Science Fiction Prototyping and Security Education: Cultivating Contextual and Societal Thinking in Computer Security Education and Beyond

Tadayoshi Kohno
Computer Science and Engineering
University of Washington
yoshi@cs.washington.edu

Brian David Johnson
Future Casting, Interactions and Experience Research
Intel Corporation
brian.david.johnson@intel.com

ABSTRACT

Computer security courses typically cover a breadth of technical topics, including threat modeling, applied cryptography, software security, and Web security. The technical artifacts of computer systems – and their associated computer security risks and defenses – do not exist in isolation, however; rather, these systems interact intimately with the needs, beliefs, and values of people. This is especially true as computers become more pervasive, embedding themselves not only into laptops, desktops, and the Web, but also into our cars, medical devices, and toys. Therefore, in addition to the standard technical material, we argue that students would benefit from developing a mindset focused on the broader societal and contextual issues surrounding computer security systems and risks. We used *science fiction (SF) prototyping* to facilitate such societal and contextual thinking in a recent undergraduate computer security course. We report on our approach and experiences here, as well as our recommendations for future computer security and other computer science courses.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Computer Science Education

General Terms

Security, Human Factors.

Keywords

Computer science, computer security, prototyping, science fiction, science fiction prototyping, security mindset.

1. INTRODUCTION

Computer security is the subfield of computer science dedicated to the design and analysis of computer systems in the presence of adversaries. These adversaries – a.k.a. "attackers" or "hackers" – are people seeking to maliciously compromise computing systems. Unfortunately, hackers do exist and they can cause serious damage. Today's threats of phishing, identity theft, and spam are well known, and new threats will emerge as computers embed themselves more intimately into our environments and daily lives. Consider, for example, the teenager who recently hacked a train control system and caused 12 people to be injured in the resulting accident [16], or the recent discovery of security vulnerabilities in modern automotive computers [15] and wireless implantable medical devices [10].

There have been numerous explorations of methods for helping students learn the technical skills necessary to protect computer systems against attackers, including both early works, e.g., [17], [22] and more recent works, e.g., [1]. Outside the classroom there have also emerged undergraduate and graduate cyber-security competitions in which students apply and expand their knowledge *in situ*. While these are great advances, three key challenges remain. First, because technology is evolving at a very rapid rate, any technical skills taught today may become dated and thus less applicable in the future. Second, because of the field's breadth, any single course or short sequence of courses must undoubtedly omit some information about technical computer security defenses – but it could be that the omitted information is exactly what a student might later need to protect his or her system against certain attacks. And third, even if a student learns all the relevant technical materials, simply knowing *how* to protect against a computer security threat *after* the threat has been identified and deemed important does not mean that one necessarily knows to look for threats in the first place, how significant those threats are once identified, what new threats might manifest in the future as the technology or its deployment environment evolve, when to apply the relevant defenses, whether there might be negative consequences to applying those defenses, and so on. Laced over this entire discussion is the fact that computer security risks do not just apply to laptops, desktops, and the Web, but can also apply to any product with an embedded computer.

A key goal of the undergraduate computer security course at the University of Washington is to help students learn how to *think* about the real world and broader societal contexts surrounding computers and computer security. Doing so directly addresses the third challenge above. Such thinking skills empower students with the ability to answer questions like those that we posed earlier. We believe that if students can learn to think critically and deeply not just about the technologies themselves, but about the interactions between technologies and society, then they will be well equipped to identify potential computer security risks when they design and deploy new computer systems – risks that they might not have otherwise identified. This applies both to traditional computing systems, like desktops and the Web, and to emerging technologies. Such a skill also makes the first two challenges above less relevant. As long as students will, after their graduation, be able to identify and reason about the potential security risks with their systems, they will be able to work with

SIGCSE'11, March 9–12, 2011, Dallas, Texas, USA.

more current technical experts to overcome those risks; on the other hand, an inability to identify potential computer security risks is almost a guarantee that those risks will not be guarded against in the resulting system. Additionally, some security defenses can be invasive to people's lives or have unexpected consequences, and the designer of security systems must be cognizant of those possibilities. Our initial approach to helping students cultivate this broader contextual thinking required students to (1) regularly review current events related to computer security and (2) analyze the security of newly announcements products; we describe this approach in a recent book [7].

