;login:
THE MAGAZINE OF USENIX & SAGE
June 2003 • volume 28 • number 3

inside:

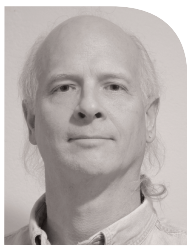
USENIX & SAGE
The Advanced Computing Systems Association &
The System Administrators Guild

musings

by Rik Farrow

Rik Farrow provides UNIX and Internet security consulting and training. He is the author of *UNIX System Security* and *System Administrator's Guide to System V*.

rik@spirit.com



Summertime, but is the living easy? As I write, the world is in turmoil, reeling from the effects of the past two US national elections. A very un-Republican interest in world affairs, coupled with increased repression in the "homeland" the neo-conservatives claim to be defending. Even the term "homeland" conjures up connections with another country which used the term "motherland."

For an example of repression, consider the Pennsylvania law that permits the attorney general to force worldwide Internet service providers to block access to a list of IP addresses. The intent of this law is lofty – to deny access to sites containing child pornography. But the side-effects include blocking access to any site co-hosted at the blocked IP address. The attorney general of Pennsylvania has refused to release this list of addresses, claiming this would be tantamount to providing access to child pornography. I wonder how this can be true if those addresses have been blocked?

Other non-American news sites have been blocked, whether from vigilante activity or perhaps some unannounced official interference. Democracy relies on freedom of speech, and blocking access to news sites that contain information that does not agree with a particular perspective is un-American.

And the economy? Let's not go there.

Instead, I'd like to write about Sendmail. Twice in March 2003, buffer overflows were revealed in Sendmail. It is hard to imagine that anyone reading this column wouldn't be aware of this, as most sysadmins (and those running their own UNIX/Linux systems) scrambled to get patches installed on all systems running Sendmail. With Sendmail running as root, and accessible through firewalls (or via internal relays, which would work just as well), there were lots of vulnerable systems around.

Connecting to Sendmail from the network can provide useful information:

```
220 spirit.com ESMTP Sendmail 8.12.8p1/8.12.10; [date removed]
```

Installing the patches provided by the Sendmail Consortium does update the version number and patch level received when you connect to port 25/tcp. The second part of the version information comes from whatever you have entered in the sendmail.cf file (or the macros that are used to construct it) to define the Z macro. I decided that my version of Sendmail would be a bit more advanced than most...

The first buffer overflow appeared in the crackaddr() function, which Sendmail uses to canonicalize addresses. The non-patched version was quite complex, and that was where the trouble lay. Each time a left angle-bracket was seen, a flag was set, and the number of bytes should have been decremented by one – but wasn't. When the right angle-bracket was encountered, the counter got incremented by one, making the overflow possible.

The Polish hacking group Last Stage of Delirium came up with an example exploit that would only work on Slackware 8. To be honest, a lot of us spent time trying to crash our versions of Sendmail just to see if they were vulnerable. Upgrading to Sendmail 8.12.8 meant (for many sites) upgrading to a new version of the config file as well, version 10. So the easiest path, replacing the Sendmail binary with the newest version, was not easy.

The LSD exploit hadn't been posted to their Web site (*http://lsd-pl.net*) but was posted to Bugtraq and archived there when I wrote this: (*http://archives.neohapsis.com/archives/bugtraq/2003-03/0054.html*). The exploit involves sending 138 pairs of <>s,

followed by a set of 0xf8s surrounded by parentheses (a pathological comment), and followed by the address to be overwritten. The exploit is not a simple one, as crackaddr() uses a statically defined buffer that gets stored on the heap instead of the stack. The consequences of this are twofold: first, that mechanisms that defend against stack-based exploits fail; and second, that any exploit writers have a much more difficult time.

The LSD programmers had an additional "concern" in that most Sendmail servers will be protected by firewalls that will only permit incoming connections to the server on port 25/tcp. Though this is a thoughtful idea, I doubt it is true. At any rate, their exploit could not be written to include a standard backdoor shell, by listening at a port and exec'ing a shell. Instead, their exploit connects back to the exploit program and executes uname -a. Most firewalls allow outgoing connections, and the LSD exploit actually assumes that a Sendmail server can reach port 25/tcp at arbitrary IP addresses (very reasonable).

The example exploit, or Proof of Concept (PoC) as such exploits are now euphemistically called, failed even to crash a RedHat 7.3 version of Sendmail that reportedly was vulnerable. Part of the problem is that heap exploits rely on the layout of memory allocated on the heap and finding the right set of bytes so that freeing a block of memory passes control of the program counter to the shell code. The LSD exploit passed their shell code as a very long line (about 2k) that gets sent right after the "Subject:" line, but with no intervening blank line (part of the normal message format defined in RFC 822).

The second buffer overflow, found by Michael Zalewski, involved a similar problem but in a different function, prescan(). Prescan() tokenizes addresses by looking for delimiters, and during this process a special value, 0xff, gets skipped, decrementing a counter. By providing specially formatted addresses, one could overflow the buffer used, pvpbuf[], which gets allocated on the stack. This buffer has a default size of MAXNAME plus MAXATOM, or 1256.

What these two exploits have in common is that they violate limits suggested in RFC 821 (*http://www.faqs.org/rfcs/rfc821.html*). If you read section 4.5.3, the suggested maximum size of an address is 256 characters, and of any line, 1000 characters. The LSD exploit sends an address that is about 300 characters long, along with a very long line containing the shell code. The vulnerability found by Zalewski appears to require an address in excess of 1256 characters, again outside the suggested limits. Note that the RFC does state that these limits are suggestions only, and that MTAs could be written that handle larger addresses and lines.

Still, sites employing application gateway (AG)-based firewalls, with the SMTP AG actually enabled, would have blocked both of these attacks without modification or updating. I actually contacted several firewall vendors (SecureComputing, Symantec, and Watchguard) and asked if their firewalls could block the LSD exploit. All three claimed that their AG firewalls (each vendor has multiple products) could block this attack if the AG were used. That's some good news, at least. One vendor, Symantec, also blocked by default access to WebDAV, another vulnerability (in IIS 5) announced in March, with the potential to be the next base for Code Red version 8.

I wonder how the patching wars have gone between the time I wrote this column and the time you will be reading it. Patching is never easy, but it is definitely easier if you have built an infrastructure for safely and reliably distributing and installing patches. I certainly wish all of you good luck, whatever the fortunes of war may bring.