2016-12-19

Crafting a national cyberdefense, and preparing to support computational literacy

Arquilla, John; Guzdial, Mark

ACM

Crafting a National Cyberdefense, and Preparing to Support Computational Literacy

John Arquilla considers how we should interpret the alleged Russian cyberattack on the U.S. Presidential election; Mark Guzdial describes the potential benefits of a 'computing lab.'

John Arquilla
The Real Lesson of the Alleged Russian Hack
http://bit.ly/2hMB9yb
December 19, 2016

What a pity that senior leaders in the American government and intelligence community have decided to play political football with the alleged Russian hacks of John Podesta's and other Democrats' email. By using these intrusions to gin up fears about the "integrity" of the electoral process—which is already befouled by the focus on finding and spreading dirt on the opposition— the real story is being neglected. And what is that real story? It is that, despite more than two decades of consistent public warnings that have reached the highest levels of government, cybersecurity throughout much of the world is in a shameful state of unpreparedness.

Take the U.S., for example. Since the mid-1990s, there have been ap-

proximately 200 cybersecurity bills brought before Congress. Only one has passed, quite recently at that, and it only calls for voluntary information-sharing about cyber incidents. Legislation aside, there have also been several government-sponsored commissions and top-level exercises focused on understanding and illuminating the cyber threat. Each of these has signaled that "the red light is flashing"; that is, American cybersecurity is in very poor shape. Indeed, former cyber czar Richard Clarke and Robert Knake, in their book, *Cyber War* (http://amzn. to/2jEymX3), list the U.S. as having the poorest cyberdefenses among the leading developed countries.

The situation around much of the rest of the world is not much better, as the cost inflicted upon societies— not to mention the wide social and political disruption caused by hack attacks—is staggering. In a speech at the American Enterprise Institute in 2012,

General Keith Alexander, then head of the National Security Agency and the Cyber Command, reckoned annual global losses at more than $1 trillion. As he put it, this was the "largest [illicit] transfer of wealth in human history." [Full disclosure: I have worked for General Alexander, and continue to do so for Cyber Command.] The situation has only become worse.

Whatever the American role in global leadership in other areas might be, when it comes to cybersecurity, Washington has been sadly lacking. Even now, in the wake of the alleged Russian hacks, leadership, right on up to the president, has decided to focus upon retaliatory action, rather than on beefing up security. My previous post (http://bit.ly/2enZtrl) made the point that deterrence based on punitive threats and actions will simply not work, so I won't repeat my lines of argument. But I will reiterate that the failure of the deterrence paradigm, when applied to cy-

berspace, means that the world must move decisively toward an emphasis on improving defenses. And it's not rocket science; better use of strong encryption, moving data around in the Cloud, and increasing use of the Fog, all these can make the situation much better.

But the most important lesson to be learned from the hapless John Podesta is that you can't wait for government policy to protect you. Cyberspace is not just the world at your fingertips; it is also a wilderness, and a dangerous one at that. Much as major commercial firms and governmental bodies must improve their own cybersecurity, individuals, too, must bear responsibility for their own security. The situation is somewhat like that described by the historian Frederick Jackson Turner, who thought of the U.S. as a society defined by its long "frontier experience." Americans were always pushing on into the wilderness, and developed a great deal of self-reliance when it came to sustenance and security. So it may be now in the virtual wilderness of cyberspace.

The alternative, reliance on government, is likely to be fraught with political bickering, endless delays, and unsatisfactory results; in the world's most democratic countries, at least. Authoritarians, on the other hand, have quickly adopted strong cybersecurity policies. As Clarke and Knake see such matters, they list North Korea as having the best cyberdefenses in the world, with China and Russia not far behind.

Perhaps, then, the true lesson of the election hack kerfuffle is not to keep making hard-to-prove charges against President Putin, but to look more closely at how he, and others of his ilk, have crafted their countries' cyber defenses.

Mark Guzdial
Designing the Activities for a 'Computing Lab' to Support Computational Literacy
http://bit.ly/2kTtyza
October 17, 2016

When I was growing up, my elementary school had a "Reading Lab," and later, so did my children's elementary school. If students were struggling with a particular reading difficulty, they could go to the lab and get help with just those specific aspects. It didn't matter what grade

they were in (though earlier grades were certainly most common). Reading was considered so important that it was worth having special help in reading.

The book *Proust and the Squid: The Story and Science of the Reading Brain* (http://amzn.to/2kTvIyN) contains interesting insights into what reading experts do to help students overcome challenges in learning to read. For example, learning to read with rhymes is easier for students because they can attend to just the initial sound and only decode the final sound once. Reading out loud rhyming words like "mat" and "rat" and "sat" are easier than "cat" or "pat" (with a hard consonant at the start) because the initial sounds (e.g., "ma") can be extended ("mmmmmmaaaaaa") while the student works to decode the final sound and put it all together.

Schools provide extra help in other areas of literacy that are highly valued.

▸ At the Georgia Institute of Technology (Georgia Tech), we have special help in writing. For example, if a student is having trouble organizing an essay, instructors in a "Writing Lab" teach techniques like using whiteboards in novel ways to brainstorm and develop an outline.

▸ I am a fan of the Math Emporium at Virginia Tech (https://www.emporium.vt.edu/), which is not just for remedial math help, but does help students to learn mathematics at a pace that works for them.

It is becoming obvious that computing is a necessary skill for 21st-century professionals. Expressing ideas in program code, and being able to read others' program code, is a kind of literacy. Even if not all universities are including programming as part of their general education requirements yet (http://bit.ly/29NbjFK), our burgeoning enrollments suggest that the students see the value of computational literacy.

We also know that some students will struggle with computing classes. We do not yet have evidence of challenges in learning computation akin to dyslexia. Our research evidence so far suggests that all students are capable of learning computing (http://bit.ly/2cqaqcD), but differences in background and preparation will lead to different learning challenges.

One day, we may have "Computing Labs" where students will receive extra

help on learning critical computational literacy skills. What would happen in a remedial "Computing Lab"? It's an interesting thought experiment.

I predict one thing that *won't* happen: students won't just program all the time. Learning to program by programming is a high cognitive-load activity (http://bit.ly/2ktg8fa). Students can learn a lot about reading and writing programs by engaging in a variety of other learning activities.

Some of the activities that we might expect:

▸ Parson's Problems (http://bit.ly/2jEvhGv), which are programming problems where the solution is given but the lines of code are scrambled on "refrigerator magnets." Students have to assemble the lines into place. There are never any syntax errors, so students can focus on the meaning of the code. We know that these problems have much lower cognitive load and are useful in learning (http://bit.ly/2kTObLl).

▸ Explaining programs from one student to another, aloud. There is a reading activity called reciprocal teaching (http://bit.ly/2ks0v8b) in which one student reads, and the other probes the understanding of the first student. A similar activity could be constructed for developing program understanding skills.

▸ Tracing programs by hand with pen and pencil. We teach a variety of sketch-based techniques to facilitate learning and practice in mathematics and science classes, from long division and "borrowing/carrying" in multi-digital arithmetic, to balancing equations in algebra and chemistry and drawing free-body diagrams in physics. Certainly, we will need similar sketch-based techniques to help students make sense of their code and data structures, too.

The exercise of defining a "Computer Lab" is not just speculation about a possible future. It helps us as computing teachers to think about what else we can do in our own classes *today* to help struggling students. We need a wide variety of teaching and learning techniques to achieve the goal of "CS for All" (http://bit.ly/2kSGKas). ⓒ