How Can America's Students Compete Against Russia's Hackers?

Rob Kolstad · Delos Enterprises · rob.kolstad@gmail.com

In a weblog published on June 17, 2017, Brian Krebs questioned why so many skilled computer hackers come from Russia:

> "…One reason so many hackers seem to hail from Russia and parts of the former Soviet Union is that these countries have traditionally placed a much greater emphasis than educational institutions in the West on teaching information technology in middle and high schools, and yet they lack a Silicon Valley-like pipeline to help talented IT experts channel their skills into high-paying jobs." [1]

As coach of the USA Computing Olympiad for 20 years, I have traveled to high school Computer Programming Olympics venues around the world, so I've had the opportunity to observe the computing culture of students from many countries. Before examining skill levels in Russia or the motivations of Russian hackers, however, it's incumbent to examine the assertions about the Russian education system to determine whether the Russian and U.S. systems really are different.

The Russian System

Let's take a quick look at the Russian system first. Evgeniy Khenner and Igor Semakin from Perm State National Research University in Russia wrote a definitive paper on computer education in Russia.[2] Two aspects of the Russian system stand out with particular prominence:

- Curriculum breadth, depth, and rigor
- Participation levels

In Russia, the list of actual topics covered in computer training is quite extensive. Here are just three from the list of 10 for the final two years of high school: [2]

- Skills and experience in software development in the chosen programming environment, including testing and debugging programs, basic skills applied to problem formalization, and documentation of programs.

- Creation of digital objects, their properties, algorithms for their analysis, data coding, and decoding; identification of the reasons for data loss or distortion in transmission; systematization of knowledge related to mathematical objects of informatics; and the ability to construct mathematical objects, including logical formulas.
- Basic knowledge of databases, their structure, and the means of creating and working with them.

One of the challenges in reading papers about education is that the nomenclature can lead to drawing conclusions that might not be warranted. Take, for example, the final topic above: "Basic knowledge of databases." Here is the basic knowledge: "Databases store information for quick and easy retrieval using a number of criteria."

If asked about basic definitions like this on a quiz in the United States, a student could regurgitate the statement above and be pretty much assured of full credit. Of course, moving beyond the superficial, such a definition is fairly useless for real-world technical application. It doesn't say anything about how to create an actual database, what sort of "information" can be stored, or how to retrieve this "information" or deal with it programmatically after it is retrieved.

The Russian system (especially for mathematics and sciences) is known for its rigor. As Carey Goldman writes in an article on why U.S. parents send their children to Russian math schools: "…The [Russian math] schools' curriculum is based on Russian teaching traditions that emphasize reasoning and deeper understanding early on, not just memorization and practice drills."[3]

Therefore, when a Russian curriculum says, "basic knowledge of databases and creating and working with them," you can bet that students who master that educational topic will know how to create and retrieve real data from a running database. This is one of the most important differences between the U.S. and Russian educational systems.

The number of Russian students who participate in these curricula is much larger than in the United States, in relative terms. Furthermore, Russian students who wish to move on in their informatics studies must take the Unified National Exam (UNE). Khenner and Semakin report that in the past five years, approximately 60,000 students have registered annually for the UNE in Computer Science, about

8% of the total number of high school graduates. Given Russia's smaller population, this means that Russian students take the very rigorous UNE at 2.35 times the rate that U.S students take the College Board APCS exam.[4] Furthermore, those taking the Russian exam were probably headed towards majoring in informatics at a university, while U.S. students were most often looking to test out of CS101 at their chosen higher educational institution or at least enhance their college application portfolio.

Finally, Russian students have far fewer electives than their American counterparts[2] – making their curriculum look almost lockstep – promoting traditional rigor and high expectations. Here's what former Russian student "arkades" said in a discussion forum [12]:

> "More was simply expected of us. Not 'hoped for' or 'aspired to,' but expected. And parents weren't at loggerheads with teachers; what teachers said was law. So if a teacher said you were learning the multiplication table this week, parents didn't argue it was too much, or encourage you to 'do your best.' You'd be drilled on those times tables until you wanted to kill someone, but you'd damn well be expected to have them memorized by the end of the week."

Outside the school system, cultural, social, and economic factors also affect the approach to programming and hacking. Russian economic mobility is generally fairly limited, and programming (even illegal hacking) can be a way to move up – discussions that are beyond the scope of this paper. The gist is, though, that valuable output of programmers is generally easier to monetize in the United States. Social pressures and norms (including gender-specific ones) in the United States and Russia differ greatly, as does the perception of what legal vs. illegal means in terms of obtaining or providing various services. Illegal access to the Internet has been de rigueur in Russia and seems to be viewed with less disdain than in the United States, perhaps paving the way to other sorts of hacking.

The U.S. System

In reviewing the history of computer education in U.S. high schools, it's important to note that while Apples and PCs debuted in schools and homes around 1975 – all with the BASIC programming language included – but by 2000 BASIC had been removed. Thus, by the time high school seniors in 2017 were born, home computers were no longer automatically equipped with a programming language.

This is a decided disadvantage compared to the situation when most folks reading this paper were younger.

