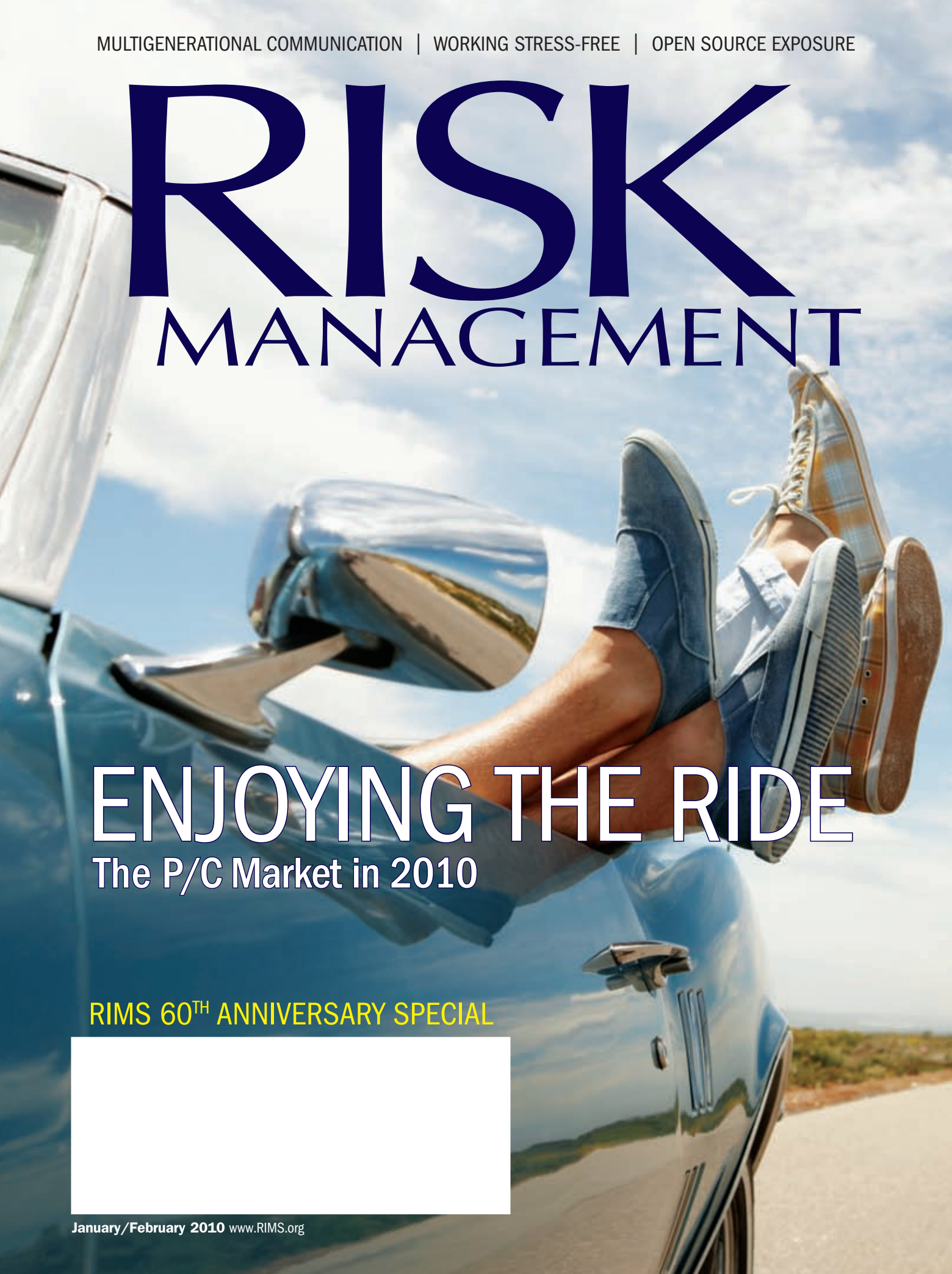
MULTIGENERATIONAL COMMUNICATION | WORKING STRESS-FREE | OPEN SOURCE EXPOSURE

RISK MANAGEMENT



ENJOYING THE RIDE

The P/C Market in 2010

RIMS 60TH ANNIVERSARY SPECIAL

January/February 2010 www.RIMS.org

Contents

32
A Multigenerational Perspective on Employee Communications

Employees of different generations have distinctly different communication needs in the workplace.

by Patricia Quinn

36
Open Source, Hidden Exposure

Using open source code in the development of new technology can jeopardize trade secrets.

by Brent A. Cossrow

42
Taking the Stress Out of Work

The workplace can be stressful, especially in tough times. But with certain steps, it does not have to be.

by James Nash

49
Time After Time

As RIMS celebrates its 60th anniversary, we look back at important milestones in the organization's history.