This past year we embarked on a new approach for helping students develop broader contextual thinking around computer security. Our new approach is based on the budding field of *SF prototyping* [12]. Originally targeted towards engineers, SF prototyping is a process through which the creators of new technologies use science fiction storytelling to help them envision their technologies in the context of future society, understand the nuances of their potential future uses, and – informed by their increased understanding of how those technologies might interact with people in the future – revise their designs accordingly. We adopted SF prototyping to our course. The principal goal was to use SF prototyping to help students develop a mindset for thinking deeply about the security of computing technologies in the context of future users and society. An additional expected benefit was to provide students the opportunity to exercise their communications and creative thinking skills in a computer science course. We describe our experiences below, as well as lessons for future courses in computer security and other areas.

2. COURSE CONTEXT AND GOALS

CSE 484, Computer Security, is an optional senior-level course for Computer Science and Computer Engineering students at the University of Washington. Prior to taking CSE 484, students will have taken two introductory programming courses, a data structures course, and a computer architecture course. Many – though not all – students will have taken operating systems or computer networks. Students are therefore familiar with multiple technical aspects of computers.

The course covers a standard range of technical computer security material at the depth commensurate with a 10-week quarter, including applied cryptography, network security, Web security, software security, and usable security. Lectures provide a high-level description of these technical aspects, and homework and software laboratories provide students with an opportunity to dive more deeply into the technical materials. A key fact stressed throughout the course is that computer security issues may arise anywhere there are computers. For example, attackers can affect the brake control computer within a car and forcibly engage or disengage a car's brakes [15], and attackers can compromise a child's wireless toy robot, drive it around, and spy on the occupants of a home [4].

Our use of SF prototyping is aimed at fulfilling the following non-technical course goal: to help students develop a deeper understanding of the broader contextual issues surrounding computers and computer security, as well as an ability to reason about those broader contextual issues. The vision is that, by empowering students with an understanding of and appreciation for the broader context surrounding computer security, students will be better positioned to identify (and therefore address) potential computer security risks with computer systems that they develop, work with, or manage later in their careers, and that their resulting solutions will acceptable to users.

3. SF PROTOTYPING AND CONTEXTUAL EXPLORATIONS

SF prototyping began as a tool to help engineers design technologies that are sensitive to the needs of future users and society [13]. Engineers charged with designing a particular technology would use the science fiction writing process to help them dive deeply into, explore, and understand the future implications of those technologies and possible future derivatives. By placing the technology in a realistic future environment with realistic characters, engineers can develop an informed and thoughtful understanding of how that technology might interact with people in the future. We conjectured that students in any computer science course – including our computer security course – could similarly benefit from the SF prototyping process. Namely, the process of preparing a realistic story set in the future about a chosen technology would force students to think deeply about the interactions between that technology and society. Moreover, the SF prototyping process would also help students develop an intuition for thinking about potential adversaries against future technologies.

Because the greatest learning arises during the *planning* of a story – the development of the characters, the setting, the plot, and the interactions between the technology and the surrounding ecosystem – and *not* the *writing* of a story, and because our course already had significant other requirements, we did not ask our students to write complete stories in prose. Instead, we culminated the assignment with two key deliverables: (1) complete outlines for short stories, and (2) detailed write-ups reflecting upon the lessons the students learned about the technologies when developing the stories.

