Technology Transfer:

A Software Security Marketplace Case Study

Gary McGraw, Cigital

Gary McGraw is one of those astounding people you meet in our industry—a technical wizard who's also a musician (check out Where's Aubrey at www.wheresaubrey.com) and a chef. He has, in effect, defined software security with a process that is true science in action (check out the BSIMM at http://bsimm.com). He's one of the most thought-provoking yet entertaining speakers and writers I know. I'm happy to include his contribution in the Insights series. *—Linda Rising, associate editor*

HP'S ACQUISITION OF source code analysis tool vendor Fortify (https://www.fortify.com/node/534) in September 2010 marks an important milestone in a decade-long technology transfer story that begins with a federal research grant and ends with a worldwide technology provider with global reach. I've been fortunate enough to occupy a front-row seat for the entire show—as it turns out, the earliest versions of Fortify's technology base were invented in Cigital's research labs way back in the late '90s.

Technology transfer is as difficult as it is rare, most likely because of the time scale involved. The story you're about to read stretches over more than a decade and involves millions of dollars of research and development.

Computer Security Is a Software Problem

Traditional approaches to computer security focused almost exclusively on the network; the idea was to keep malicious hackers away from vulnerable machines by placing a barrier between the two. The network firewall was introduced in the late '80s as a way of creating such a barrier between a local area network and the Internet. Though firewalls certainly have their place in computer security and have since become ubiquitous, serious security problems persist.

Since the late '90s, a new paradigm in computer security has evolved—software security (sometimes called application security), the idea of engineering software so that it continues to function correctly under malicious attack. Although as a discipline software security is relatively young, much progress has been made on ways to integrate security best practices into the software development life cycle. Microsoft, for example, has helped spearhead software security through its Trustworthy Computing Initiative and the resulting Security Development Lifecycle (SDL). Cigital has also been instrumental in bringing software security to the wider market through its professional services.

The BSIMM project (http://bsimm.com) describes 109 activities making up more than 40 full-scale, enterprise-level software security initiatives and provides a measuring stick for them. The three most common software security methodologies (Microsoft's SDL, Cigital's Touchpoints, and the

0740-7459/11/$26.00 © 2011 IEEE

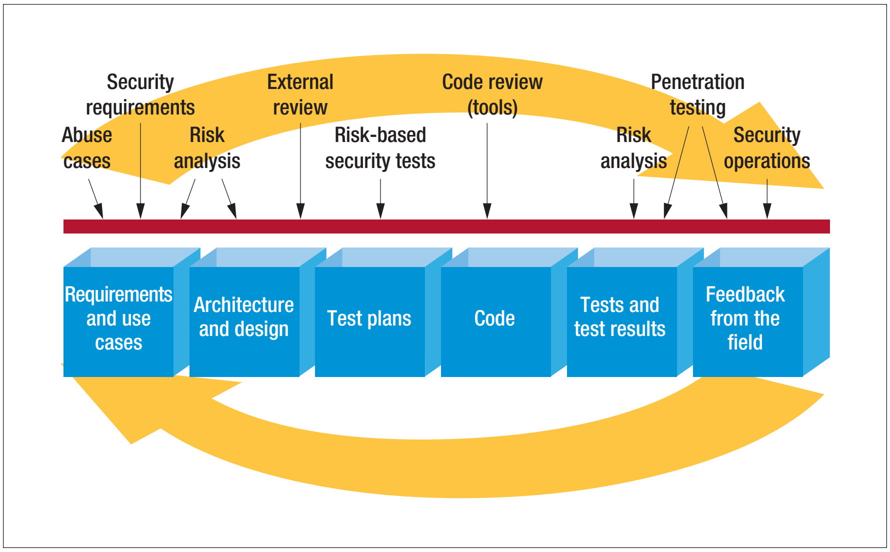


FIGURE 1. Cigital Touchpoints. The Touchpoints approach to software security is process agnostic and focuses on best practices associated with software artifacts commonly created in many diverse SDLCs.

Open Web Application Security Project's [OWASP's] Comprehensive, Lightweight Application Security Process [CLASP]) share a focus on common best practices. Figure 1 shows a picture of the Cigital Touchpoints as described in my book *Software Security* (Addison-Wesley, 2006). Without question, the two most common best practices are code review (with a static analysis tool) and architectural risk analysis.