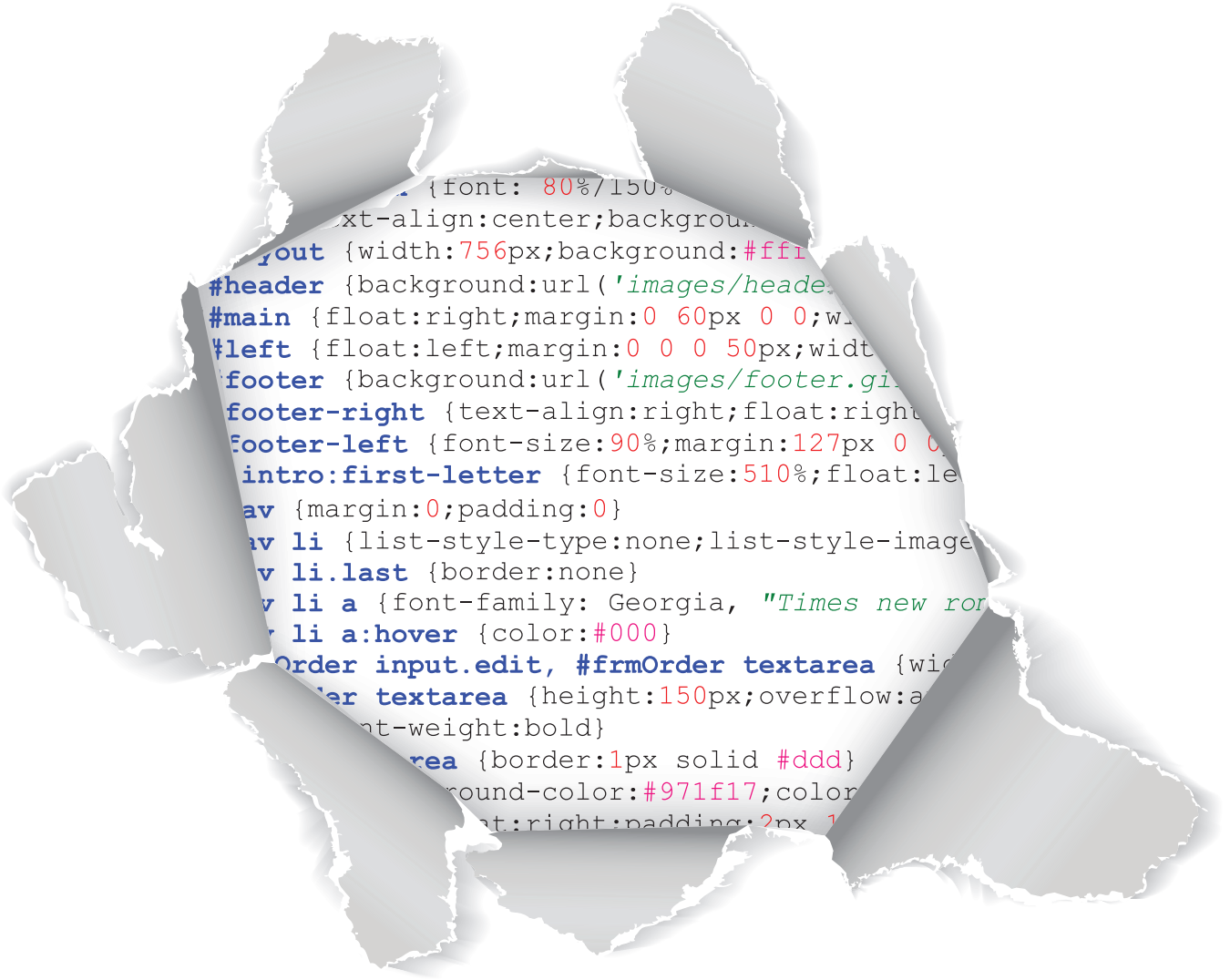
RISK MANAGEMENT

26
Cover Story
Enjoying the Ride

The financial crisis has made it difficult for businesses to make ends meet. But there is good news for insurance buyers who will be able to enjoy the low prices of a soft property/casualty market that shows every sign of continuing.

