
6. Criteria for Accrediting Computing Programs (2016-2017). Retrieved from https://www.abet.org/
7. Csorny, L. (2013, April 9). Careers in the growing field of information technology services. *U.S. Bureau of Labor Statistics*, Retrieved from https://www.bls.gov/
8. Davis, N. (2016, December 6). Email.

9. Entous, A. Nakashima, E. Miller, G. (2016, December 9). Secret CIA assessment says Russia was trying to help Trump win White House. *The Washington Post*, Retrieved from https://www.washingtonpost.com/
10. Fiegerman, S. (2016, September 23). Yahoo says 500 million accounts stolen. *CNN*, Retrieved from https://www.cnn.com/
11. Jackson Higgins, K. (2016, April 7). Top US Undergraduate Computer Science Programs Skip Cybersecurity Classes. *DARKReading*, Retrieved from https://www.darkreading.com/
12. Ledoux, K. (2016, December 8). Email.
13. Pagliery, J. (2015, December 8). Congress puts terrorism and tech in the spotlight. *CNN*, Retrieved from https://www.cnn.com/
14. Timberg, C. (2015, June 22). A Disaster Foretold -- And Ignored. *The Washington Post*, Retrieved from https://www.washingtonpost.com/
15. Weise, E. (2016, October 5). Johnson & Johnson warns of insulin pump hack risk. *USA Today*, Retrieved from https://www.usatoday.com/
16. Zatko, S. (2016). Rethinking the Role of Security in Undergraduate Education. *IEEE*, *14*(2), 73-78.

Appendix

A.a. Survey Instrument

What is your expected year of graduation? *

- ○ 2017
- ○ 2018
- ○ 2019
- ○ 2020
- ○ Post-Bachelor

What is your relationship to the Computer Science Department? *

- ○ Computer Science Major
- ○ Computer Science Minor
- ○ Computer Engineering Major
- ○ Cognitive and Brain Science Major
- ○ Cognitive and Brain Science Minor
- ○ Post-Bachelor program
- ○ none of the above

What core classes have you taken in the Computer Science Department? (including this semester) *

- ☐ COMP11: Intro to Computer Science
- ☐ COMP15: Data Structures
- ☐ COMP40: Machine Structure and Assembly Language Programming
- ☐ COMP61: Discrete Mathematics
- ☐ COMP105: Programming Languages
- ☐ COMP160: Algorithms
- ☐ COMP170: Computation Theory

How confident are you that, given enough time, you could write a functional program that lacks any major flaws? *

	1	2	3	4	5	
not at all	○	○	○	○	○	without question

Describe your feelings about the topics/workload currently covered by core classes? (in a few words to a couple sentences per class) *

Your answer

Which of your classes has touched upon security? In how much depth was it discussed? *

	not at all	mentioned in passing	mentioned and discussed	discussed in depth	discussed and applied	I don't remember	I haven't taken this class
COMP11	○	○	○	○	○	○	○
COMP15	○	○	○	○	○	○	○
COMP40	○	○	○	○	○	○	○
COMP105	○	○	○	○	○	○	○
COMP160	○	○	○	○	○	○	○
COMP170	○	○	○	○	○	○	○

When do you think would be an appropriate point to introduce security in our Computer Science curriculum? *

- ○ COMP11
- ○ COMP15
- ○ COMP40
- ○ beyond COMP40
- ○ Just in the security elective, COMP116
- ○ Just in electives generally
- ○ Don't cover it at all in undergrad
- ○ I don't know

Is there anything else you'd like to add?

Your answer

A.b. Interview Transcripts

A.b.i. Tufts University Professor Ming Chow

What kind of effect an increased emphasis on security in undergraduate education would have on the tech industry at large?

- What I think? If you're gonna increase the emphasis on security, you're actually gonna have people thinking about it. Just having a clue. Just even thinking about it. I'm gonna give you a basic example-- if you're gonna have a civil and environmental engineering program, I mean, if you're gonna build a bridge or you're gonna build a building, you have to think about health and safety paramount. And so, if you actually think about security in a computer science perspective, it's the same idea. Maybe people will actually think about things like safety first. Right now, they don't even know. So right now, everything is just being churned out, being churned out, being built built built. That's it.

What kind of costs: educationally, monetarily, or otherwise, do you think increasing the emphasis on security would have on the Tufts Computer Science Department?

- Yeah, I think it's… the problem is, there is a cost. There's a big cost in which the instructors have to do a lot more. And it's kinda hard to do. It's really almost gotten to the point where it's deflating to ask. Let's say if a professor is teaching a 120 person course, and you're gonna tell them to do security, to cover security, it's going to be more burden of work on them.

