DOI:10.1145/2700341

Carl Landwehr

Privacy and Security
We Need a Building Code for Building Code

A proposal for a framework for code requirements addressing primary sources of vulnerabilities for building systems.

The market for cybersecurity professionals is booming. Reports attest to the difficulty of hiring qualified individuals; experts command salaries in excess of $200K.[4] A May 2013 survey of 500 individuals reported the mean salary for a mid-level "cyber-pro" as approximately $111,500. Those with only an associate's degree, less than one year of experience, and no certifications could still earn $91,000 a year.[7]

Is cybersecurity a profession, or just an occupation? A profession should have "stable knowledge and skill requirements," according to a recent National Academies study,[5] which concluded that cybersecurity does not have these yet and hence remains an occupation. Industry training and certification programs are doing well, regardless. There are enough different certification programs now that a recent article featured a "top five" list.

Schools and universities are ramping up programs in cybersecurity, including a new doctoral program at Dakota State University. In 2010, the Obama administration began the National Initiative for Cybersecurity Education, expanding a Bush-era initiative. The CyberCorps (Scholarships for Service) program has also seen continuing strong budgets. The National Security Agency and the Department of Homeland Security recently designated Centers of Academic Excellence in Information Assurance/Cyber Defense in 44 educational institutions.

What do cybersecurity professionals do? As the National Academies study observes, the cybersecurity workforce covers a wide range of roles and responsibilities, and hence encompasses a wide range of skills and competencies.[5] Nevertheless, the report centers on responsibilities in dealing with attacks, anticipating what an attacker might do, configuring systems so as to reduce risks, recovering from the aftereffects of a breach, and so on.

This whole economic boom in cybersecurity seems largely to be a consequence of poor engineering.

If we view software systems as buildings, it appears cybersecurity professionals have a lot in common with firefighters. They need to configure systems to reduce the risk of fire, but they also need to put fires out when they occur and restore the building. Indeed, the original Computer Emergency Response Team (CERT) was created just over a quarter-century ago to fight the first large-scale security incident, the Internet Worm. Now there are CERTs worldwide. Over time, CERT activities have expanded to include efforts to help vendors build better security into their systems, but its middle name remains "emergency response."

This whole economic boom in cybersecurity seems largely to be a consequence of poor engineering. We

IMAGE BY THOMAS FREDRIKSEN

have allowed ourselves to become dependent on an infrastructure with the characteristics of a medieval firetrap—a maze of twisty little streets and passages bordered by buildings highly vulnerable to arson. The components we call firewalls have much more in common with fire doors: their true purpose is to enable communication, and, like physical fire doors, they are all too often left propped open. Naturally, we need a lot of firefighters. And, like firefighters everywhere, they become heroes when they are able to rescue a company's data from the flames, or, as White Hat hackers, uncover latent vulnerabilities and install urgently needed patches.[10]

How did we get to this point? No doubt the threat has increased. Symantec's latest Internet Threat report compares data from 2013 and 2012.[8] Types and numbers of attacks fluctuate, but there is little doubt the past decade has seen major increases in attacks by both criminals and nation-states. Although defenses may have improved, attacks have grown more sophisticated as well, and the balance remains in favor of the attacker.

To a disturbing extent, however, the kinds of underlying flaws exploited by attackers have not changed very much. Vendors continue to release systems with plenty of exploitable flaws. Attackers continue to seek and find them. One of the most widespread vulnerabilities found recently, the so-called Heartbleed flaw in OpenSSL, was apparently overlooked by attackers (and everyone else) for more than two years.[6] What was the flaw? Failure to apply adequate bounds-checking to a memory buffer. One has to conclude that the supply of vulnerabilities is more than sufficient to meet the current demand.

Will the cybersecurity professionals we are training now have a significant effect on reducing the supply of vulnerabilities? It seems doubtful. Most people taking these jobs are outside the software development and maintenance loops where these vulnerabilities arise. Moreover, they are fully occupied trying to synthesize resilient systems from weak components, patching those systems on a daily basis, figuring out whether they have already been compromised, and cleaning them up afterward. We are hiring firefighters without paying adequate attention to a building industry that is continually creating new firetraps.

How might we change this situation? Historically, building codes have been created to reduce the incidence of citywide conflagrations.[a,9] The analog of a building code for software security could seriously reduce the number and scale of fires cybersecurity personnel must fight.

Of course building codes are a form of regulation, and the software industry has, with few exceptions, been quite successful at fighting off any attempts at licensing or government regulation. The exceptions are generally in areas such as flight control software and nuclear power plant controls where public safety concerns are overwhelming. Government regulations aimed at improving commercial software security, from the TCSEC to today's Common Criteria, have affected small corners of the marketplace but have had little

a Further history on the development of building codes is available in Landwehr.[3]

effect on industrial software development as a whole. Why would a building code do better?

First, building codes generally arise from the building trades and architecture communities. Governments adopt and tailor them—they do not create them. A similar model, gaining consensus among experts in software assurance and in the industrial production of software, perhaps endorsed by the insurance industry, might be able to have significant effects without the need for contentious new laws or regulations in advance. Hoping for legislative solutions is wishful thinking; we need to get started.

Second, building codes require relatively straightforward inspections. Similar kinds of inspections are becoming practical for assuring the absence of classes of software security vulnerabilities. It has been observed[2] that the vulnerabilities most often exploited in attacks are not problems in requirements or design: they are implementation issues, such as in the Heartbleed example. Past regimes for evaluating software security have more often focused on assuring that security functions are designed and implemented correctly, but a large fraction of today's exploits depend on vulnerabilities that are at the code level and in portions of code that are outside the scope of the security functions.