by Morgan O'Rourke

OPEN SOURCE



HIDDEN EXPOSURE

BY BRENT A. COSSROW

In an age when the ubiquity of computers has made the exchange of data effortless, a school of thought has emerged that suggests that "information wants to be free." Many computer programmers have taken this philosophy to heart with the creation and promotion of "open source code," computer programming language that by its very nature, is designed to be shared, at no cost and for no profit, by anyone who wants to use it. But what happens when open source aspirations come into conflict with the proprietary need to protect an organization's most vital trade secrets?

On July 3, 2009, computer programmer Sergey Aleynikov was arrested by the FBI and charged with stealing trade secrets belonging to his former employer, Goldman Sachs. According to prosecutors, in the days leading up to Aleynikov's last day of work at Goldman, he transferred, encrypted and uploaded 32 megabytes of data from Goldman's computer systems to an outside website. This data allegedly was the code for Goldman's proprietary high-speed trading program, which Aleynikov helped the firm develop. Prosecutors alleged that Aleynikov planned to use the code in order to help a new employer compete with Goldman.

Shortly after his arrest, Aleynikov claimed that he intended to only download "open source code." Although the charges against him may ultimately be dismissed, Aleynikov's claim that he was only after open source code residing on Goldman's computers marks one of the most recent appearances of the complicated issue of open source code's impact on a company's trade secrets. It also offers an opportunity for companies to better understand the hidden—and potentially serious—risks of their employees' use of open source code to do company business.

The Open Source Code Revolution

The context for understanding how an employee's use of open source code can impact a company's trade secrets begins with the unprecedented use of computers in the workplace. Today, ever-increasing numbers of employees are assigned individual work computers with internet and e-mail access. As employee use of computer systems has proliferated, so too has the use and movement of computer code, which consists of instructions given to a computer in order to make it perform tasks. There are two types of code used to run computers: the first is source code, which contains the instructions in a format that can be used, written and modified by employees. The second type of code is object code, which is written in binary code, a language using combinations of the numbers 0 and 1. Computers translate source code into the numerical object code, which then runs the computer.

As explained in a 2008 article by Donna Ruscitti and Jeremy Logsdon in the *Bloomberg Law Journal*, source code has been considered a trade secret "to gain a strategic advantage over competitors by not allowing competitors to see how the code was written." The authors explain that companies would license the right to use this code as part of software packages sold to the public or licensed to other users. By protecting the secrecy and proprietary nature of source code, companies profited from software licensing and other use fees, or simply gained a competitive advantage by retaining for themselves the efficiencies generated by the code they had created. This practice is widely used, and remains a profitable model for many companies.

This strategy and its rationale have come under attack by the open source software movement, however. Primarily through the internet, computer programmers now regularly make the source code of a wide range of useful programs available at no cost. As part of a philosophical shift, these creators no longer demand to be paid licensing fees for their work. This means that there now is freely available code that is capable of doing things that, in the past, you could only accomplish by paying for proprietary code. With open source, no individual or firm is supposed to possess classic proprietary control of the code—this is the essence of the "revolution" that open source code was intended to foster.

The hallmark of the open source revolution is the open source license that is supposed to govern open source code and its signature provision: the "copyleft" clause. Mocking "copyright," a copyleft clause typically requires the free and open redistribution of any modifications to original open source code.

As explained by the Free Software Foundation (FSF), which claims ownership of the phrase "copyleft" and one of the most widely used licenses, the GNU General Public License (GPL), copyleft is "designed to make sure that you have the freedom to distribute copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things." Copyleft verbiage varies, however. While the GNU GPL incorporates copyleft requirements, there are literally hundreds of other licenses now in use. Each license must be individually reviewed for its copyleft restrictions.

