Zack's Kernel News

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

By Zack Brown

Response to Breach on kernel.org Servers

Recently, the kernel.org servers were cracked by attackers who were able to gain root-level access. The attackers then inserted trojan horses into the source releases for certain Linux kernel release candidates (`-rc` releases).

This attack caused a lot of work for the kernel.org system administrators and resulted in a number of discussion threads on the linux-kernel mailing list, considering ways to avoid similar security compromises in the future.

In one thread, Junio C Hamano, the Git maintainer, asked the kernel folks if there were any special Git features they wanted, that might increase the security of a Git archive that involved many contributors (e.g., the Linux kernel). He suggested providing the ability to cryptographically sign all pushes, as well as having Git produce more output on certain types of failure modes. Linus Torvalds replied, saying he liked the idea of increased verbosity; but, about cryptographic signatures, he said:

"I realize that cryptographic signatures sound very important right now, but in the end, *real* trust comes from people, not from signatures. Realistically, I checked a few signatures this time around due to the kernel.org issues, but at the same time, the thing that made me trust most of it was just looking at commits and the email messages. The unconscious and non-cryptographic 'signature' of a person acting like you expect a person to act.

"Technical measures can be subverted, and I think we should also think about the social side. Every time somebody mentions a signature, I want to also mention 'human readability', because I think that matters as much, if not more."

A certain level of cryptographic credentialing can't be avoided, however, and H. Peter Anvin has posted documentation explaining how to re-establish the GPG "web of trust," enabling people to modify Git repositories on kernel.org again. The idea is that establishing a web of trust would make it more difficult for a hostile attacker to gain that level of trust, which would potentially eliminate the "social networking" vector of a potential attack.

The web of trust is centered around developers who need access to kernel.org to update Git trees. Regular developers who submit patches primarily via email are not being asked to generate GPG keys or travel to key signing events, or anything like that. Anyone who's been emailing patches without cryptographic signatures may continue to do so, according to Theodore Y. Ts'o. But, Ts'o also says, if developers do have a GPG key and the ability to participate in a key signing event, the benefit for you would be an added level of trust that, in fact, the kernel repositories you download from kernel.org are uncompromised.

Aside from setting up cryptographic keys and improving the readability of Git output, the kernel.org system administrators have been rebuilding their servers and attempting to eliminate any attack vectors at that end. One important element of that will be restricting access to kernel.org itself. From now on, H. Peter announced, developers maintaining Git repositories on kernel.org would no longer have shell access to the system. If an attacker could crack a single user's password and gain shell access, they'd have a relatively easy time gaining root privileges again. In his announcement, H. Peter said that kernel.org would henceforth use the "gitolite" tool to allow repository maintainers to access only their repositories, and not the underlying system.

H. Peter also said that other services that had previously been accessible to users with kernel.org credentials would be brought back over time, as soon as the admins could figure out how to implement them securely.

It's interesting to note that by attacking the kernel source trees on kernel.org, the attacker was attempting to compromise the security not just of some Linux users, but of all Linux users, everywhere. To me, it's reminiscent of some of the cyber attacks seen recently between individual nations of the world.

Ugly Fix Still Best for Binary Breakage

In other news, Andi Kleen is continuing to peddle his patch to make 3.0 kernels masquerade as 2.6 kernels, so that binary-only software expecting to run on 2.6 systems won't break for no good reason. He said that people were running into this problem with "all kinds of software."

Apparently having the kernel pretend to be an earlier version is the only way to get those binary-only applications to run. As Andi has said, it's an ugly solution. But he thinks it

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**.

needs to be implemented in the official kernel source – not just for a short time, but as a permanent feature of the kernel. As he said, "binary compatibility is important. It's one of the things that made Linux successful."

Linus Torvalds replied to the thread, saying he didn't want to include this patch without a good reason. He pointed out that Andi had been reluctant to say exactly which applications were breaking. Linus said, "I'm not at all interested in these kinds of 'all kinds of software' reports. Details. Examples. Name the f&*cking names already. Shame them publicly."

Eric Dumazet said that various HP management software would fail to detect controllers correctly unless the system pretended to be running a 2.6 kernel. And Andi said that his original motivation for the patch was to handle the "pintool" tool (though that tool had subsequently been fixed).

Colin Walters pointed out that the Python programming language would report the OS as "linux3" instead of "linux2," which wasn't a bug in itself but would cause any script to break that did an improper check on that string. Andi searched on *http://www.google.com/codesearch#/* for the improper code and found many cases, although they are not binary-only applications, being coded in Python.

Pavel Machek pointed out that the kernel's own "ketchup" tool was one of the tools broken by the transition to the 3.0 version number. The ketchup tool is used to switch between versions of the Linux kernel. As Pavel said to Linus, "no, it is probably not what you wanted, and no, it is not easy to fix." Stratos Psomadakis also pointed out the ketchup problem, but Andi reported that his patch wouldn't fix ketchup, because ketchup would try (and fail) to download the nonexistent 2.6 kernel that the 3.0 kernel would masquerade as. However, Stratos added that ketchup was being updated to handle 3.0 and later kernels.

Linus didn't reply to the thread again; it's unclear whether Andi's patch will ever make it into the source tree. The thought of accepting that patch probably makes Linus want to retch into his hand, but there may be no alternative if Linux is to support those older binary-only applications.

New SLIMbus Driver

Kenneth Heitke posted a patch originally from Sagar Dharia implementing the SLIMbus specification for a two-wire cable used to communicate with audio devices and other peripherals. SLIMbus is intended to replace a variety of other buses and provide a single solution for a range of devices. Kenneth's and Sagar's code implemented a number of software APIs for message-passing along the wires and for distinct data channels.

Arnd Bergmann liked Kenneth's and Sagar's work, thought it was in the right spot in the source tree, and had good documentation, but the device registration method seemed a bit outdated. He suggested they update the code to match current practices, but it seems he had no objection to the code ultimately being accepted into the kernel.

Mark Brown also expressed interest in the patch, asked to be copied on future updates, and offered his own set of technical suggestions. Kenneth said he'd incorporate both Mark's and Arnd's comments into the next revision.

Russell King dissented, saying that this driver seemed to be doing the same thing as the SPI and I2C code, and that at a recent conference, he'd heard of a third bus type also addressing the same problems. He suggested consolidating the four projects to avoid multiple solutions to the same problems. Jean Delvare remarked, "The similarities are certainly due to the fact that SPI and I2C were designed by the same person (David Brownell), and SLIMbus most probably was originally cloned from either." Kenneth agreed with Russell's objection and said he was open to suggestions. The rest of the discussion involved technical implementation details for Kenneth's and Sagar's code, and no one seemed to worry much about merging SLIMbus with other projects. It's likely the SLIMbus code will be accepted at some point, although it seems some of the technical issues still need to be hashed out before that can happen.