Those familiar with computer security might ask how this process compares to traditional threat modeling. Threat modeling is the practice of rigorously identifying the adversaries against a system and how those adversaries might try to attack the system. SF prototyping is complementary to, not a replacement for, rigorous threat modeling. The first few lectures of our course – and some of our assignments – are focused specifically on threat modeling. Threat modeling is however only effective if one knows to apply it – i.e., if one knows that there might be a need for computer security, which is not always the case. Our SF prototyping efforts are designed to help students develop a predisposition for evaluating technologies in the context of future users, which would thereby give students a better intuition for when and where security is needed. Additionally, traditional threat modeling alone may not identify all the complex – and potentially important – interactions between technologies and society. For example, threat modeling may find that tattooing passwords on patients' bodies may be a good solution for securing wireless implantable medical devices because the patients will never be without the tattoo; however, threat modeling may not uncover the fact that such tattoos may also negatively remind patients of WWII concentration camps [2]. In short, SF prototyping is a vehicle to help students develop a habit of not evaluating technologies in isolation, but to instead realize that there are multiple layers of complexities with these technologies, to challenge assumptions, and to think deeply about the interactions between technologies and people and the associated potential computer security risks.

We elaborate on the SF prototyping process that we used in our course below. We also note that the creation of SF prototypes for didactical purposes is different from, though complementary to, supported learning through reflection on existing science fiction, e.g., [20].

4. TOPIC SELECTION

We asked students to form groups of up to three people. The first task was to select a technology on which to focus. We gave students three key criteria for selecting their technology to study. Different groups were allowed to pick different technologies.

First, the technology had to be emerging and forward-looking. We wanted to empower students with an opportunity to think outside the box, with innovative new emerging technologies, and creatively envision future implications of these technologies. So emerging technologies like robots or "app stores" for cars were preferred over contemporary and well-understood technologies like Web browsers (unless students could identify some emerging and as-of-yet understudied aspects of such contemporary technologies).

Second, the chosen technology had to interact with other technologies, people, and society in non-trivial ways. An example might be a future wireless continuous glucose monitor for children with diabetes. Parents can monitor the child's glucose levels constantly over the Internet, and call their children or the school's nurse if a child's glucose levels get too low or too high. Such interactions are non-trivial; for example, such a technology could affect the relationship between the parent, the child, and the school. What if the parent was abusive or over-controlling? What if there was a bug in the software that made a parent think that the child was in danger when in fact the child was safe? What if the child wanted to hide his condition from his friends, or wanted to sneak a cookie without his parents finding out?

Third, because this was a computer security course, the chosen technology had to have some relationship to computer security. We explicitly told students that we would be very liberal in our definition of computer security. Any technology would qualify as long as there could conceivably be an attacker who might wish to compromise some desirable property of that technology. Students could therefore pick emerging technologies that might have computer security risks, or emerging technologies that are designed to mitigate security risks.

We encouraged students to explore numerous venues for finding a technology to study. They were highly encouraged to look for potential candidate technologies while reading the news – such as Slashdot or other media sites. This recommendation was aimed at helping students develop a habit of thinking about computer security risks – and broader contextual issues – outside the classroom and throughout their daily activities. For example, suppose a student saw an ad for a new game system. They might ask themselves: would there be interesting computer security issues and/or societal issues with the game system that would make a nice story? Even if they decided not to use this technology in their story, the process of rejecting this technology would have forced students to practice their deep, context-focused thinking about the broader issues surrounding technology.

We also encouraged students to familiarize themselves with emerging technologies presented at both research and industry conferences. For example, we pointed students to the HRI, Ubicomp, and CHI research conferences, as well as the industry CES conference. We encouraged students to consider computer security risks that are not present with today's versions of these technologies, but that might be present in future embodiments. Having students peruse the proceedings of academic conferences had a secondary benefit of helping students develop a better appreciation for the breadth of computer science.

Finally, we provided students with pointers to known information in the computer security research community about potential risks with emerging technologies. The risks are complex enough not to have clear solutions, and hence the known candidate solutions raise complex societal questions like how to appropriately balance security with safety, cost, and device lifetime, or whether it's appropriate for users to change their behavior to increase security (e.g., as an extreme example, patients might receive password tattoos [2], [21]). Examples of such security research abound, though we focused on examples with which we were already familiar, including implantable neural devices [3], other wireless implantable medical devices [10], wireless robots [4], and automobiles with wireless interfaces [15]. We also provided pointers to emerging security systems, like anonymity systems [5] and self-destructing data systems [9] that, if they become more feasible and widespread, could affect the way people interact with technologies.

