Winning the War Against Spam WITH JAVA

Mollom eliminates unwanted postings from Websites worldwide.

BY DAVID BAUM AND ED BAUM

Dries Buytaert, Mollom cocreator, at the company's Burlington, Massachusetts, office

PHOTOGRAPHY BY DAVE BRADLEY

Got spam? Who doesn't! But if you think finding and deleting a few hundred spam messages from your inbox or Website is a challenge, try deleting a few hundred million. That's the volume that leading spam-fighters such as Dries Buytaert battle each day. As cocreator of Mollom, a Java-based technology that processes hundreds of requests per second to keep the invading horde at bay, Buytaert is continually confronted with the "dark side" of the internet: it has become a petri dish of sorts, breeding countless unsolicited automated messages and comments like invasive bacteria. This infestation weakens worldwide productivity and threatens to compromise the way the Web works. The battle for the internet is raging, and it's us against the spambots.

These *spambots*—automated programs that assist in sending and posting spam—are getting smarter and more intrusive all the time. Thankfully, defensive strategies keep getting better too, as evidenced by Buytaert and his team. They recently migrated Mollom from home-

SNAPSHOT

MOLLOM
mollom.com

Headquarters:
Antwerp, Belgium

Industry:
Open source software

Employees:
4

Java version used:
Java EE 6

Buytaert drops in and answers questions from participants in a class at Mollom's Burlington, Massachusetts, office.

grown systems to GlassFish Server Open Source Edition—the open source version of Oracle GlassFish Server—to better manage the demands of keeping some of the world's busiest Websites free from spam.

"Java is a great fit for what we do," Buytaert says. "Because Java is part of a much larger technology ecosystem that includes lots of development tools and open source application servers like GlassFish Server Open Source Edition, it enables us to create a hosting service for thousands of customers with the back-end infrastructure that delivers the reliability, scalability, and availability of an enterprise-scale information system."

BEYOND PHISHING

Internet spam and the tactics used by spammers have become increasingly sophisticated since first appearing on internet bulletin board systems (BBSs) in the 1980s. Back then, spamming was as simple as repeating a word or phrase over and over to scroll other users' text off the screen. By the early 1990s, spam had evolved, and multiple repeat postings began to flood Usenet, where the first commercial spam messages appeared.

Today, most of us know spam as an abuse of our e-mail inboxes. And even though only a small percentage of people purchase the products (or fall prey to the scams) advertised through spam, these e-mail messages continue to proliferate because of their extremely low cost. If you manage a Website, you know how complex spamming strategies have become. Some spam messages on blogs and forums are targeted toward readers in an attempt at product marketing or *phishing* (trying to acquire information such as usernames, passwords, and credit card details). However, the spam that Website administrators battle most is a tactic deployed in an attempt to increase page rank on search engines such as Google.

"The more incoming links you have to your sites, the more relevance Google will give to those sites," explains Buytaert. "The spammers' goal is to abuse Websites' commenting systems to create links back to their own Websites in an effort to get the number one spot for certain keyword searches."

Buytaert says the strategies these spammers use are becoming so clever that it's often difficult to label the invasive content as spam. "They provide contextually relevant comments along with links back to their sites," he says. "The line between what's spam and what's not gets blurry. Identifying and blocking these types of abuses is becoming very difficult for Website owners and administrators."

ACCURATE AIM

Mollom's efficiency rate at identifying spam is 99.96 percent. Only 4 messages out of 10,000 succeed in fooling its Java programs.

The challenge of blocking this type of spam gets even greater when you consider its volume. Since Mollom's launch in 2008, the service has blocked more than half a billion spam messages. In fact 82 percent of all the comments on the sites Mollom monitors are spam.

Buytaert is a prominent figure in the Java community. In 2008 when he was at Belgium's Ghent

Buytaert and Mollom team members break for a quick lunch.

University defending his PhD thesis about Java performance and improvements, his defense committee included the father of the Java programming language, James Gosling, who was then senior vice president at Sun Microsystems, and Michael Hind, a senior IBM staff member for Java research.

Buytaert is best known as the founder and project lead of the Drupal project, an open source content management platform that powers about 2 percent of all the Websites on the internet. He has been using Drupal for his personal blog for many years, and as his blog became more popular over the years, spam comments proliferated. "I was wasting time manually removing unwanted content from the site," Buytaert says, "and I noticed a lot of other people in the Drupal community with the same problem. I saw an opportunity not only to improve my situation, but also to give back to the community. That led me to start working on Mollom."

TYPE THESE LETTERS, THEN DO A BACKFLIP

As kids, most of us probably never dreamed that we would someday be asked to prove to a computer that we're human. Yet that's what the ubiquitous CAPTCHA antispam solutions accomplish with their tedious challenge/response tests. The distorted backgrounds and warped text provide a problem easy enough for most humans to solve, while impeding most automated software. We dutifully squint at the twisted letters and numbers and retype them. Not only is it a nuisance to Website visitors; it's also discriminatory against visually impaired people. And some spambots are becoming smart enough to solve the CAPTCHA challenges by using advanced optical character recognition techniques.

