Zack's Kernel News

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

By Zack Brown

This edition of Zack's Kernel News is dedicated to David Brownell, a kernel contributor who inspired and encouraged many other hackers to work on free software. Rest in peace, David. May your name linger long in the source tree.

Easy Virtualization

Pekka Enberg announced a native Linux KVM tool, to make it easy to boot virtualized guest images on your Linux box and log into them without having to perform a lot of housekeeping chores.

In his announcement, Pekka said, "The goal of this tool is to provide a clean, from-scratch, lightweight KVM host tool implementation that can boot Linux guest images (just a hobby, won't be big and professional like QEMU)."

Steven Rostedt caught the irony, referring to a comp.os.minix post from 1991 that read, "I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu)."

Pekka's announcement was met with general enthusiasm, along with the customary references to similar projects also in the works, and recommendations of better naming possibilities than "native Linux KVM tool."

In the course of discussion, Pekka added that a graphical user interface was also being worked on and that, in theory, the tool could some day support generic guests (i.e., other operating systems); but, Pekka said he personally was only interested in supporting Linux.

Reliable Message Passing for Embedded Systems

Tony Ibbs announced KBUS, a kernel-based messaging system for use primarily between applications running under phones and other embedded systems. The main goal of KBUS is to provide a reliable message-passing infrastructure, in which messages always arrive successfully and in a predictable order.

Jonathan Corbet took a look at the code and asked why a kernel module was necessary for this sort of thing. Why not a userspace daemon instead? Tony replied that the main reason to take a kernel module approach was reliability. Specifically, he said, if a process using KBUS crashes before it can give an expected reply to a message, the kernel module would be able to detect that and send a synthetic reply to the other waiting process. Trying to do this in userspace, he said, would be much less reliable.

Jonathan had other technical comments, as did James Chapman. Their general response to Tony's announcement was a sense that KBUS wasn't really necessary and had some interface problems. But, the discussion didn't continue long enough to clear things up.

Securing the Heap

Dan Rosenberg wanted to change the `/proc/slabinfo` file permissions to `0400`, which would make the file readable only by its owner and not by regular users. This change, Dan said, would make it harder for hostile attackers to take advantage of kernel bugs that produce heap corruption.

Dan remarked that although this would make it impossible for regular users to debug a running kernel, an admin could give users that ability by manually changing the `/proc/slabinfo` permissions for them.

Dave Hansen didn't like this approach. He said it would make systems less secure by encouraging people to do debugging operations as root instead of as a regular user. He suggested that if any systems needed Dan's suggested level of security for `/proc/slabinfo`, the admin for that system could change the permissions to `0400` manually.

Dan argued that the vast number of kernel users would never do any kernel debugging, so the default should be to protect those users as much as possible.

Matt Mackall entered the debate at this point, after having examined a number of heap exploits to determine how effective Dan's solution would be in practice. His conclusion was that anyone attempting a heap exploit would not need to rely on accessing `/proc/slabinfo` in all cases. True, he said, access to `/proc/slabinfo` made things slightly easier for the attacker, but there were alternative approaches that didn't require any access to it.

Dan replied that, yes, his patch would only increase security by a relatively small amount, but it would effectively increase the costs that hostile attackers would have to incur to produce an exploit. By making the exploit more expensive, Dan hoped to discourage some percentage of those attacks from ever occurring. He said, "the primary goal of exploit mitigation isn't necessarily to completely prevent the possibility of exploitation (time has shown that this is unlikely to

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**.

be feasible) but, rather, to increase the cost of investment required to develop a reliable exploit."

Matt wasn't convinced that Dan's patch would significantly raise the cost of producing an exploit, but he said he was on the fence about it anyway and would be fine with Dan's approach.

But Theodore Ts'o stepped in then to tell this story: "Being able to monitor `/proc/slabinfo` is incredibly useful for finding various kernel problems. We can see if some part of the kernel is out of balance, and we can also find memory leaks. I once saved a school system's Linux deployment because their systems were crashing once a week, and becoming progressively more unreliable before they crashed, and the school board was about to pull the plug.

"Turned out the 'virus scanner' was a piece of garbage that slowly leaked memory over time, and since it was proprietary code that was loaded as a kernel module, it showed up in `/proc/slabinfo`. If it had been protected, it would have been much harder for me to get access to such debugging data."

Ted suggested that there might be a way to modify the slab allocator itself to improve security, without having to make `/proc/slabinfo` less accessible.

Linus Torvalds seemed to think Dan had at the very least identified a problem worth addressing. He suggested perhaps modifying `/proc/slabinfo` to expose slightly different information than it currently does – information that would still be useful to the user, but not so much to an attacker.

At this point, the discussion became more technical, with actual exploits and attack vectors considered and more kernel-hackey patches submitted for consideration. With Linus's endorsement, a variety of approaches toward improved heap protection likely will be implemented and tried out at least, and some will probably make it into the kernel.

Copyright Violation in the Source Tree

David Johnston noticed some of his code in the kernel sources that he had not given permission to include there. He posted to the linux-kernel mailing list, saying, "my header with my name and copyright has been removed. A different copyright has been added, and it has been licensed under the GPL without my knowledge."

He added, "I am a happy user of Linux; I don't want to cause trouble and I'd be quite honored to have some of my code in the kernel, so I'm not demanding the immediate removal of this code. I'm willing to GPL my code if necessary, but I do require proper attribution and acknowledgment of my copyright on my code."

Theodore Ts'o, H. Peter Anvin, and Greg Kroah-Hartman discussed the matter. Apparently, the same company that submitted the code in the first place might also have been submitting it elsewhere, so Ted recommended that David contact that company directly to get their portion of the problem straightened out.

Regarding the kernel source tree itself, some discussion arose as to whether the code should be removed immediately while the copyright issue was figured out, and Greg also pointed out that the original code had been released to the "public domain," without any specific license. So, he said it was at least reasonable for whoever took the code to assume they had the right to do so.

Because David was happy to have his code in the kernel, the issue was resolved very quickly and easily. He formally released the code under the GPL version 2, and that was that. He also said he'd written to the company that had originally used his code, and was working to straighten it out with them, too.. ■■■