5. MECHANICS: THE STORY

The key educational aspects of our SF prototyping process are in the *thinking* about the chosen technology in the context of a future society. The resulting story is the driver for this thinking. The story provides a framework for students to evaluate their technologies in realistic settings. We describe the process of developing a complete story outline below.

5.1 The Idea and the Plot

Since our class was targeted at senior-level computer science students – not writing students – we began with a brief introduction to the structure of stories. We stressed that the structural purpose of the outline is the capture the *idea* behind the story and put it into a *plot*. We provided students with the following quote from Alan Moore, comic book writer and creator of *The Watchmen*, *V for Vendetta*, and the *Sandman* series. This quote describes the distinction between the *idea* and the *plot* as follows [18]:

> The idea is what is the story is about; not the plot of the story, or the unfolding of the events within the story, but what the story is essentially about. As an example from my own work (not because it's a particularly good example but because I can speak about the work with more authority about it than I can the work of other people) I would cite issue #40 of *Swamp Thing*, "The Curse."
>
> The story was about the difficulties endured by women in masculine societies; using the common taboo of menstruation as a central motif. This was not the plot of the story – the plot concerned a young married woman moving into a new home built upon the site of an old Indian lodge and finding herself possessed by the dominating spirit that still resided there, turning her into a form of a werewolf.

The *idea* for the students' stories therefore come from the technology that they chose to study. The *plot* of the story is what is explored in the outline and is what allows the students to explore the broader contextual issues surrounding the technology.

We provided students with one more example to help crystallize the difference between the idea and the plot. In the story *Nebulous Mechanisms* [11], the idea of the story comes from a scientific paper [6] exploring the benefits of building irrationality into the artificial intelligence of domestic robots to improve their ability to adapt to complex environments. The plot of the story revolves around a character's investigation of why the robots from the Ceres mine have started going to church on Sundays.

In *Nebulous Mechanisms*, the *idea* is why the story is being told; the idea is what is being worked out in the fiction. The *plot* is what actually happens in the narrative. It is a linear set of events involving characters, locations, and situations where we can explore the implications of the idea. The SF prototyping process therefore involves putting the idea (technology) into a real world setting and seeing how the story plays out in order to develop a better understanding of the idea's affect on and interaction with both the characters and the locations.

5.2 Exploring the Technology: Planning

In planning their stories, students were instructed to consider future versions of the technologies they selected to explore. Students began by asking themselves some basic (and entertaining) questions, including:

- What are the implications of the mass adoption of the technology?
- What is the worst thing that could go wrong and how would it affect the people and locations in the story?
- What is the best thing that could happen and how would it better the lives of the people and locations of the story?
- If this technology were in an average home how would it actually work?

As with the choice of technologies to study, because this was a computer security course, we required an element of computer security in (at least some of) the questions that the students explored. But, as noted before, we explicitly told students that we would take a fairly broad and liberal definition of security.

After gaining some initial insights from these questions, students began to brainstorm about one or more potential broader contextual issues raised by their chosen technology. We stressed that it was important for students to remember that they were placing their topic (idea) in a real world. Even though the goal was a science fiction story, and the real world might be in the future, the world must feel real. It is still governed by the laws and logic of science, for example. We also stressed that real people will still populate this world. These real people will have real problems that have nothing to do with the students' chosen technologies. For example, in the future people will still not want to go to a boring job. In the future people will still fall in love and some will have their hearts broken. In the future we still will feel too lazy to take out the trash.

