;login:

THE MAGAZINE OF USENIX & SAGE

October 2001 • Volume 26 • Number 6

inside:

SECURITY

Musings

By Rik Farrow

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

I just got accused of being anti-open source. Because I had mentioned the BIND weakness being exploited by the Lion worm, I was suddenly the enemy. I know that I sometimes suffer from what I call "executive reading" – the ability to read plain text and totally misunderstand important sections of it. I invented this term in jest based on the responses I often get when exchanging email with busy people. I thought this was a disease that comes with age.

The executive-read seems to happen to me just when I am already on the edge of exploding, and I read something that appears to be the most dubious thing ever. Sometimes it is. And sometimes it makes better sense when I slow down enough to really comprehend what the author was saying.

I'd like to set aside the notion that I am against open source. I consider open source, including most of its variants, a wonderful idea. And, with that out of the way, I'd like to rave for a few paragraphs.

Linux systems have been the most commonly exploited UNIX platforms for many years now. And why is this? Is it because the open source community doesn't examine its own code? Or perhaps because the code wasn't examined before it was released, so that it could be exploited later? Maybe it is because the programmers are working at Internet speeds, and a couple of little problems slipped past them.

The real issue with security problems in Linux has little to do with any of these things. Certainly, better code review would help. The OpenBSD folk have made a serious effort at this, and Web defacement statistic sites like attrition.org and alldas.de reflect this, even though there are a LOT more Linux Web servers in the world than BSD-based ones.

The security problems we face today go beyond code review, however. What we face instead is a design crisis.

Out-of-the-Box

Just take any recent version of Linux and do a vanilla workstation install. And you know what? You have just become a server, and you might not even know it. You will have over a dozen listening TCP servers, so not only are you ready to rock and roll but you also have just opened your system up for potential attacks on all of these ports. And why?

Ease of use.

You can now perform DNS lookups without having to enter an IP address in /etc/resolve.conf. You can have email sent directly to your system (if an MX record already exists for your IP address, that is). The finger daemon is ready to reveal your login name, and the r commands are just waiting to serve. At least Linux doesn't come with an /etc/hosts file with a lonely plus sign in it, welcoming all who might drop by, the way Sun Microsystems did for so many years.

One of the simplest things that anyone can do to reduce the threat of network attacks is to disable unnecessary network services. Shut off all the services, and your only network footprint becomes the knee-jerk responsiveness of the IP stack to certain ICMP messages and broadcasts.

But, from the perspective of a network-based attacker, you have just become invulnerable. Nothing they can send you on the network will give them a shell prompt, execute a command, delete a file, or divulge any information other than the identity of your OS. Perhaps an attacker can convince you to do something dumb – social engineering is a powerful mechanism, as old as fraud – but without your unwitting assistance, you have configured the rock of Gibraltar.

If it is really that simple, why don't more people do this? Even better, why don't vendors do this for them? The answer is that vendors know that operating systems preconfigured to "just work" sell better than those that take a bit of tinkering to get them to do anything. And you can buy or download operating systems that are set up correctly already. There are Linux distributions configured out-of-the-box for better security. And, of course, there's OpenBSD.

Feature Quest

The quest for ever more interesting features has a much more enthusiastic participant than any open source group. I speak of Microsoft, which I will denote as MS.

While you can make a UNIX system invulnerable to network attacks pretty easily, the same is definitely untrue about MS systems. One of the saddest things I have to say when teaching security classes is that the easiest way to attack an MS box is with email. MS, in its quest for features, first loaded its browser, Internet Explorer, up to its gunwales with features, making the remote execution of code on a targeted machine child's play. Or should I say script-kiddie play?

Then, by linking IE to Outlook and Outlook Express, you can now send email and expect to have interesting things happen. Note that if you plan on updating every MS platform in your organization every three months or so (or whenever the next gaping hole is uncovered), you should be okay. Of course, everybody already does this, right? A better way of looking at this problem is to realize that if you are using a year-old version of IE, you should expect to have a Trojan installed on the system every couple of weeks. Fortunately, you do have anti-virus software installed to detect the Trojans that have now become endemic on MS platforms, don't you?

Once upon a time, MS platforms weren't considered interesting enough to bother hacking. Now, when an $800 PC comes with an 800MHz processor, scads of disk space, and may be connected full-time via cable modem or DSL, MS boxes have become a lot more interesting. Last time I checked, you could download over 100 different variations of MS Trojans (the source code, I mean, so you can create your own variation). A popular twist is to use private IRC channels for remote control. And the people controlling these channels typically instruct the Trojan to upload a new version every several days or so.