Among these two best practices, code review with a static analysis tool is the easiest and most straightforward to adopt. There are two reasons for this. First, every software project has code that can be reviewed (they should all have an architecture, too, but that's a topic for another article). Second, code review has been partially automated with sophisticated tools.

This article describes the invention and commercialization of such tools. More information on static analysis technology and its inner workings appear elsewhere.[1]

Born in the Research Lab

Cigital was founded in 1992 as Reliable Software Technologies (RST). In the early years, RST was a scientific research lab funded exclusively by federal grants. Since 1992, several agencies, including DARPA, the NSA, NASA, the National Science Foundation, and the Advanced Technology Program of the Department of Commerce, awarded Cigital more than $15 million in various government grants.

In 1999, Cigital turned its attention from early work in Java security, fault injection, and software testing to software security.[2] Given Cigital's software-centric research focus, it was only natural for us to pursue the notion of scanning code for security problems (especially in Java).

Several of Cigital's early research projects involved work on code scanning, including DARPA contract DAAH01-98-C-R145. The open source release of ITS4 in February 2000 (www.cigital.com/its4) marked an important milestone in source code analysis tools originating at the company. ITS4 was the world's first code scanner for C and C++ code security, but it was far too simple for industrial use; ITS4 was basically a glorified grep engine with some simple vulnerability patterns. In the lab, we were exploring much better compiler- and parser-based technology that took advantage of intermediate representations such as abstract syntax trees and could thus search for more sophisticated patterns. We published this research at a number of academic conferences, including the Annual Computer Security Applications Conference.

Negotiating the Research Valley of Death

The research valley of death (www.allbusiness.com/management/304250-1.html) is defined as the time in a technology's lifespan between its early prototyping in the research lab and its readiness for the kind of cash injection offered at later stages by venture capitalists. Many promising research prototypes languish in the valley of death, never to emerge as full-fledged technologies.

The Advanced Technology Program (run by the US Department of Commerce) exists to help bridge early-stage technologies so that they persist and evolve through the valley of death. Cigital's budding code-scanning prototypes were supported and further developed under Advanced Technology Program cooperative agreement number 1997-06-0005, Certifying Security in Electronic Commerce Components. This ATP research resulted in two patents: US Patent 7,302,707 (static analysis for buffer overflows) and US Patent 7,284,274 (combining static and dynamic analysis for security certification). During that work, we also built a working research prototype code scanner named Mjolner.

Though Mjolner's technical approach to code scanning far surpassed the capabilities of ITS4, it wasn't at all ready for prime time use by non-scientists. In final analysis, the ATP funding supported the work's evolution into an almost-usable tool and certainly helped it negotiate the research valley of death.

Consultingware: Mjolner to SourceScope

Between 2000 and 2002, Mjolner was renamed SourceScope. At worst, SourceScope was a hairy research prototype that required use in concert with a handful of open source tools to actually work. At best, it was "consultingware"—software written for use by savvy, well-heeled consultants willing to forgive its quirks and flaws in order to get some work done. SourceScope did work, but barely. It was supported by an internal engineering team at Cigital called Core Technologies and driven by our consultants' use of it in the field.

During this time, Cigital delivered SourceScope only in the form of consulting engagements for code review. Attempts to sell the technology directly to users always ended in failure—mostly because the technology was too difficult for normal developers or security analysts to use. SourceScope was able to ferret out more interesting source code vulnerabilities than ITS4 (and ITS4's closely related cousin RATS), but using it was painful and involved a nontrivial understanding of how to navigate source code while reviewing code during the build process.

Venture Capital to the Rescue

In 2003, Ted Schlein, a partner at the venerable Silicon Valley venture capital firm Kleiner Perkins Caulfield & Byers (www.kpcb.com) contacted me. Knowing that Kleiner was the VC responsible for incubating such companies as Google, I immediately dropped everything and flew out to meet him on Sand Hill Road. Ted wanted to start a company in the software security space. Roger Thornton, one of Fortify's cofounders, was already involved in the project.

After intense discussions and negotiations, Cigital licensed the SourceScope technology and its associated rules to the Kleiner startup that eventually became Fortify. At that time, Fortify had four employees, all founders.