There has been substantial progress in the past 20 years in the techniques of static and dynamic analysis of software, both at the programming language level and at the level of binary analysis. There are now companies specializing in this technology, and research programs such as IARPA's STONESOUP[1] are pushing the frontiers. It would be feasible for a building code to require evidence that software for systems of particular concern (for example, for self-driving cars or SCADA systems) is free of the kinds of vulnerabilities that can be detected automatically in this fashion.

It will be important to exclude from the code requirements that can only be satisfied by expert and intensive human review, because qualified reviewers will become a bottleneck. This is not to say the code could or should ignore software design and development practices. Indeed, through judicious choice of programming languages and frameworks, many kinds of vulnerabilities can be eliminated entirely. Evidence that a specified set of languages and tools had indeed been used to produce the finished product would need to be evaluated by the equivalent of a building inspector, but this need not be a labor-intensive process.

If you speak to builders or architects, you will find they are not in love with building codes. The codes are voluminous, because they cover a multitude of building types, technologies, and systems. Sometimes builders have to wait for an inspection before they can proceed to the next phase of construction. Sometimes the requirements do not fit the situation and waivers are needed. Sometimes the code may dictate old technology or demand that dated but functional technology be replaced.

Nevertheless, architects and builders will tell you the code simplifies the entire design and construction process by providing an agreed upon set of ground rules for the structure that takes into account structural integrity, accessibility, emergency exits, energy efficiency, and many other aspects of buildings that have, over time, been recognized as important to the occupants and to the community in which the structure is located.

Similar problems may occur if we succeed in creating a building code for software security. We will need to have mechanisms to update the code as technologies and conditions change. We may need inspectors. We may need a basis for waivers. But we should gain confidence that our systems are not vulnerable to the same kinds of attacks that have been plaguing them for an embarrassing period of years.

I do not intend to suggest we do not need the cybersecurity professionals that are in such demand today. Alas, we do, and we need to educate and train them. But the scale and scope of that need should be an embarrassment to our profession.

The kind of building code proposed here will not guarantee our systems are invulnerable to determined and well-resourced attackers, and it will take time to have an effect. But such a code could provide a sound, agreed-upon framework for building systems that would at least take the best known and primary sources of vulnerability in today's systems off the table. Let's get started! C

I am honored and delighted to have the opportunity to take the reins of *Communications'* Privacy and Security column from Susan Landau. During her tenure, Susan developed a diverse and interesting collection of columns, and I hope to continue down a similar path. I have picked up the pen myself this month, but I expect that to be the exception, not the rule. There is so much happening in both privacy and security these days that I am sure we will not lack for interesting and important topics. I will appreciate feedback from you, the reader, whether in the form of comments on what is published or as volunteered contributions.

—Carl Landwehr

References

1. Intelligence Advanced Research Projects Activity (IARPA): Securely Taking on New Executable Software Of Uncertain Provenance (STONESOUP); http://www.iarpa.gov/index.php/research-programs/stonesoup.
2. Jackson, D., Thomas, M. and Millett, L., Eds. *Committee on Certifiably Dependable Systems, Software for Dependable Systems: Sufficient Evidence?* National Academies Press, 2007; http://www.nap.edu/catalog.php?record_id=11923.
3. Landwehr, C.E. A building code for building code: Putting what we know works to work. In *Proceedings of the 29th Annual Computer Security Applications Conference (ACSAC)*, (New Orleans, LA, Dec. 2013).
4. Libicki, M.C., Senty, D., and Pollak, J. *H4CKER5 WANTED: An Examination of the Cybersecurity Labor Market*. RAND Corp., National Security Research Division, 2014. ISBN 978-0-8330-8500-9; http://www.rand.org/content/dam/rand/pubs/research_reports/RR400/RR430/RAND_RR430.pdf.
5. National Research Council, Computer Science and Telecommunications Board. *Professionalizing the Nation's Cybersecurity Workforce?* D.L. Burley and S.E. Goodman, Co-Chairs; http://www.nap.edu/openbook.php?record_id=18446.
6. Perlroth, N. Study finds no Evidence of Heartbleed attacks before flaw was exposed. *New York Times* Bits blog (Apr. 16, 2014); http://bits.blogs.nytimes.com/2014/04/16/study-finds-no-evidence-of-heartbleed-attacks-before-the-bug-was-exposed/.
7. Semper Secure. *Cyber Security Census.* (Aug. 5, 2013); http://www.sempersecure.org/images/pdfs/cyber_security_census_report.pdf.
8. Symantec. *Internet Security Threat Report 2014: Vol. 19.* Symantec Corp. (Apr. 2014); www.symantec.com/content/en/us/enterprise/other_resources/b-istr_main_report_v19_21291018.en-us.pdf.
9. *The Great Fire of London, 1666.* Luminarium Encyclopedia Project; http://www.luminarium.org/encyclopedia/greatfire.htm.
10. White hats to the rescue. *The Economist* (Feb. 22, 2014); http://www.economist.com/news/business/21596984-law-abiding-hackers-are-helping-businesses-fight-bad-guys-white-hats-rescue.

Carl Landwehr (carl.landwehr@gmail.com) is Lead Research Scientist the Cyber Security Policy and Research Institute (CSPRI) at George Washington University in Washington, D.C., and Visiting McDevitt Professor of Computer Science at LeMoyne College in Syracuse, N.Y.