**Take a Long View: Integrate Security Topics into ALL Software Development Education**

The software development community does a lousy job of delivering software that minimizes the attack surface. In the National Vulnerability Database [8], an exact match search on the keyword Microsoft identifies 275 records for the last 3 months. A similar search on the keywords Linux and Oracle identifies 218 and 326 records, respectively, for the last 3 months. Neither proprietary nor open source software are immune from bad or ignorant secure software development practices. This situation is not new. In the SANS report on the Top 25 Software Errors [10], the current list identifies 16 errors that also appeared in the 2010 list.

Our current state of ineptitude is even more perplexing when one considers that two researchers published eight security principles in 1975 [9], over forty years ago! Five more security principles were described in 2013 [7]. Why aren't these thirteen security principles - economy of mechanism, fail-safe defaults, complete mediation, open design, separation of privilege, least privilege, least common mechanism, psychological acceptability, secure the weakest link, defend in depth, be reluctant to trust, promote privacy, and use your resources – discussed and practiced in all undergraduate curricula that has a role in software development?

There appears to be some positive momentum in emphasizing secure software development in undergraduate computing programs.

The most recent computer science undergraduate curriculum guidelines (CS2013) represents the first time security was recognized as a separate knowledge area with the inclusion of Information Assurance and Security [1]. The most recent software engineering and information systems undergraduate curriculum guidelines - SE2014 and IS2010, respectively - have significantly increased the visibility of security.

The current CISO of Turner Broadcasting System is calling for a “moonshot to reestablish our digital strength (via) a profound, coordinated effort to bolster our cybersecurity systems and protect our democracy from hackers” [3]. In his book, Chronis draws inspiration and lessons learned from other moonshots – getting a man on the moon, defeating fascism, and eradicating polio. One of his pillars for fixing cybersecurity is to minimize software vulnerabilities through better software development practices, market incentives that provide more information to consumers about the safety and security of products, and software

technologies that make it easier to identify/fix security defects (e.g., self-healing code, deep learning platforms).

It is clear that both educators and industry see the need for vast improvements in how we develop software. The question becomes, how do we cover security topics in our computing-based programs so that we have the greatest impact on the next generation of information technology leaders? While this question pertains to the three curriculum guidelines (CS2013, SE2014, and IS2010) most directly related to software development, only the CS2013 perspective is described below.

One option is to create a separate computer science course that covers cybersecurity. Assuming CS programs make this course a requirement and not an elective; this would likely improve students understanding of security topics and their use in software development. Another option is to integrate security topics into the entire CS program. This is what we have done in our INCUBATE project [4, 11]. One example of this integration is in our CS1 course, where we introduce security principles (e.g., CIA, anonymity, authentication, assurance, and non-repudiation) and input validation, with hands-on exercises that ask students to apply various types of input validation checks. While our assessment results to-date are positive, our first cohort of students that will have experienced four years of integrating cybersecurity topics into CS will graduate in May 2019. While we expect assessment results to be positive for this cohort, the full impact of our efforts will be unknown for at least another 5-10 years, or until these students have gained enough work experience to influence the culture within their respective organizations.

Changing the culture of the software development industry to adhere to security policies and to apply security controls and mechanisms will take time. Perhaps twenty years from now, when current college students start to take on leadership positions, we will see results of the educational decisions we make over the next few years.

**Diversity of Thought: Social and Political Perspectives on Cybersecurity**

Since technology has created our cybersecurity problems, technology can solve these problems. This thinking is shortsighted because it ignores the fact that humans develop and use these technologies, and humans are the source and target of cybersecurity attacks.

Having students study the social sciences as part of a cybersecurity program provides these students with other ways of thinking about the issues that confront us. A workshop on social science, computer science, and cybersecurity held in 2013 [5] had as its goal to develop communities of researchers from social science and technology fields that cooperate in the development of new and improved cybersecurity systems. In the summary report from this workshop [5], white papers written by the attendees provide their perspective on the workshop goal. The following quote exemplifies the workshop discussions in support of the need for educational opportunities that blend social sciences and information & system security technology.