In the United States, only a few high schools teach advanced computer courses. An exception is the phenomenal 10 courses (including artificial intelligence and parallel computing) offered at what some call the country's best high school, Thomas Jefferson High School for Science and Technology in Alexandria, Virginia. [5,6]

More typically, unfortunately, most high schools offer only the College Board's Advanced Placement Computer Science course at the A level, the more advanced level having been phased out owing to lack of interest. Computers do show up in many other courses, though, including business applications, web design, and any course that might utilize tools like MSWord, PowerPoint, and/or Excel.

In 2017, the College Board inaugurated the new Advanced Placement (AP) "Computer Science Principles" course to "challenge students to explore how computing impacts the world."[7] This course has dramatically increased participation in the exam by young women and minority students, with over 45,000 total students sitting the first offering of the exam.

Newspapers across the country trumpeted this record-breaking count for the AP exam (which is the "APCS Principles" exam, abbreviated APCSP in the literature). Unfortunately, this is a course where one learns "about" computers, rather than rigorous hands-on application of that knowledge. So while the public relations focused on the quite legitimate value of the course in attracting women and minorities to take it, the course title connotes, to the layman, more complexity and application than the course's syllabus actually requires, as we'll see below.

While some course sections that teach the APCS Principles might require a project that involves programming, the list of six "Essential Questions" for the APCS Principles syllabus section on Programming reflects the contrast with the depth of Russian instruction. Those six questions are:[8]

- How are programs developed to help people, organizations, or society solve problems?
- How are programs used for creative expression, to satisfy personal curiosity, or to create new knowledge?

- How do computer programs implement algorithms?
- How does abstraction make the development of computer programs possible?
- How do people develop and test computer programs?
- Which mathematical and logical concepts are fundamental to computer programming?

Interesting questions all, but none of them requires creation of an actual program in an actual programming language on an actual computer. Furthermore, they are disturbingly similar to questions whose answers can be memorized and regurgitated for an exam. To appreciate the full impact, substitute the word "numbers" for "computer programs" and you'll get an analogous section about arithmetic or mathematics.

Challenges

In sum, the APCSP course seems to focus more on "talking about computers" than on hands-on learning and application of computers skills. These curriculum issues are compounded by yet more challenges that really ought to be solvable:

- Few agree on exactly what "coding" or "programming" are. Is creation of a web page with HTML actually programming? Look at the count of supposedly distinguishable terms for programmers: programmer, coder, analyst, hacker, [software] developer, architect, software engineer, computer scientist, etc. When vocabulary gets muddled like this, it's no wonder that folks are confused about what is being taught in courses.
- Competent computer teachers too often don't stay in academia when they can double their salary and halve their political challenges by taking a job in some computer industry. Of course, some teachers do come from industry, but in general the salary disparity creates the problem that computer teachers often lack industry experience (that is, they are teachers, not scientists or engineers).
- There is massive inertia in developing the necessary public school system curriculum owing in part to decentralized control to states and local school boards. Add in an emphasis on testing/evaluation along with requirements

for teacher certifications and methodology accreditations, and the gears turn even more slowly.

- There is a relatively recent trend toward rote memorization [9,10,11]—which is generally anathema to actual programming—as the focal point of teaching (or at least of learning).

- The College Board drives the APCS curriculum (albeit with teachers' assistance), a problem compounded by the observation that universities often can't agree on how to teach CS101 (as opposed to Calculus 101, which is now a 400+ year-old technology).

On the bright side, some U.S. high schools have plenty of STEM education, including robots, web development, databases, Cisco's networking education efforts, and initiatives such as integrating STEM with the arts (though with few formal computer science courses like discrete math). Science contests and Olympiads abound at the local and state levels (in addition to National and International Olympiads for the extremely gifted). In other words, better students are not held back, but they often find themselves having to use their own resources to gain specific technical knowledge. Both the Internet and local experts seem to fill these voids.

In counterpoint, despite its shortcomings, the U.S. education system does have some sort of superior property that enables the best (or at least the most innovative) students to end up as inventive captains of tech industries. We must take care, then, not to kill the goose laying the golden eggs!

Summary

- Russian and U.S. school systems differ in rigor and depth, expectations of students, and breadth of offered material.
- Russian economics for the masses differs from that of the United States, which affects learning processes in a variety of ways.
- Opportunity and monetization systems in the United States far surpass those in Russia.
- Both countries now have resources for their students to use to gain computer skills. U.S. students aren't widely encouraged to learn and build sophisticated software, while Russian students perhaps find more opportunity for piracy and other sorts of hacking in order to achieve standard

teenage computer goals (like obtaining games and joining large social networks).

Moving Forward

Presuming that technical computer prowess (e.g., computer security and programming) is a desirable skill for U.S. students, how can the United States move forward? Applied educational growth must occur within or outside of the school system and be driven by skilled experts. At present, the process of training teachers, accrediting curricula, and obtaining buy-in all encounter massive inertia that could take half a decade or more to overcome.