This radical challenge to the traditional profit model for computer code has considerable practical upside for businesses and their human resources managers. They can reduce software development and related costs by downloading open source code and customizing it for their (or their clients') needs without reinventing the wheel.

Professionals in the financial services, pharmaceuticals and accounting industries are increasingly using open source code in Microsoft Excel spreadsheets and other commercially available programs residing on an employees' work computer—employees use the open source code to modify

WITH OPEN SOURCE, NO INDIVIDUAL OR FIRM IS SUPPOSED TO POSSESS CLASSIC PROPRIETARY CONTROL OF THE CODE—THIS IS THE ESSENCE OF THE "REVOLUTION" THAT OPEN SOURCE CODE WAS INTENDED TO FOSTER.

or "soup up" commercially available programs to better suit their particular needs. Open source code provides employees with access to a range of digital tools and can be downloaded directly to their work computers in seconds.

Indelibly Diluting Trade Secret Protections

While "free" and "quick" are usually very attractive features for a business when constructing systems, companies must be aware of the risk before becoming too enamored of using open source code. Your company should consider the potential exposure from using open source code, particularly when it comes to your company's trade secrets. In order to qualify as a trade secret, your company's valuable information must not be generally known to or not easily ascertainable by others.

This definition would appear incompatible with open source code, which, by definition, is required to be publicly available. This also explains the potential explosiveness of Aleynikov's initial claim: if he downloaded open source code, was it stand-alone code or something integrated into Goldman's secret sauce? Whether open source code could tear down the walls your company has built to protect its trade secrets depends in large part on how your employees use the open source code and the terms of the controlling license.

Some guidance has been offered by FSF where its GPL license is involved, and companies can reflect on the scenarios set forth below (which incorporate some of FSF's commentary and can be found at *www.gnu.org*) as a starting point to investigate their employees' use of open source code.

For instance, if your employees download open source code and insert it into your company's proprietary software that is sold to the public, there is some risk your employees have infected your software, according to Ruscitti and Logsdon. Under the GPL, your company could be obligated to distribute all of your previously proprietary code into which the open source code was incorporated. This could have a devastating impact on the value of your company's proprietary assets, and inadvertently jeopardize trade secret protection. Even more disturbing, your employees can do this much damage in a few keystrokes without any notice to your company.

Open source code is not a communicable disease, however. Its mere presence on your company's servers will not deprive your company's prized assets stored in the same folder of their trade secret status. Thus, your employees can save non-integrated open source software on computer systems, digital storage devices and media without any exposure to other data.

Your employees' use of open source code for internal purposes (meaning exclusively in software that never leaves the company) should not jeopardize trade secrets. Your employees can download open source code and use it internally, without having to release the modified version of the code to the public.

Your employees can also use open source programs to help service your clients without exposure. For example, if an employee downloads open source code from the internet, then customizes the code in an internal program designed to analyze your client's quarterly sales, and shares the report with the client, your company should not have any exposure to trade secret dilution as long as your employee does not make the internal program containing the code available to the client or the public.

Open Source Code and Employee Defection

While the above scenarios provide some clarity, the issue of employee defection complicates matters. As the evolving Aleynikov case demonstrates, trade secret theft frequently occurs during an employee defection. For companies that failed to proactively monitor their employees' use of open source code, it could be too late: when these companies move to enforce their trade secret protections against a departing employee, the company might learn for the first time that the trade secret contained open source code added by an employee. This oversight could undermine the company's ability to enforce its trade secret protections, as evidenced by the 2004 case of *Computer Associates International v. Quest Software, Inc.*, which illustrates how fine the line can be between a use of open source code that dilutes a trade secret and a use that does not and how employee defection can confuse the issue.