We suggested that students pick a setting in the "near" future. A near-future setting, rather than a far-future setting, would make it easier for students to place their technologies in a realistic environment. To help students understand the challenges with creating a plausible near-future setting, we provided students with the following passage from a book on writing science fiction [14]:

> Because most science fiction takes place in the future, the backgrounds are largely products of the writers' imaginations. The future can be researched only to a limited extent, for when it comes to saying exactly what the years ahead hold for us, even the most well-informed scientists can offer only conjecture. The SF (science fiction) writer's vision of the future must be detailed and believable, or ultimately the reader will not believe *anything* about the story – not the characters, the motivation, or the plot.
>
> The *near future*. Structuring a story background of near future – twenty, thirty, or forty years from now – is in some way more difficult than creating an entire alien planet in some impossibly distant age, for the near-future background cannot be *wholly* a product of the imagination. The writer must conduct extensive research to discover what engineer and scientists project for every aspect of future life. From that data, the author then *extrapolates* a possible world of tomorrow, one which might logically rise out of the base of the future which we are building today.

This passage also helped to solidify for students that the writing of science fiction requires a deep understanding of the interactions between technology and science / engineering – which is exactly what motivated us to use SF prototyping in our course.

5.3 Exploring the Technology: The Outline

Students culminated their stories with complete outlines. An outline provides a step-by-step description of what happens in the story – "a linear arrangement of related incidents, episodes or events leading to a dramatic resolution [8]." We informed students that brief descriptions would be sufficient to describe event in their outlines. We also provided students with a simple outline structure for their stories, adopted from Field [8]. The plot is broken down into five parts:

- **Act I:** Act I is where one sets up the world of the story and introduces the reader to the people and locations. Act I answers very simple questions like: who are the main characters, where will the action take place, and what is this future society like? Act I also begins to explore an explanation of the students' chosen technology – what is the technology, what is it designed to do, how does it work, and so on.
- **Plot Point I:** A *plot point* is "an incident or event that 'hooks' into the action and spins it around into another direction. It moves the story forward [8]." For our course, the plot point is the implication of the chosen technology on the world within the story. For a typical science fiction story, this is how the science affects the people and locations in the story in a way that is unexpected or surprising. There might be an unexpected risk, possibility, or reaction to the technology, for example.
- **Act II:** Act II is where students explore the implications of Plot Point I on their story's world. What affect does the technology have? How does it change people lives? Does it

create a new danger? What needs to be done to fix the problem?

- **Plot Point II:** Plot Point II describes lessons learned from seeing the chosen technology placed in the real world. What needed to happen to fix the problem? Does the technology need to be modified? Is there a new area for experimentation or research?
- **Act III:** Act III explores the possible implications and areas for exploration from Plot Point II.

For clarity, we provided students with an example outline for the previously mentioned story *Nebulous Mechanisms* [11]. The example outline is also included in a new text [12].

Stepping Back. The outline development process directly facilitates broader thinking about technologies. Students must envision realistic future scenarios for deploying their chosen technologies (Act I), think about how the technologies will be used and uncover potentially unexpected properties (Plot Point I), explore the implications of those unexpected properties (Act II), draw lessons from those implications (Plot Point II), and reflect upon those latter lessons (Act III). Additionally, most students would not simply write one outline from start to finish. Rather, most students would iteratively revise their plots and produce new outlines as they progressively thought more deeply about their chosen technologies – and each level of iteration provides additional learning opportunities and greater insights into the broader contextual issues surrounding the chosen technology.

6. DELIVERABLES

Students completed their SF prototyping projects in two phases. The first deadline was several weeks after the announcement of the project, and the second deadline was several weeks after that.

The first deadline required each group to submit: a short description (at most one paragraph) of the technology that they planned to explore (including background references); a short (at most one paragraph) description of the broader contextual issues that the students anticipated encountering; and a short (at most one paragraph) synopsis of the envisioned story, including key plot points and other story artifacts. We carefully read each of these reports to gauge progress and spot unforeseen obstacles. We also provided feedback to students, including suggested questions that the students might ask themselves in order to more deeply explore the broader contextual issues surrounding the technologies and to help them continue to shape their the stories.

The final deadline required students to submit a short description of the technology that the students explored (in case the choice of technology changed between the first and second deadlines) and a story outline as described in Section 5.3.

