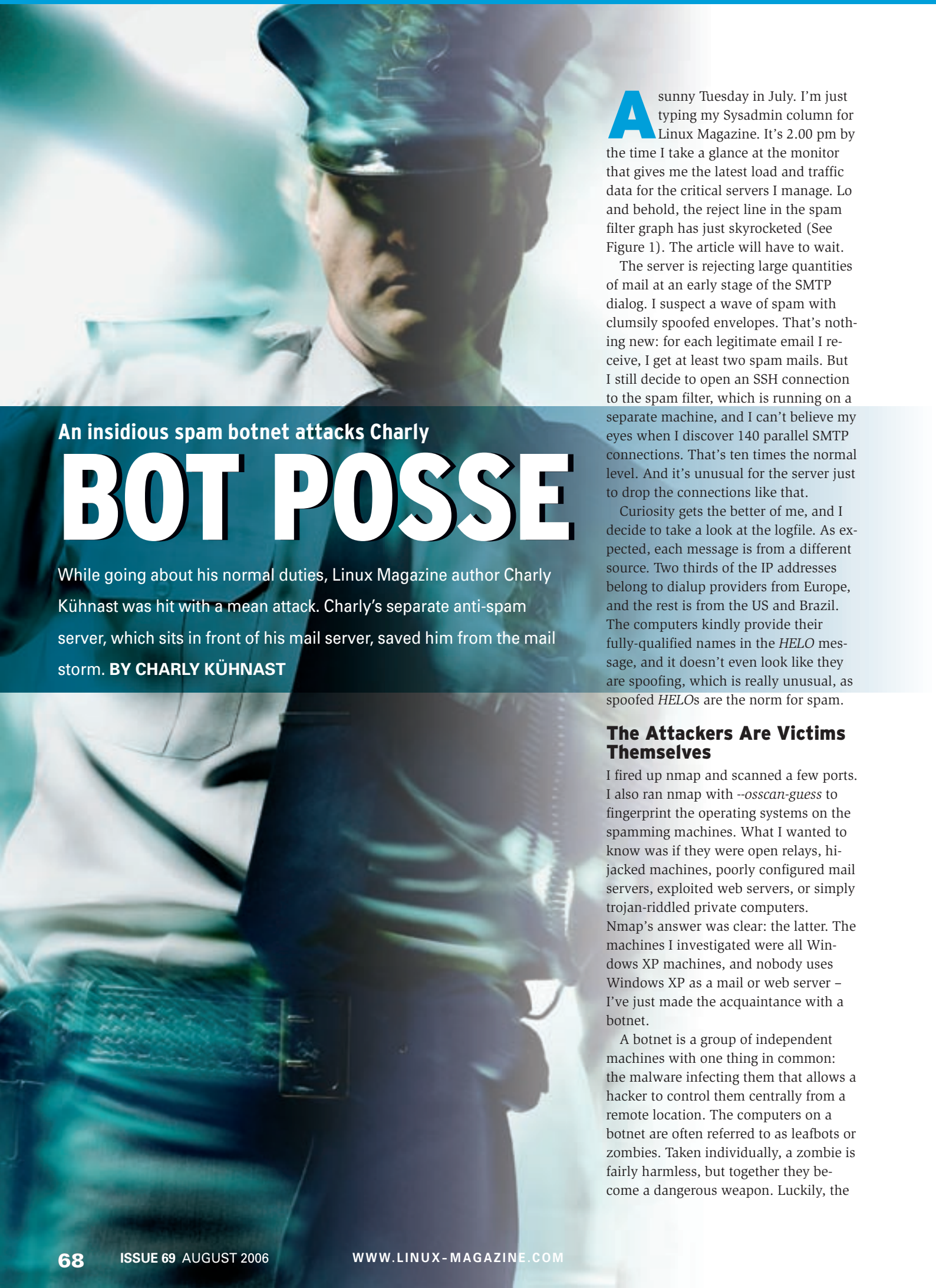

An insidious spam botnet attacks Charly

BOT POSSE

While going about his normal duties, Linux Magazine author Charly Kühnast was hit with a mean attack. Charly's separate anti-spam server, which sits in front of his mail server, saved him from the mail storm. **BY CHARLY KÜHNAST**