Platinum International, a company acquired by Computer Associates in 1999, produced and marketed software known as Enterprise Database Administrator (EDBA). It allowed database administrators to automate and multi-task time-consuming work. To capture greater market share, Platinum modified EDBA so that it would be compatible with IBM's database software. In order to achieve this, however, Platinum had to first create a software tool known as a parser, which

WHETHER OPEN SOURCE CODE COULD TEAR DOWN THE WALLS YOUR COMPANY HAS BUILT TO PROTECT ITS TRADE SECRETS DEPENDS IN LARGE PART ON HOW YOUR EMPLOYEES USE THE OPEN SOURCE CODE AND THE TERMS OF THE CONTROLLING LICENSE.

they used open source code, called Bison, to do. This tool was critical to the court's analysis of the impact of the open source code on the trade secrets at issue.

The trouble started after Computer Associates purchased Platinum. Several Platinum employees who developed EDBA then resigned to join a third company, Quest, that hired Platinum's former employees to develop a product to compete with EDBA. The employees did not come to Quest empty-handed: tucked under their arms was the source code for EDBA.

When Quest released its product, Computer Associates received information that its source code was used by Platinum's former employees to develop Quest's competing software. Computer Associates filed a motion for a preliminary injunction seeking to enjoin Quest from using, selling, marketing, licensing and/or distributing its competing product. Computer Associates argued that Quest's code incorporated Computer Associates' protected trade secrets, which it rightfully acquired by purchasing Platinum.

In the portion of the litigation addressing whether EDBA's two million lines of source code qualified for trade secret protection, Quest raised the open source code issue. Responding to allegations that Quest misappropriated the source code for EDBA, Quest argued that portions of the EDBA code were derived from the Bison open source code, which was made available on the internet pursuant to GPL. This license, Quest argued, required that Platinum and Computer Associates freely distribute any modifications of the code. Accordingly, the portions of EDBA that were derived from Bison could not qualify for trade secret protection.

Quest's arguments were rejected by the court, and its analysis scrutinized how Platinum's former employees used Bison. The court determined that EDBA could remain a trade secret, even though the employees used the open source code in Bison. The court based this conclusion on two aspects of the employees' use of the open source code. First, the court observed that even if portions of EDBA's code were publicly available through Bison, the code's overall organization could remain a trade secret unless the organization of the code also had been disclosed. Neither Platinum nor Computer Associates disclosed the organization of the code, distributed the source code, or made it available to the public, so Computer Associates remained on the correct side of this line.

The court also observed that Platinum employees modified Bison to create the parser to produce the final EDBA product, but the parser was not the final product itself. Importantly, the employees used the parser only internally; the parser was not sold or marketed the way that EDBA was. This distinction is critical because Bison's license probably would have prevented Computer Associates from asserting rights in modified Bison open source code and the parser. For these reasons, the court determined that the entire EDBA code may qualify for trade secret protection, even though Platinum's employees used open source code in the process of creating EDBA.

Crossing the Line

Computer Associates and the unfolding Aleynikov case demonstrate the multitude of nuanced issues involved in employees' use of open source code. Rather than risk improperly—and unknowingly—crossing one of these fine lines, companies need to take a proactive, comprehensive approach to determining and monitoring how their employees are using open source code and analyze the controlling licenses. This process will likely require collaboration between counsel, risk management, product/business development, human resources, information technology and employee supervisors. These necessary steps might appear costly, but are insignificant compared to the potential damage that could result from the loss of trade secret protections. ■

Selected Open Source Products

Thousands of open source products and resources allow users and programmers to perform nearly every vital computing function. The following are a sampling of some of the most widely used and well-known open source products available.

LINUX. An operating system alternative to Windows or Mac OS that is perhaps the most famous open source product in the world.

APACHE. The most popular web server in the world since 1996. More than half of all websites use Apache and, in 2009, it became the first web server to be used by more than 100 million websites.

MYSQL. A database management system used by websites such as Facebook, Wikipedia, craigslist and YouTube.

FIREFOX. A web browser from Mozilla that is second after Internet Explorer in terms of worldwide users.

BIND. The most widely used Domain Name System server on the internet.

OPENOFFICE.ORG. An open source alternative to Microsoft Office that provides word processing, spreadsheet and other applications.

MEDIAWIKI. The software application behind Wikipedia.

WORDPRESS. A popular blog publishing platform used by such notable blogs as Anderson Cooper 360, TechCrunch, Perez Hilton and our very own Risk Management Monitor.

–Morgan O'Rourke