Smooth Sailing for Application Security

So many of an organization's business practices are now pushed to the Web. In-house and external applications are extending secure data and proprietary information out into a pool of expanded users in order to satisfy business needs. The dangers of this practice include other pool dwellers that might be malicious in nature, seeking to exploit what so many software professionals and system users hold dear – information.

The nature of this issue's focus, *Application Security*, reminds me of the story of Dame Ellen MacArthur, who completed the fastest solo non-stop circumnavigation of the globe in less than 72 days. MacArthur courageously sailed dangerous waters, surrounded by icebergs, massive swells, and sea-dwelling predators similar to the shark on this month's cover (but without the protective capabilities of the porcupine fish). Like MacArthur in her sailboat, making her way through a treacherous sea, precious organizational information flows through application software out to a community of users, vendors, and customers, with malicious hackers always present. This is risky business indeed.

Joe Jarzombek, Director for Software Assurance for the National Cyber Security Division of the Department of Homeland Security (DHS), while addressing an Advanced Software Acquisition Management class at the Defense Acquisition University, held up a copy of the June 2008 issue of CrossTalk on Software Quality as a useful source and said, "Rather than attempt to break or defeat network or systems security, hackers are opting to target application software to circumvent security controls." The numbers bear that out as Jarzombek noted that Gartner, Inc., an information technology research and advisory company, found that 90 percent of software attacks were aimed at the application layer. Jarzombek also believes that "most exploitable software vulnerabilities are attributable to non-secure coding practices and not identified in testing." He asserted, "Functional correctness must be exhibited even when software is subjected to abnormal and hostile conditions."

So what is a software professional to do? This month's articles provide an application security lifeline. The lead article, *Securing Legacy C Applications Using Dynamic Data Flow Analysis* by Steve Cook, Dr. Calvin Lin, and Walter Chang offers solutions in securing existing code within applications. *Building Secure Systems Using Model-Based Engineering and Architectural Models* by Dr. Jörgen Hansson, Dr. Peter H. Feiler, and John Morley, and *Practical Defense in Depth* by Michael Howard offer structural support to anyone looking to bolster their security practices on future designs. Likewise, Karen Mercedes Goertzel's article, *Enhancing the Software Development Life Cycle* [SDLC] *to Produce Secure Software*, shares the DHS Software Assurance Program's perspective of what makes software secure, while leaving specific SDLC security-development up to the reader. In *Supporting Safe Content-Inspection of Web Traffic*, Dr. Partha Pal and Michael Atighetchi show how Hypertext Transfer Protocol Secure (HTTPS) proxies enable safe interception and inspection of HTTPS traffic, while Corey P. Cunha's article *Hazardous Software Development* explores past safety-critical systems failures and modern software solutions. And finally, this issue's co-sponsor, the DHS Software Assurance Program, offers many free resources for Application Security – just follow the link on the back cover.

So, to all those charged with the difficult duty of securing your organizations applications, remember the promising success of Dame Ellen MacArthur. She, just like you, set sail into dangerous waters, yet she arrived safely and without a single scratch. To all our CrossTalk readers: I hope this issue helps in navigating through the sea of threats and leads to smooth sailing in all of your security efforts.

Kasey Thompson

Kasey Thompson
Publisher