dæmon news
BSD's gone global

I've been hacked! How OpenBSD saved our project.

by John Horn
jhorn1@desperate.ci.tucson.az.us

I am one of the Unix SAs in the Information Technology department at the City of Tucson municipal government in Tucson, Arizona. Almost two years ago, our City Library approached us with a resource utilization problem. They had a number of Wyse 60 terminals installed in the various library branches which were a hold over from earlier days when these terminals were a primary means of electronic access to library catalogs.

In recent years, PCs have largely replaced these terminals as a means of accessing online library catalog information. The PCs are also used for library patron web surfing. However, there are always more patrons wishing to browse the Internet than there are PCs to accommodate them. Consequently, the library department requested us to provide a means for these patrons to use the Wyse terminals to browse the Internet. Access to these terminals would be unrestricted. Also we were informed, anyone already on the Internet from any location would be provided access to our solution without restriction via the library's DEC Alpha server.

Looking back on this today, it is clear this should have sent up bright red flags all over the place - but like many busy SAs might have done, we just lumped it in with our many other ongoing projects.

In a week or so we located a temporarily unused, small Sun server, installed Solaris 2.5.1 and gcc. We obtained Lynx, compiled it and set up the server to communicate with the DEC Alpha and its library application. Taking basic precautions, we installed Gene Spafford's tripwire and Dan Farmer's COPS. In the rush of this and other projects, we neglected to relocate the server, it was at this point still behind our firewall.

Our library modified their online menu to include our cobbled together Lynx server and they were off and running, as were we - on to other projects.

The next day, both tripwire and COPS began complaining. Initially, we took little notice of their reports; we were all involved in meetings and constant work on other projects. The reports produced by tripwire and COPS became increasingly bizarre and finally, after a week or so I logged into the Lynx server to see what was what. Well, there wasn't much wrong really, aside from the fact that some of our lynx users were engaged in a race to see who could compromise more of the system than the other. I think there were at least three different and independent folks who had already broken the root account. One was busily compiling and determining the relative merits of various free sniffer packages. Another apparently wished to set a record for the number of different back doors that could be placed on a Solaris system. Both of these had selected different odd corners of the file system to store their code. Another (at least I think it was yet another one), just wanted to experiment with different settings of things like /etc/system etc. One of these (or yet another one altogether) had decided that Sun's /bin/login just wouldn't do at all and had replaced it with one of his or her's own choosing. Frankly I thought Sun's was better - it didn't record all the passwords entered in /usr/local/lib/guffaws.a. Now don't get me wrong reader - I'm as much in favor of the volunteer spirit as the next guy. Heck, I volunteer for things myself from time to time, but I get paid to administer these systems and I really hadn't asked for any volunteer system administrators to help us out. Well, at least not formally. To some volunteers, a breakable system is request enough.

Well, I guess it would be safe to say I panicked slightly. I raced downstairs to the machine room and hit the panic button, also labeled on/off for those not familiar with panic buttons. Next, I collected the few wits I had responsible for getting me into this in the first place and went to my boss. Luckily, he was very understanding. He also wanted to know what I would do to correct the situation.

Here I will digress slightly. Fortunately, I had attended the 7th USENIX security symposium in San Antonio, Texas in January that year. On the last evening of the conference, after packing in preparation for the next day's flight, I meandered down to the Marriott lounge for a beer. By chance, I sat near a couple of the OpenBSD folks and we eventually became engaged in conversation. Most specifically, I recalled that they had been running a contest during a part of the conference in which they were offering a large amount of beer to anyone who could break the root account on their system. Being a pragmatist of Irish descent and thus knowing the value of a good beer as well as a secure system, I had watched this contest for some time but had not seen anyone walk off with a stack of beer. Well, after rather little effort, one of these OpenBSD folks, a fellow named Theo de Raadt, talked me into buying a CD set right in the lounge of the Marriott Hotel. I distinctly recall him telling me it is stable and secure. Theo and his companions soon left the lounge and I looked at the jewel case. 'Stable and Secure' it said. Uh-huh, I recall thinking, and ordered a second beer.

I recalled this as I sat in my boss' office and that I had loaned the CD set to a coworker. He had installed it and was quite satisfied with OpenBSD on his desktop. At this point though, I had still not used OpenBSD, though I had used other BSD systems.

I wasn't completely ignorant though. I had been to www.openbsd.org several times when I had a few spare minutes to try to learn just what I had purchased for $30.00 on a whim. According to the website, a default install would be absolutely secure. The site claimed that every line of source code had been audited for security. Now, I'm not an ace programmer but I've been coding off and on for a long time and I know that the only way to make any program secure is to audit every line of code and also, the smaller the program is the easier it is to audit. Still, there are a lot of lines of code in almost any OS. Some of course, have a whopping lot more than others if you know what I mean. The BSDs aren't like that, but still it is a lot of code to audit.