A sunny Tuesday in July. I'm just typing my Sysadmin column for Linux Magazine. It's 2.00 pm by the time I take a glance at the monitor that gives me the latest load and traffic data for the critical servers I manage. Lo and behold, the reject line in the spam filter graph has just skyrocketed (See Figure 1). The article will have to wait.

The server is rejecting large quantities of mail at an early stage of the SMTP dialog. I suspect a wave of spam with clumsily spoofed envelopes. That's nothing new: for each legitimate email I receive, I get at least two spam mails. But I still decide to open an SSH connection to the spam filter, which is running on a separate machine, and I can't believe my eyes when I discover 140 parallel SMTP connections. That's ten times the normal level. And it's unusual for the server just to drop the connections like that.

Curiosity gets the better of me, and I decide to take a look at the logfile. As expected, each message is from a different source. Two thirds of the IP addresses belong to dialup providers from Europe, and the rest is from the US and Brazil. The computers kindly provide their fully-qualified names in the *HELO* message, and it doesn't even look like they are spoofing, which is really unusual, as spoofed *HELO*s are the norm for spam.

The Attackers Are Victims Themselves

I fired up nmap and scanned a few ports. I also ran nmap with *--osscan-guess* to fingerprint the operating systems on the spamming machines. What I wanted to know was if they were open relays, hijacked machines, poorly configured mail servers, exploited web servers, or simply trojan-riddled private computers. Nmap's answer was clear: the latter. The machines I investigated were all Windows XP machines, and nobody uses Windows XP as a mail or web server – I've just made the acquaintance with a botnet.

A botnet is a group of independent machines with one thing in common: the malware infecting them that allows a hacker to control them centrally from a remote location. The computers on a botnet are often referred to as leafbots or zombies. Taken individually, a zombie is fairly harmless, but together they become a dangerous weapon. Luckily, the

botnet attacking me seemed to have just 200 machines.

A Word From Our Sponsor

I still haven't gotten around to finishing my Linux Magazine column, because the intervals at which spam is arriving seem to be getting shorter. Because I restricted the number of simultaneous SMTP processes, there is a danger of legitimate messages not getting through, due to a lack of resources to handle them. Let's take a look at the system load: the spam-filter still has capacity to spare. I remove the limit, and the botnet hits the accelerator: a few minutes later, I have 580 parallel SMTP connections (Figure 2).

Somehow these messages are getting past my graylister. Graylisting tells the server to reject a message with an error (*450 please try later*), and to accept the messages at the second attempt.

Background: Spambots don't normally have a queue to store undelivered messages temporarily before sending them off for a second try. It looks like graylisting is so widespread by now that spammers are starting to think about working around it. Listing 1 shows what effect this had on my site. Line 1 from the logfile shows how a zombie connects to my server. *greylist = update* in Line 2 shows the zombie having a second try, after my server rejecting the first attempt with a temporary error message.

Recipient Address Verification

Back to the log: the infinitely spoofable envelope sender is always the same, that is, < >, the null sender. The advantage of this sender address for the spammer is that any RFC-compliant mail server will accept it. And many anti-spam measures that rely on verifying the sender address, such as SAV (Sender Address Verification), are useless if you have a null sender.

The recipient addresses issued by the spam botnet are even more interesting: none of them exist, and all of them are words from an extinct Coptic dialect – or more likely from a random string generator. This explains why the spam filter drops the messages before the SMTP dialog gets to the *DATA* command phase. The filter performs Recipient Address Verification, the counterpart to SAV.