Mollom's solution to spam is unique in that it intelligently combines three of the most effective defenses against unwanted automated comments while remaining mostly invisible to legitimate site visitors and comment posters. Mollom only delivers a CAPTCHA challenge as a last resort—for most cases, a CAPTCHA is never shown, which makes it easier for people to visit Mollom-protected Websites. Mollom achieves this feat by deploying other antispam techniques first. For example, it uses a clever trick based on JavaScript to hide form fields from human view. People don't see the form fields, but spambots assume the form fields are visible so they fill them out. When they do, their submissions are blocked. This technique is known as a *honeypot*.

Text analysis is another element of the Mollom solution. Like Defensio or Akismet (the latter being the default spam filter in WordPress blogs), Mollom examines the written content of a post to establish whether it's a legitimate submission or spam. The problem is, sometimes it's difficult to determine. When used alone, text analysis can wrongfully classify a legitimate post as spam, and block a genuine user. "Mollom recognizes that some posts can be difficult to categorize," Buytaert explains. "It's not always black or white, spam or not spam. So we have a third category: *unsure*. When Mollom is unsure if a post is spam or not, it serves a CAPTCHA."

Because it's a hosted software-as-a-service application, Mollom benefits from network effects. All the people using the system work together, providing feedback that helps the spam-identifying classifiers improve. In other words, Mollom continually gets smarter. "Using these network effects, we can even observe spambots as they assault multiple sites simultaneously,"

JAVA ON THE FRONT LINES

Java-based Mollom has blocked more than half a billion spam messages. Approximately 82 percent of comments on the sites Mollom monitors are spam.

Mollom: Under the Hood

The initial version of the back end for Mollom, a Web service that keeps invading hordes of spambots at bay, was based on a custom Java SE–based XML-RPC server. To allow increased focus on Mollom's core Machine Learning technology, Mollom co-creators Dries Buytaert and Benjamin Schrauwen decided to switch to a Java EE 6 back end, hosted on a GlassFish Server Open Source Edition 3.1.1 cluster. The Mollom team now extensively uses several core technologies such as JAX-RS with Jersey, the Java Persistence API, and Enterprise JavaBeans 3.1 (both singleton and stateless session beans). This has enabled the team to quickly develop an extended REST version of their API. For storage, Buytaert and Schrauwen chose a Cassandra cluster, because the back end is very write-intensive. Cassandra also allows the Mollom team to easily scale the storage layer across multiple data centers. Currently Mollom is running on dedicated machines, and to further reduce maintenance work, Buytaert and Schrauwen plan to move the infrastructure to Amazon's cloud environment, and to take advantage of the upcoming cloud features in GlassFish Server Open Source Edition.

Buytaert says. "We know that's not human behavior. It helps us identify and block these malicious automated programs."

By intelligently combining all these techniques and only using the CAPTCHA as a final test to weed out the small number of false positives and false negatives, Mollom provides the best-possible quality in terms of blocking spam while delivering the best-possible user experience for visitors to the sites it protects.

ENLISTING JAVA

Mollom is a real-time service, eliminating spam, filtering profanity, and safeguarding all kinds of sites—from small blogs that may only get one spam message per week to large social networks such as Europe's Netlog, with its 85 million registered users, and large media companies such as Sony Music Entertainment, which host hundreds of popular artists' sites. When someone makes a comment on a protected site, that site sends the comment to Mollom. Mollom analyzes the content and then returns a reply to the site. "It's important for us to have a low latency so we can process complex statistical analysis in milliseconds," Buytaert explains. "Visitors to Websites don't want to wait after clicking Submit to see if their comments were accepted. To accomplish this, we need a technology that can keep a lot of data in persistent memory so we can instantly classify content."

Buytaert clears his head by Mollom's "campfire," a place where staffers can brainstorm or just relax.

When Buytaert created Mollom with his partner, Benjamin Schrauwen, they initially built everything from scratch. But as Mollom grew to the point where it was handling millions of messages per day, Buytaert and Schrauwen found that they were spending too much time on infrastructure-related issues. That's when they switched to GlassFish Server Open Source Edition.

"We migrated to GlassFish because it let us worry less about memory management, XML parsing, database connection pooling, REST handling, and a lot of other factors that make big information systems work," Buytaert says. "It lets us scale that runtime environment more effectively. By migrating from our home-grown Java-based solution to GlassFish, we freed ourselves to focus more on the domain-specific challenges of Mollom. It's a great fit."

Buytaert says new users can get Mollom up and running within minutes. They simply create an account on mollom.com, install a plug-in, and enter public and private keys.

"The volume of spam comments we block is one measure of Mollom's success, but the quality of our service is equally important," Buytaert says. "Our efficiency rate is currently 99.96 percent. That means if we see 10,000 messages, we make four mistakes. We know we are helping to keep the Web a little bit cleaner. In business terms, that means Website owners can spend a lot less time worrying about the content that doesn't belong and more time creating valuable services for their user base. `</article>`

David Baum and **Ed Baum** are freelance writers specializing in innovative businesses, emerging technologies, and compelling lifestyles.