Zack's Kernel News

Chronicler Zack Brown reports on the latest news, views, dilemmas, and developments within the Linux kernel community.

By Zack Brown

ZACK BROWN

The Linux kernel mailing list comprises the core of Linux development activities. Traffic volumes are immense, often reaching 10,000 messages in a week, and keeping up to date with the entire scope of development is a virtually impossible task for one person. One of the few brave souls to take on this task is **Zack Brown**.

The kernel.org Security Saga

The recent security breach on kernel.org is still being dealt with. The kernel.org servers themselves are back up and offering some of the old services in new, more secure forms. One such service is Git repository hosting, and we're starting to see a lot of folks bring their Git repositories back to kernel.org. Nicholas A. Bellinger recently announced that the lio-core Git tree has returned to kernel.org. Konrad Rzeszutek Wilk made a similar announcement about the Xen Two tree, as did Takashi Iwai about the sound Git tree, Chris Ball regarding the MMC tree, Theodore Y. Ts'o about the ext4 tree, and Roland Dreier about the InfiniBand tree.

One by one the disruptions caused by the attack are fading, but the protocols and procedures regarding secure kernel code submission and distribution have only begun to form and will undoubtedly continue to develop over the coming years. With his Linux 3.1 announcement, Linus Torvalds remarked, "I really want the pull request to be validated some way. With the small changes late in the `-rc` series, I could afford to spend the time to look at commits and try to verify them, but with the merge window (and the 11k commits or so that I saw pending in the last linux-next tree), that just isn't reasonable. So, use git.kernel.org or some other host that I can trust is really you."

Meanwhile, the new GPG "web of trust" continues to broaden. This is where one group of trusted developers meet up with other people who they know and trust and sign their public key; then, those people can sign other people's keys who they know and trust, and so on. Recently, Jonathan Cameron announced a key-signing event in Cambridge, England, in the Cambridge University engineering department; H. Peter Anvin announced one in Santa Clara, California. Others have been organized all over the world.

H. Peter also recently posted a checklist for developers to follow to restore their kernel.org accounts. The main requirement was to attend a key-signing event to get a "signing path" that led back to the kernel.org administrative team. Although, of course, as Phillip Lougher pointed out, kernel developers living in out-of-the-way places might not be able to find fellow kernel developers nearby who could sign their key. With no easy remedy for that situation, you'll just have to travel to a key-signing event and bring sufficient proof of identity with you.

Steven Rostedt announced a patch for the "quilt" system of patch submission, to allow users who were already part of the web of trust to sign their outgoing patches cryptographically with their GPG key. He remarked in his announcement, "After the attack of kernel.org, several kernel developers are getting paranoid about who is really who. A lot of focus is on signing emails that verify who people really are using GPG signatures." His patch addressed that concern, and several other developers pitched in with suggestions and technical considerations.

(Ironically the quilt tool, originally written by Andrew Morton to avoid having to use Git, is now maintained and developed in its own Git repository.) Recently Greg Kroah-Hartman posted some tips on how developers could check their own systems for security compromises. The thread turned out to include a number of useful posts: *https://lkml.org/lkml/2011/9/30/425*. In his announcement, Greg said, "The compromise of kernel.org and related machines has made it clear that some developers, at least, have had their systems penetrated. As we seek to secure our infrastructure, it is imperative that nobody falls victim to the belief that it cannot happen to them."

The security issues surrounding the kernel.org break-in and tainting of one of the Linux `-rc` releases is reminiscent of the infamous SCO lawsuit that went from 2003 to 2007. In it, SCO claimed that Linux violated Unix source code copyright and demanded that all Linux users around the world purchase a license from SC, if they wanted to continue to use Linux.

That lawsuit was eventually defeated, but it led to the implementation of the "signed-off-by" protocol and related mechanisms for tagging kernel patches to create a clear path of developers validating that a given piece of code came either from a free source or from the mind of a given developer and did not violate the copyright of any other code.

These signed-off-by rules set up by Linus went through their own process of development over time, and although fairly stable by the current time, they remain, like all of Linux, a work in progress. Undoubtedly, the key-signing process and other security measures being implemented these days will continue to evolve and probably lead to simple protocols that newcomers will find easily manageable. In the meantime, the immediate inconveniences continue.

New Suspend Daemon

Neil Brown has written a suspend daemon to provide a simple interface to control when and how a given system will suspend to disk. It uses files in `/var/run/suspend/` to allow all running software to communicate with the daemon and suspend gracefully – or prevent suspending altogether.

For example, if any process holds a shared file lock on the file named `disabled`, the system will refuse to suspend.

Not surprisingly, one of the first comments had to do with the location of those control files and, really, whether the interface should be file-based at all. With the proliferation of `/proc`, `/sysfs`, `ioctls`, system calls, and other attempts to create a clean interface between userspace and the kernel, it's never really obvious which interface to use.

A number of technical issues cropped up over the course of the discussion. Part of the problem was that Neil's code had been intended as a proof of concept, but to make it really robust would involve adding various features and handling various cases. Thus, the discussion went off in a number of different directions. At one point, Rafael J. Wysocki admonished, "Well, you're now considering doing more and more changes to the kernel just to be able to implement something in user space to avoid making some _other_ changes to the kernel. That doesn't sound right to me."

So, various technical issues and constraints came up, but in general, no one seemed to take the position that a suspend daemon was a bad idea; I'd imagine something resembling Neil's tool will get into the kernel eventually.

VirtualBox Tainting

Dave Jones had some harsh things to say about VirtualBox. He said, "The number of bug reports we get from people with VirtualBox loaded are truly astonishing. It's GPL, but sadly that doesn't mean it's good. Nearly all of these bugs look like random corruption (corrupt linked lists, corrupt page tables, and just plain 'weird' crashes)."

He posted a patch to taint kernels that had VirtualBox loaded, in a similar way that kernels using features from the `staging` directory are tainted. This way, automatic bug-reporting tools can opt out of filing bugs for kernels that use those features that are known to be problematic. Of Dave's four-line patch, one of the lines was the single comment `/* vbox is garbage. */`.

Greg Kroah-Hartman liked the patch so much he said he'd add it to the openSUSE kernels, which got lots of bug reports because of just that driver. He also added, "we should have a list of these types of modules, as I think there are a few others out there we should mark this way."

The discussion got somewhat intense. Some folks felt it would be best to lock down kernels that included anything compiled from out-of-tree sources, but because that idea had holes in it, one suggestion was to require GPG signatures to ensure that people couldn't work around the restrictions.

At one point, Alan Cox remarked, "If you want to get into a sophisticated fight with someone over hiding the presence of a module then that's a pointless exercise. If you want to just make it easier to sort and detect out-of-tree modules then fine, but the only actual pressure you have controlling its effectiveness is going to be the embarrassment of a vendor who gets caught out. GPG is thus I think over-engineering it somewhat." During the whole discussion, Frank Mehnert, the VirtualBox maintainer, said:

"I can understand that you would rather ignore bug reports from external kernel modules. On the other hand, I don't like the `TAINT_CRAP` flag as you can probably imagine. Why not just mark external modules like Bastian Blank suggested? I can assure that we will not try [to] circumvent a `TAINT_OOT_MODULE` flag.

"Please also note that we always have good relations to the open source community so feel free to point me [to] an archive where all these kernel panic reports arrive which you've got. We fixed some bugs in our kernel modules in the past, and it is even possible that some of the current bug reports are from older versions of VirtualBox which might have been fixed in the meantime." ■■■