Wouldn't want to have an out-of-date remote control Trojan running on victims' systems.

MS XP

Speaking of remote control, MS has promised to "fix" the consumer marketplace. With MS XP, Windows NT finally comes to the consumer desktop, with an announced release date of October 25. Systems appearing in stores will no longer come with insecure Windows 98, but instead with a security-enhanced Windows XP Home Edition. Let's check out some of the features.

While you can make a UNIX system invulnerable to network attacks pretty easily, the same is definitely untrue about MS systems

. . . the reality of it is that it only takes a tiny group of people to take remote control of Internet connected systems designed with flexibility and features instead of reasonable security

You can get an idea of what is in store for MS users by visiting *http://www.microsoft.com/windowsxp/home/guide/dependable.asp*. Let's try a short quote:

"Using Remote Assistance, you can turn over control of your computer to a friend or technician who can solve your technical problems – without visiting your home. Once you give permission, the other person can control your computer remotely, over a network. . . . For extra security, you can also set a password that the recipient must use to connect to your computer."

I like that. It is as if MS has built BO2K or SubSeven right into Windows XP. I wonder if you can control the CD drawer too? And you can even set a password. I guess this is what MS meant by "enhanced security." Let's look at the next feature.

"Network Setup Wizard makes it easier than ever to set up your own home network so you can share printers, devices, files, and Internet connections among all the computers in your home." Sounds like a great idea, making all of that unused disk space available for remote use.

"System Restore: If you experience system failure or another significant problem, you can use the System Restore feature to roll back your computer to a previous state when it was working normally." Now this sounds really useful. The next time NTFS corrupts a critical file, or installing a game overwrites a key .DLL with an older version, you might actually recover without re-installing. This one feature alone sounds like a good reason to upgrade, as it will pay for itself in days.

There is even an "Internet firewall" included. Too bad it won't block email attachments, VB Script, HTML, XML, JavaScript, and the dozen other things that have proven dangerous for MS systems.

Lost

It is as if somehow the designers of operating systems got lost along the journey to the future. They believed it is all fun and games, and hey, we trust everybody! When the reality of it is that it only takes a tiny group of people to take remote control of Internet connected systems designed with flexibility and features instead of reasonable security.

Operating systems can be designed like ships, and I don't mean the Titanic. The notion of sandboxes, similar to the watertight compartments in Navy ships, would be a great addition to operating systems, especially if it included real hardware support to facilitate a strong implementation. Instead, we have the Titanic, a poorly designed ship, but boy, was it fast, and the accommodations (at least on the upper decks) were wonderful.

Back in 1982, when microprocessors were getting really cheap, I thought I saw the writing on the wall. I was working as a consultant at Morrow Designs, and they were designing disk controllers, and even serial port cards, with their own embedded processors. You build a system with distributed intelligence instead of having a single processor that has access to the entire system.

Let's try an analogy from Star Trek (whatever generation). There is always a method for self-destruct, to prevent the starship from falling into enemy hands. Enabling this self-destruct mechanism takes key phrases provided by the command staff (and always at least two members). Good security design, and quite appropriate given the seriousness of the occasion.

Now, if the Enterprise were designed like a modern operating system and its underlying hardware, self-destruct buttons would be sticking out of the walls, hanging from the ceiling, located next to the "Flush" button, and of course, right where your hand would reach to turn on the light in the middle of the night.

Our efforts to secure our existing systems resemble the crew of this sorry Enterprise going around putting big warning signs next to all the self-destruct buttons, as well as taping plastic cups over the ones little kids might press just for the heck of it (there are always a few kids on Federation warships it seems).

WARNING!! DO NOT PRESS THIS BUTTON!! YOU HAVE BEEN WARNED!!

Okay, I do sound a little cynical. I really shouldn't be complaining, as I don't have a ready solution for the problem.

Perhaps it is time to put on my Picard hat again, and make it so.

In the meantime, please remember to disable all unnecessary network services on every UNIX/Linux system under your control. If you are running a public Web server, set it up so it is ONLY a public Web server, and not a DNS server, POP server, IMAP server, FTP server, print server, rsh server (AARGH!), and so on. PCs are cheap, even if electricity is precious, so dedicate a system as a public Web server. And beware those self-destruct buttons.

In the meantime, please remember to disable all unnecessary network services on every UNIX/Linux system under your control