The final step in the SF prototyping process is to reflect on what one has learned from taking the chosen technology through the story development process. Therefore, we also required students to submit a short (two- to three-page) reflection on the broader contextual and societal issues surrounding their chosen technology. Given the focus of this course, we asked students to primarily focus their reflection on the computer security issues surrounding their chosen technologies. We asked students to discuss how their understanding of the relevant computer security issues evolved through the SF prototyping process. For example, did this process expand their understanding of the risks or potential defenses and, if so, how? What issues were raised that the students did not foresee? We also asked students to reflect upon how they would design future technologies to address the issues that they uncovered.

Finally, to facilitate cooperative learning and the sharing of lessons, we encouraged students to post their outlines and reflections to the online class forum. We also encouraged students to discuss others' outlines and reflections on the forum.

7. REFLECTIONS AND NEXT STEPS

We found numerous benefits to the use of SF prototyping in our computer security course. As evidenced both from student feedback, and our subjective observations of student progress, the SF prototyping process successfully catalyzed broader thinking about the contextual and societal issues and computer security risks associated with emerging technologies. This broader contextual thinking complemented the more traditional technical components of our course, such as threat modeling, cryptography, software security, Web security, and usable security.

Many students also enjoyed this project, as evidenced by feedback like "The process of developing our science fiction prototype was very informative, in addition to being fun and rewarding" and "It was great to do an assignment in a computer science class that actually got an alternate stream of imagination and creativity flowing, and I hope this prototyping assignment is given in future iterations of this class." Our experience suggests that because of the fun, non-traditional nature of this project, many students were more motivated to think about it outside the traditional classroom and lab setting, thereby further aiding in its educational impact.

There were however some students who were surprised to find such a strong creative writing component in a senior-level computer science course. We briefly discussed this project at the beginning of the term, but did not officially assign the project or provide full details until after the course's drop date. Some students might prefer computer science courses with less creative writing components, and hence might have preferred to drop the course. We therefore suggest clearly describing this project, and its scope and purpose, at the beginning of the term to minimize any misalignment in expectations.

A second downside with our approach is that the final science fiction outline can only reasonably explore one or two broader contextual issues in depth (the plot points). Some students might have explored numerous other broader contextual issues in depth before finalizing on the ones for their stories, but other students might not have. Informed by our successes with SF prototyping, but cognizant of this concern, we anticipate exploring the use of vignettes in lieu of or in addition to SF prototypes in future courses. The use of vignettes, similar to their use in "value scenarios" for system design [19], would allow students to create multiple short scenes to explore particular issues about emerging technologies, rather than full stories around only a few issues.

Our experience with SF prototyping in a senior-level computer security course makes us even more enthusiastic about its potential impact and utility in other courses. A first year undergraduate computer science course on SF prototyping, especially if cross-listed with another department, could appeal to a larger audience than a traditional computer science course. Students in our computer security course explored topics ranging from ubiquitous embedded digital cameras to prosthetics, autonomous vehicles, nanotechnologies, and future uses of biometrics. We expect similarly broad technology coverage in a

first-year undergraduate course. Covering such a wide range of technologies will likely help broaden these students' understanding of what constitutes computer science, e.g., students new to computer science would learn that computer science is much broader than just laptop, desktop, and Web computing. This awakening could attract new students to the field. Additionally, the broader contextual thinking established in this first-year undergraduate course would help future majors develop a predisposition for thinking about the broader world in which technologies may be deployed. We believe that this perspective would prove valuable to students throughout their careers.

8. CONCLUSIONS

Computers do not exist in isolation, but rather interact intimately with people and society. This is particularly true as computers become more pervasive and begin to embed themselves throughout our environments and daily lives. Given this trend, we argue that it is important for students in computer security courses to develop a mindset for thinking about computers, computer security risks, and their defenses in the context of people and society. Further, we argue that the ability to think rationally about computer security risks can be useful to students throughout their careers, whereas knowledge of today's latest and greatest security weaknesses and defenses may prove ephemeral as technologies evolve. Therefore, and in addition to covering standard technical materials, we introduce a non-standard component into our undergraduate computer security course: SF prototyping.