- For example, perfect example, right now in COMP40, buffer overflows are not covered. I think Norman got rid of that a long time ago. But if you bring it back in, you're gonna be adding more content, cramming more stuff into a course. But, on the flip side of it, it's… that's what they do in most courses in civil and environmental engineering anyway. There's a code of ethics that goes into each and every course.

How has the enrollment for your security elective changed over the course of the past couple of years?

- I like this question because I've thought a lot about this. The first time I ever taught security, it was like 20, 30 students. Now it's gotten bigger and bigger each and every year. I think the one nice thing about the enrollment of security that is also very nice, is that not only has it increased, there's a lot more awareness that we've given for some of the courses I teach here.

- The other thing that's nice, is that, in terms of the gender ratio, you're in the class and you can confirm, it's like 60/40. It's not like you're gonna have eight to one or nine to

one. And that happens in some computer science courses. And that's a very good thing. Having a pretty even ratio is a big goal of mine, because as you know, women in cyber security is only like 5, 10%, some extremely low number. So having more female involvement in the course is also very near and dear to me.

Tell me a little bit about the new security policy class that you'll be helping to teach next semester. Specifically, why did you create the course, and what do you intend to accomplish with it?

- So here's the problem. The big problem with cyber security is not the tech piece. It's the policy piece. The non-tech piece. As you see right now, look in this room. How many times have you, Emma, if you wouldn't mind, how many times have you talked to a non-CS person about security?

Uhh, several while I'm at home. They always seem very interested, but like.. They always seem very willing to listen, but not willing to take it any further than listening to what I have to say. They don't really care.

- And that's unfortunately happening right now. I mean, this is the same issue as fake news. The reason we're making the security policy course is right now, we have policy makers that don't know a damn thing about the technical side of things. And you have us, the computer science folk, who don't know a damn thing about the policy side of things.
- So the goal of the course is this; to have a sustained, interdisciplinary, and joint discussion on cyber security, so everybody is sitting at the same table. Because right now, we could not be living in a more polarized world where people can't even do that.

Excellent. So to that end, what are the qualities you are looking for in the bridge professor that we are looking to hire.

- Someone that can communicate with... not only explain the CS topics, not only needs to have a good technical underpinning, but more importantly can explain the technical issues, and cyber security to the non-tech folks. That's the big thing.

A.b.ii. Tufts University School of Engineering student Jeremy Colebrook-Soucie

Are you interested in security as a topic in Computer Science?

- In general, absolutely not. In specific cases there are some interesting topics within security like cryptography, or maybe some of Kathleen Fisher's work. Those are

interesting problems in and of themselves. Their application as problems related to security in particular, that has no relevance. It's just that they're interesting problems.

Do you find that classes you've taken tend to mention security? To what extent is it discussed?

- So we don't really have a software engineering class here. That's worth talking about. Like, of all the required courses, it doesn't really make sense to teach security in any of the big four. Perhaps it should be a required course, I'm not sure.
- We're not a school teaching software engineering; we're a school teaching computer science, and a lot of students go into software engineering. So I don't really think there's a spot in the curriculum right now to teach security. Any attempt to insert it is just like, well this is kinda half-assed, why? Plus it's in courses where it matters, like Ming's course [COMP20], it's talked about very reasonably there, right? He recognizes the importance of it.
- I mean the thing is, Tufts isn't building software engineers, right? Which isn't to say it shouldn't teach those skills, but it certainly isn't as high priority as it would be if Tufts were

What is your understanding of the distinction between software engineering and computer science?

- Sure yeah okay so if you look at a school like WPI, they have a software engineering course, and maybe the closest thing we have here is the engineering senior praktikum, the two course sequence. Well, they have a bunch of those, where it's like, go build like an app with a UI and backend fully integrated and go do all the design, all the data-piping, go do that yourself. And that's preparation for building real-world software. And Tufts does not do any of that. We've got like, here is this like, let's learn about the low-level hardware on a machine. That's not software engineering, right? It's teaching good design practices, but it's really, really only COMP40 that can kind of count towards it. Like, 105 certainly isn't, 160 would probably be in that cirriculum but certainly isn't software engineering, and 170 sure as hell is not. Yeah, so it's that theory vs praktikum i guess.

With regard to theory vs praktikum and you mentioned earlier how we are a CS based curriculum that churns out a lot of people that go into software engineering, so these are people that push code to production that theoretically millions of people will end up using across the country, worldwide. How do you reconcile that with having theoretically no knowledgebase in making this code that they're pushing to production secure? And how confident are you, at large, that applications that you use are secure?

- There's two questions there. The later one-- I mean, I'm not.

And are you okay with that?