> "The fact that humans from several different walks of life are interacting with these systems on a daily basis has prompted a paradigm shift: rather than designing secure systems with arbitrarily defined use models, we must design secure systems with use models informed by how people interact with each other, computers, and information. This security paradigm necessitates a close collaboration between technical and social scientists so that the design of secure systems incorporates an understanding of the needs and capabilities of the billions of people that will rely on them." (Page 28, Chris Kanich, Computer Science Department, University of Illinois at Chicago.)

In addition, a 2014 paper published by the National Council in the Social Studies [2] includes the following quote.

> "… the disciplines of the social sciences promote ways of knowing and deliberating about data and information that are critical to policy development and the implementation of cybersecurity initiatives. Building the capacity of the next generation of social scientists to tackle these emerging issues is imperative."

While Chronis [3] believes that minimizing software vulnerabilities is crucial to his cybersecurity moonshot, the other pillars of his moonshot relate to social and political perspectives. His other pillars: educating everyone about social engineering attacks; federal government leadership in the form of regulations and incentives; and better corporate governance of their cybersecurity programs.

Le Moyne College launched a new cybersecurity undergraduate program in fall, 2017 developed by faculty in anthropology, computer science, criminology, political science, and sociology [6]. This program has used the Catholic Jesuit mission of *educating the whole person* as motivation for *educating the whole cybersecurity professional* with perspectives in: crime, society & culture; information & system security; and policy & law. Our thinking in developing this new program is to position our students for success in a variety of career paths, some of which may have an ancillary relationship to cybersecurity.

**Bio Sketch**

David Voorhees is an associate professor of computer science at Le Moyne College. He is the director of the computer science, software applications and systems development (i.e., a software engineering program), and cybersecurity undergraduate programs. Dave worked for 19 years in industry before starting as a visiting assistant professor at Le Moyne in August 1999. He earned his Ph.D. in computer science from Nova Southeastern University in 2005. Dave is the PI of the NSF-funded INCUBATE project briefly described in this paper.

**References**

[1] ACM, (2018). *Curricula Recommendations*. Retrieved April 29, 2018 from https://www.acm.org/education/curricula-recommendations.

[2] Berson, M. J., & Berson, I. R. (2014). Bringing the Cybersecurity Challenge to the Social Studies Classroom. *Social Education (National Council for the Social Studies), 78(2), 96-100*.

[3] Chronis, P.K. (2017). *The Cyber Conundrum: How do we Fix Cybersecurity?.* CreateSpace Independent Publishing Platform.

[4] Das, A., Voorhees, D., and Choi, C. (2018). *INCUBATE: Injecting and assessing cybersecurity education with little internal subject matter expertise*. Retrieved April 29, 2018 from http://research.lemoyne.edu/incubate.

[5] Hofman, L. J. (2013). Social Science, Computer Science, and Cybersecurity, Workshop Summary Report. Cyber Security Policy and Research Institute, The George Washington University, Report GW-CSPRI-2013-02 retrieved on October 21, 2016 from https://cspri.seas.gwu.edu/sites/g/files/zaxdzs1446/f/downloads/Final+08+22+13+1301+Report+Social+Science.pdf.

[6] Le Moyne College Catalog, (2018). *New Cybersecurity Undergraduate Program*. Retrieved April 29, 2018 from http://collegecatalog.lemoyne.edu/arts-sciences/cybersecurity/.

[7] McGraw, G. (2013). *Thirteen principles to ensure enterprise system security*. Retrieved on July 28, 2015 from searchsecurity.techtarget.com/opinion/Thirteen-principles-to-ensure-enterprise-system-security.

[8] NIST, (2018). *National Vulnerability Database*. Retrieved April 29, 2018 from https://nvd.nist.gov/vuln/search.

[9] Saltzer, J.H. and Schroeder, M.D. (1975). The Protection of Information in Computer Systems. In *Proceedings of the IEEE, 63(9)*.

[10] SANS, (2018). *CWE/SANS Top 25 Most Dangerous Software Errors*. Retrieved April 29, 2018 from https://www.sans.org/top25-software-errors/archive/2010.

[11] Voorhees, D. and Das, A. (2018). Injecting cybersecurity into a CS program: a non-specialist perspective. *Journal of Computing Sciences in Colleges, 33(6)*.