Our experiences suggest that SF prototyping can be a valuable facilitator for broader contextual and societal thinking about computers, computer security risks, and security defenses. Our experiences also suggest natural next steps for the evaluation of this effort, such as the use of vignettes instead of or in addition to complete stories in upper-division courses. Our experiences also suggest that SF prototyping could serve as a valuable, entertaining, and enjoyable bridge for exposing lower-division students in other disciplines to the breadth of computer science. We plan to explore these latter observations as future work.

9. ACKNOWLEDGMENTS

We thank all the students in CSE 484 for their participation and feedback, as well as the other instructional staff: Slava Chernyak, Alexei Czeskis, and Miro Enev. This work was supported in part by NSF Award CNS-0846065.

10. REFERENCES

[1] S. Bratus, A. Shubina, and M. E. Locasto. Teaching the principles of the hacker curriculum to undergraduates. In *SIGCSE*, 2010.

[2] T. Denning, A. Borning, B. Friedman, B. T. Gill, T. Kohno, and W. H. Maisel. Patients, pacemakers, and implantable defibrillators: Human values and security for wireless implantable medical devices. In *CHI*, 2010.

[3] T. Denning, Y. Matsuoka, and T. Kohno. Neurosecurity: Security and privacy for neural devices. *Neurosurgical Focus*, 27, July 2009.

[4] T. Denning, C. Matuszek, K. Koscher, J. R. Smith, and T. Kohno. A spotlight on security and privacy risks with future household robots: Attacks and lessons. In *Ubicomp*, 2009.

[5] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *USENIX Security*, 2004.

[6] S. Egerton, V. Callaghan, and G. Clarke. Using multiple personas in service robots to improve exploration strategies when mapping new environments. In *Intelligent Environments*, 2008.

[7] N. Ferguson, B. Schneier, and T. Kohno. *Cryptography Engineering: Design Principles and Practical Applications*. Wiley Publishing, Inc., 2010.

[8] S. Field. *Screenplay: The Foundations of Screenwriting*. Dell, 1979.

[9] R. Geambasu, T. Kohno, A. A. Levy, and H. M. Levy. Vanish: Increasing data privacy with self-destructing data. In *USENIX Security*, 2009.

[10] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *IEEE Symposium on Security and Privacy*, 2008.

[11] B. D. Johnson. Nebulous Mechanisms. In *Intelligent Environments*, 2009.

[12] B.D. Johnson. *Science Fiction Prototyping: A Framework for Design*. Morgan and Claypool, 2010.

[13] B.D. Johnson. Science Fiction Prototypes Or: How I Learned to Stop Worrying about the Future and Love Science Fiction. In *Intelligent Environments*, 2009.

[14] D. Koontz. *How to Write Best Selling Fiction*. Writers Digest Books, 1981.

[15] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage. Experimental security analysis of a modern automobile. In *IEEE Symposium on Security and Privacy*, 2010.

[16] J. Leyden. Polish teen derails tram after hacking train network, 2008. http://www.theregister.co.uk/2008/01/11/tram_hack/.

[17] P. Mateti. A laboratory-based course on Internet security. In *SIGCSE*, 2003.

[18] A. Moore. *Alan Moore's Writing for Comics*. Avatar Press, 2008.

[19] L. P. Nathan, B. Friedman, P. Klasnja, S. K. Kane, and J. K. Miller. Envisioning systemic effects on persons and society throughout interactive system design. In *Designing Interactive Systems*, 2008.

[20] D. Sanderson. Using Science Fiction to Teach Computer Science. In *WWW@10*, 2004.

[21] S. Schechter. Security that is meant to be skin deep: Using ultraviolet micropigmentation to store emergency-access keys for implantable medical devices. In *USENIX Workshop on Health Security and Privacy*, 2010.

[22] G. White and G. Nordstrom. Security across the curriculum: Using computer security to teach computer science principles. In *National Information Systems Security Conference*, 1996.