- I mean, I think that the applications where I have a lot of trust placed, are like, banking applications, and like Google. Both of which I have faith in. Perhaps somewhat misguided, but I think that those are as secure as they could be. There's no work that banking companies through regulation and potential shit-hitting the fan, and that's insured by the government, so that's comforting I suppose. And google because… could they do anything better? And perhaps that's misguided, but I have trust there. So is it definitely secure? No. But like, I still have faith. All the other crap, ehhh, no not really, but that's okay though.

- As for the other one, I don't think Tufts has a responsibility to teach students security. It's really good that it's offered. And that's incredibly valuable. Should not be required. Definitely not. And even then, I don't know. I feel like maybe security is a holistic part of a software engineering degree, so maybe engineers should have that be a required course. I think that's reasonable. For ABET accreditation to say that you have to take a security course to be an accredited computer science BSCS, whatever it is. For that degree of accreditation, you have to have a security course. That's really reasonable. For the Liberal arts side, it's like, ehh, I don't know, and even then I'd be super salty if someone made me take a security course.

A.b.iii. Rapid7 Software Engineer Nick Davis

Would you hire a recent graduate to work on a security team? If so, what level of competence would you expect?

- While I am not on my companies internal security team, I would say that I would absolutely advocate for hiring recent graduates onto our security team. I think that building talent internally is the best way to get truly excellent team members, and although it is always a bit of an upfront investment, if you hire correctly it will pay off in the long term. When hiring recent graduates, I don't except them to have high levels of real-world skill. I measure competence for entry level team members as the ability to learn quickly, adapt, and consume information independently where possible. I think at least a basic level of security knowledge would be necessary. As an example, if I were hiring a junior Application Security Engineer, I'd want that candidate to know the difference between strcpy and strncopy (if it was for a C/C++ code base), as well as be able to define for me what a stack overflow is, what a heap overflow is, and what sql injection is.

What kind of effect do you think would be produced by requiring some form of security education in undergraduate computer science programs?

- I think security education for undergraduates students (I'm assuming you are talking about CS students) should be *required*. As Gary McGraw (CTO of Cigital) has said many times, you need to build security in from the beginning of the software development lifecycle. Not only is it cheaper and faster that way, but it is also more effective. Security is becoming more and more prevalent in every portion of the software industry (as well as the hardware industry but that's a different topic entirely). Just as we teach new web development technologies (Ming teaches MongoDB in Comp 20), we should be teaching modern security methods as well.

A.b.iv. Rapid7 Security Analyst Katie Ledoux

Would you hire a recent graduate to work on a security team? If so, what level of competence would you expect?

- "We do hire recent grads on our team! Because we don't have a very formal onboarding/training program, we prefer when these new grads have internship experience or at least one "real" job".

What kind of effect do you think would be produced by requiring some form of security education in undergraduate computer science programs?

- That's *not* required? That seems insane (I obviously wasn't a comp sci major).

A.b.v. RSA Product Manager Sandra Carielli

Would you hire a recent graduate to work on a security team? If so, what level of competence would you expect?

- It depends on the type of team and the size of the team. If it's a reasonably large IT Security organization where the graduate would have the opportunity to work with and learn from more senior people on the team, then yes. And the hire could work on Security Operations, Compliance, or Product Security - as long as there were more senior people on the team to work with, then great.

  If it's a small organization with only a few people on the security team, then I don't think I'd hire a recent grad - I'd need people who had a broad set of experience and could hit the ground running - a recent graduate wouldn't have that, and there wouldn't be a lot of senior people there that would take the time to mentor them.

  Assuming that I had an organization that could support hiring a recent graduate, what level of competence would I expect? Hmmm - expect or hope for? :-) Given the state of security education today, I would expect someone who had a foundational

understanding of the basics - coding, scripting, maybe networking or systems, maybe some basics of crypto. I'd hope they had an understanding of secure coding principles, and I'd love if they had an understanding of the basics of system architectures - but that might be asking a lot. I wouldn't expect them to have used any particular products, but I'd be looking for people that demonstrated that they were fast learners, eager learners, able to apply principles in new ways, logical thinkers, and team players.

What kind of effect do you think would be produced by requiring some form of security education in undergraduate computer science programs?

- I think there would be huge benefits for industry if CS students had to study security - particularly secure coding practices, secure architectures, and the impacts of vulnerabilities. Then new grads would be less likely to show up at development jobs and create code with lots of security bugs. This would be less about preparing students for security careers than about improving the security and quality of software and systems.

  Now, as I mentioned during my visit, I also think there would be huge benefits in also requiring classes in communication and soft skills for someone interested in security. I also think there's something to be said for a public policy class that focused on security and technology.