Recipient Address Verification is based on a simple principle: if the delivering server quotes the recipient address in *RCPT TO:*, the spam filter first checks where the mail has to go, that is, to my mail server (*mail.kuehnast.com* in Line 3 of Listing 1). It opens an SMTP dialog and checks the response to *RCPT TO:* . This is *user unknown* in the zombies' case. This causes the spam filter to terminate the dialog (Line 4), which is reflected in the masses of rejects in the graph.

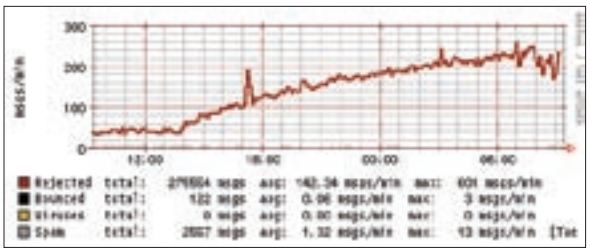

Figure 1: The reject line in the spam filter mail graph suddenly skyrockets – I've been attacked by an army of spam bots.

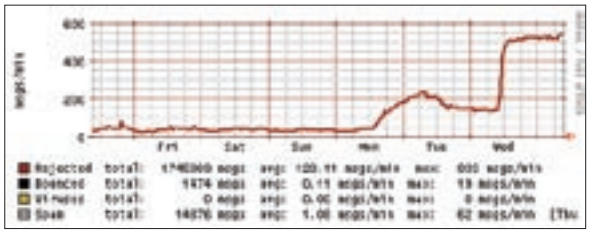

Figure 2: After removing the SMTP limit, the zombies up the attack against the server. loadavg shows 0.3 – and all is well.

It's starting to dawn on me why the spam filter is relatively unstressed, although it is handling way above 500 SMTP processes. None of these processes actually gets to deliver a message, Recipient Address Verification is blocking them well before that stage. If I had fed the messages to Spamassassin before verifying the recipient address, the spam filter would have bitten the dust by now due to the volume of incoming messages. My advice to mail admins: Recipient Address Verification makes life less eventful.

Interceptors

I decided to capture a few of the incoming spam messages, because I wanted to know what the message was that the bot army had been trying to hammer into my spam filter. I expected the usual gibberish plus a trojan disguised as a GIF or PDF. Or the usual adverts for erectile aids, party drugs, or breast enlargements (disappointingly, nobody seems to have a treatment for beer bellies). But what I discovered was just a jumble of ASCII. Either the spammer is just trying to annoy me, or they really don't understand MIME. I guessed the latter, and I was even considering running the alphabet soup through a Base64 decoder. But do I really need to know the price of Viagra this week? I think I'll just concentrate on the logfile, and watch the messages pearling off the spam filter. I can always finish the article later tonight. ■

Listing 1: Logfile Excerpt

```
01 May 12 04:16:07 spamfilter2 postfix/smtpd[32629]: connect from
   hcm-ms-185.vnn.vn[203.162.4.185]
02
03 May 12 04:16:07 spamfilter2 policyd: rcpt=598727, greylist=update,
   host=203.162.4.185 (hcm-ms-185.vnn.vn),
04 from=<>, to=shaynsimo@kuehnast.com, size=5228
05
06 May 12 04:16:07 spamfilter2 postfix/smtpd[29010]: NOQUEUE: reject:
   RCPT from hcm-ms-185.vnn.vn[203.162.4.185]:
07 550 <shaynsimo@kuehnast.com>: Recipient address rejected: unverified
   address: host mail.kuehnast.com[80.190.243.62] said:
08 550 <shaynsimo@kuehnast.com>:no such user (in reply to RCPT TO
   command);  from=<> to=<shaynsimo@kuehnast.com>
09 proto=ESMTP helo=<HCM-MS-185.vnn.vn>
10
11 May 12 04:16:07 spamfilter2 postfix/smtpd[32629]: disconnect from
   hcm-ms-185.vnn.vn[203.162.4.185]
```