Looking outside the school system, we find a few existing organizations with at least somewhat similar technical aspirations that could serve as models for broader U.S. initiatives:

- Codeclub.org.uk, a nationwide network of volunteer-led after-school coding clubs for children ages 9-11 in the United Kingdom. Some 5,933 clubs supply 75 different projects for their 83,000 members to tackle.

- The FIRST Robotics Competition, an international high school team-oriented robotics competition created in 1992. The 384 teams generally have from 10 to 25 students each and fundraise $50,000 to $500,000 for the six-week robot construction season and subsequent weekend trips to regional and national competitions. The program builds camaraderie through its cooperative construction sessions and spirited competitions that include more collaboration and sharing than most sports events. FIRST has singlehandedly filled America's mechanical engineering schools to capacity.

- Capture-the-flag cyber competitions like picoCTF,[13] a computer security game that targets middle school and high school students. The game consists of a series of challenges centered around a unique storyline where participants must reverse-engineer, break, hack, decrypt, or do whatever it takes to solve the challenge. The challenges are all set up with the intent of being hacked, making it an excellent, legal way to get hands-on experience

Volunteer adults and college students form the foundation of these successful programs and provide assistance on the technical side. They are mostly up-to-date professionals who deal with students outside of a classroom setting (i.e., no tests, assessments, credentials, or accreditation, and little administration beyond issues of student safety).

Two of the three programs above are competitions. Some have found, though, that students do not necessarily learn best in competitive environments, since the very nature of competitions means that some parts of the contest will be too challenging for some students and could discourage them. Instructional and collaborative activities (like Codeclub in the United Kingdom) can also yield desirable results. Competitions do, however, provide a fun and extremely motivating goal for secondary students: a trip to a regional or national competition.

CyberStart

Embracing ideas similar to the ones above, the SANS Institute has partnered with the government of the United Kingdom to foster the online CyberStart program with 300 hours of challenging and engaging narratives that tackle everything from Linux to cryptography and programming to forensics.[14] Students proceed at their own pace through an extensive suite of challenges, tools, and games that introduce them to the field of cybersecurity, develop their interest, and enable them to consider the field as a career.

CyberStart is not a spoon-fed curriculum – students often have to research solution techniques and must try several solutions in order to solve any given challenge. Best of all, the curriculum is respectful, treating students as peer cybersecurity interns instead of as minions or junior trainees.

CyberStart and other initiatives mentioned above can provide the underpinnings of a new way to enable U.S. students to catch up and surpass their Russian peers in the serious business of cybersecurity.

References

[1] Krebs, Brian, "22 Why So Many Top Hackers Hail from Russia," *Krebs on Security*, https://krebsonsecurity. com/2017/06/why-so-many-top-hackers-hail-from-russia/, Jun 17, 2017.

[2]   Khenner, Evgeniy, and Semakin, Igor, Perm State National Research University, Russia "School Subject Informatics (Computer Science) in Russia: Educational Relevant Areas," *ACM Transactions on Computing Education*, Vol. 14, No. 2, Article 14, https://www.cs.cmu.edu/˜cfrieze/courses/a14-khenner. pdf, June 2014.

[3] Goldberg, Carey, "Why Thousands Of American Parents Are Sending Their Kids To 'Russian Math'," WBUR *Edify*, http://www.wbur.org/edify/2017/04/13/russian-math-school, April 13, 2017.

[4] College Board, *Program Summary Report*, https://secure-media.collegeboard.org/digitalServices/pdf/ research/2016/Program-Summary-Report-2016.pdf.

[5] Thomas Jefferson High School for Science and Technology, "Computer Science," https://tjhsst.edu/ research-academics/math-cs/computer-science/, March 29, 2017.

[6] Thomas Jefferson High School for Science and Technology, "Computer Science at TJHSST for 2017- 2018," https://tjhsst.edu/research-academics/math-cs/computer-science/docs/FlowCS1718.pdf, 2017.

[7] College Board, "AP Computer Science Principles," https://advancesinap.collegeboard.org/.

[8] College Board, "Course and Exam Description; AP Computer Science Principles . . . ," https://secure-media. collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-course-and-exam-description. pdf, page 26, Fall, 2017.

[9] Boaler, Jo, and Pablo Zoido, "Why Math Education in the U.S. Doesn't Add Up," *Scientific American*, https://www.scientificamerican.com/article/why-math-education-in-the-u-s-doesn-t-add-up/, November 1, 2016.

[10] Gearon, Christopher J., "High School Students Need to Think, Not Memorize," *U. S. News and World Report*, https://www.usnews.com/education/high-schools/articles/2012/09/17/high-school-students-need-to-think-not-memorize, September 17, 2012.

[11] Orlin, Ben, “When Memorization Gets in the Way of Learning,” *The Atlantic*, https://www.theatlantic.com/education/archive/2013/09/when-memorization-gets-in-the-way-of-learning/279425/, September 9, 2013.

[12] “Why So Many Top Hackers Hail from Russia,” *Hacker News*, https://news.ycombinator.com/item?id=14613180, June 16, 2017.

[13] CMU, “PicoCTF: CMU Cybersecurity Competition,” https://picoctf.com/, 2017.

[14] SANS Institute, “CyberStart,” https://www.sans.org/CyberStartUS, 2017.