On the other hand, given a choice between a surprisingly plump /bin/login and a wonderful library like /usr/local/lib/guffaws.a - which was rather lacking in object modules - and an OS which even made a serious attempt to be secure, I would take the serious attempt any day.

I proposed that we try OpenBSD, similarly configured, but that I would take additional precautions this time around. Sheepishly, I also suggested we move the machine outside the firewall.

This proposal was accepted. I scrounged around to find a suitable PC and finally located an old 486/66 and proceeded to do an Internet based install from a dd'd floppy disk bootup. This worked flawlessly, though, like some other OpenBSD newbies, I hosed up the swap partition and had to eventually bail out of the install. The install instructions are very clear and concise on this issue and I still hosed it. The second time through everything worked fine. I let the install run overnight and the next morning the system was ready. Next I sat down and wrote some 'C', Perl and shell code to create a ridiculously paranoid monitoring system that would track in detail virtually every command issued on that OpenBSD box. Every time someone issued an `ls` command I would get paged and emailed with the `ls` and every argument passed to it. The same for many dozens of other commands. All of this stuff was being logged as well.

OpenBSD veterans are no doubt rolling on the floor by now, but the reader should understand that being hacked isn't a pleasant feeling and I over reacted. I have been hacked before, though never nearly as badly as this. Also, though I sometimes attend security symposiums, I am not a security guru. I am just another run of the mill Unix SA. I attend security symposiums to try to keep at least a toe in the water on security related issues but security is not my primary job function. Like any Unix SA, I have some familiarity with security issues, but some familiarity does not make one a security expert.

Having spent a few moments on the preceding day reflecting on what we were doing with this public access server concept and what had happened to the Sun box in the last week or so, I felt I harbored no illusions about the future of this OpenBSD machine. Deep down, I doubted if it would survive a couple of weeks of incessant hacking attempts by a virtually unlimited crew of dedicated folks in desperate need of a life. After all, we were already handing them the keys to the system, an open account from which they could do anything anyone with an account on a system could do.

I deployed the server on our DMZ and hunkered down over my pager and Pine, awaiting the first breakin. A day passed and the breakin attempts began anew, more than I care to enumerate. I watched them all from Pine and on my pager. A week passed, a month, a year. No successful breakins. Now almost two years have passed and I have upgraded that server to a newer version of OpenBSD. No breakins. Countless attempts, but none have been successful. I long ago disabled the email and paging programs I had set up. They were unnecessary and superfluous, though you could never have convinced me of that the day I wrote them. Still, they provided much amusement as I watched folks do some of the darndest things to try to reach what must be one of the pinnacles of hackerdom, a root account on an OpenBSD box. I have absolutely no idea if that mystical height has ever been scaled. Probably it has somewhere, sometime, by somebody, but I haven't seen it. That system is still installed completely in the default configuration. I haven't ever changed a single thing except for adding ssh for admin use - no Tripwire, no COPS, no IDS. Sure, folks are still trying to break in, but I have a lot of other projects to work on. It has been up for 95 days (when I did the last upgrade) and I'll reboot it in a year or so when I get the time, or when I finally get around to upgrading to 2.5. Our other OpenBSD system has been up for 281 days because, well, I just haven't gotten around to upgrading it yet either.

I have no idea how many folks Mr. de Raadt has working with him on the OpenBSD project. Apparently though, it is enough to really audit every line of code in every binary with an eye to security. Of course even that isn't nearly enough. In the UNIX world I've grown up in there are SUID programs, SGID programs, directory issues, and on and on. I haven't the faintest idea how all this was addressed in such a way that a completely default install is absolutely secure. I am, however, convinced now that they somehow have addressed each and every one of these issues. Vendors, take note - you're being badly outclassed by the Open Source movement. The quality of OpenBSD software is as good or better than any I've ever seen, even in the much vaunted Open Source arena. If commercial OSes don't take a few pages from the Open Source book soon, many of them will be completely replaced by high quality operating systems like OpenBSD. I've used most of the popular commercial ones and there isn't one I would trust on my DMZ with a completely default install, particularly wide open to anyone who would like to use the box anonymously.

I'll close this article with one observation. When Theo de Raadt told me that evening in the Marriott Hotel in San Antonio Texas that OpenBSD is 'stable and secure', he really meant it. And, when I saw him again recently at the USENIX Annual Technical Conference in Monterey at the OpenBSD booth, I wrote out a small donation check. That is the least I could do for the unimaginable effort that must have gone into OpenBSD.

John Horn UNIX Systems Administrator City of Tucson, Tucson Arizona