Cigital's SourceScope technology was delivered wholesale to the Fortify engineering team, who proceeded to tear it apart and create a real software product from its guts. Fortify's engineers and scientists spent huge amounts of time and money transforming SourceScope from barely working consultingware into a commercial-grade software product. They assembled a world-class engineering team. They lived with early customers. They hired usability consultants. And they kept a relentless focus on creating an excellent and usable software tool.

After seven years percolating at Fortify—time that included several product release cycles and use by hundreds of real customers—the technology originally hatched in the labs at Cigital was finally ready for prime time. Between 2005 and 2009, the market for software security grew steadily larger (www.informit.com/articles/article.aspx?p=1623792), spurred on in no small part by static analysis tools.

The biggest players in technology took notice of the software security trend and have since been bulking up in the software security tools space. Their first purchases were black-box Web application testing tools (www.cigital.com/papers/download/0411sec.appsec-tools.pdf). Next came the static analysis tools for white-box code review. IBM purchased Ounce Labs, and HP purchased Fortify. Competition between these two global technology providers should be fierce and will certainly help develop the software security market even further.

To be sure, much work remains to be done on source code analysis, regardless of this technology transfer success story, and source code analysis does not by itself solve the software security problem. The current set of commercial code review tools all have limitations, especially when it comes to dataflow capabilities. At the end of the day, this story teaches us an important lesson—the nontrivial amount of time, money, and sweat that technology transfer really takes.

ABOUT THE AUTHOR

GARY MCGRAW is Cigital's chief technology officer. He's also the author of *Exploiting Online Games* (Addison-Wesley, 2007), *Software Security: Building Security In* (Addison-Wesley, 2006), and seven other books. McGraw has a BA in philosophy from the University of Virginia and a dual PhD in computer science and cognitive science from Indiana University. Contact him at gem@cigital.com.

Acknowledgments

This article is adapted by permission from Gary McGraw, "Technology Transfer," ***informIT*** **(26 Oct. 2010); www.informit.com/articles/article.aspx?p=1648912.**

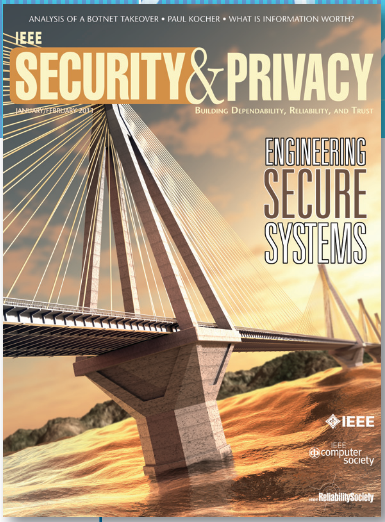
References

1. G. McGraw, "How Things Work: Automated Code Review Tools for Security," *Computer*, vol. 41, no. 12, 2008, pp. 92–95.
2. G. McGraw, "Software Assurance for Security," *Computer*, vol. 32, no. 4, 1999; www.cigital.com/ssw/softsec_infosec.pdf.

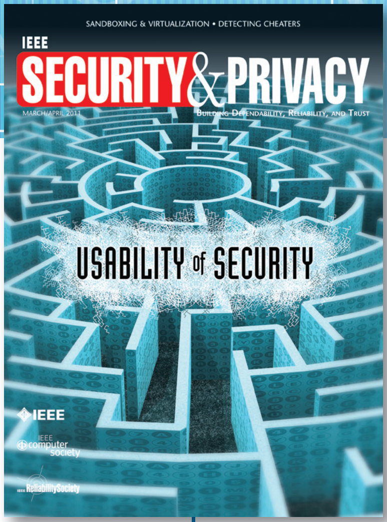
IEEE SECURITY & PRIVACY

$19⁹⁵

Subscribe to *IEEE Security & Privacy!*



ANALYSIS OF A BOTNET TAKEOVER • PAUL KOCHER • WHAT IS INFORMATION WORTH?
IEEE SECURITY & PRIVACY
BUILDING DEPENDABILITY, RELIABILITY, AND TRUST
ENGINEERING SECURE SYSTEMS
IEEE
IEEE computer society
ReliabilitySociety



Protect Your Network

Stay Ahead of the Competition

Prevent Attacks

> “*IEEE Security & Privacy* is the publication of choice for great security ideas that you can put into practice immediately. No vendor nonsense, just real science made practical.”

—**Gary McGraw**,
CTO, Cigital, and author of
Software Security and *Exploiting Software*

www.qmags.com/SNP
for your digital subscription