LWN's Latest: An OpenSUSE 'Foundation', Security, Programming and Kernel (Linux)

By *Roy Schestowitz*
Created *04/07/2019 - 5:30pm*
Submitted by Roy Schestowitz on Thursday 4th of July 2019 05:30:10 PM Filed under Development [1] Linux [2] SUSE [3]

- **An openSUSE foundation proposal [4]**

> The idea of spinning openSUSE out into a foundation is not new; it has come up multiple times along the way. The most recent push started back in April at two separate board meetings where it was discussed. It picked up steam during a board meeting at the openSUSE Conference 2019 in late May. While waiting for the outcome from that meeting (though there was a panel session with the board [YouTube] at the conference where some of the thinking was discussed), the community discussed ideas for a name for the foundation (and, possibly, the project itself). Now, board member Simon Lees has posted a draft of the foundation proposal for review.
>
> The proposal outlines the current thinking of the board. It notes that the move to a foundation is not meant to pull away from SUSE, "but to add more capabilities to the openSUSE Project". In particular, having a separate entity will allow the project to "receive and provide sponsorships (in terms of money, hardware, or contracted services)". Currently, any kind of agreement between the project and some other organization has to be done via SUSE, which can complicate those efforts. The new foundation would be able to partner with others, receive donations, spend money, and sign contracts with venues, service providers, and the like, all on behalf of the openSUSE project.

SUSE would clearly have a role in the new foundation; the board is requesting some funding to set up the organization as well as one or two people to help with the administrative side. The new foundation's board would take the place of the existing project board, with the same election rules as there are today (which results in a board of six, five elected from the members of the project and the chair appointed by SUSE).

The board is looking at setting up a German stiftung foundation as the legal entity for the new organization, though that was not clearly specified in the draft proposal. An eingetragener Verein (e. V.) was considered, but the structure of that type of entity is inflexible; in addition, the purpose of an e. V. can be changed if there was a "hostile takeover" at some point. Umbrella organizations (e.g. the Linux Foundation) and simply keeping things the same were also looked at, but were deemed unworkable for various reasons.

There is also a handful of open questions, including logistical issues such as whether SUSE or the new foundation would own the IT infrastructure, trademarks, and so on. Also, who would be responsible (in a GDPR sense) for the project's data collection and storage. The biggest open issue is to create a charter for the foundation, which requires legal advice. The Document Foundation (TDF) is something of a model for what openSUSE is trying to achieve; it is also a stiftung and shares some of the attributes with the proposed structure.

- **CVE-less vulnerabilities [5]**

More bugs in free software are being found these days, which is good for many reasons, but there are some possible downsides to that as well. In addition, projects like OSS-Fuzz are finding lots of bugs in an automated fashion?many of which may be security relevant. The sheer number of bugs being reported is overwhelming many (most?) free-software projects, which simply do not have enough eyeballs to fix, or even triage, many of the reports they receive. A discussion about that is currently playing out on the oss-security mailing list.

- **C, Fortran, and single-character strings [6]**

The calling interfaces between programming languages are, by their nature, ripe for misunderstandings; different languages can have subtly different ideas of how data should be passed around. Such misunderstandings often have the effect of making things break right away; these are quickly fixed. Others can persist for years or even decades before jumping out of the shadows and making things fail. A problem of the latter variety recently turned up in how some C programs are passing strings to Fortran subroutines, with unpleasant effects on widely used packages like LAPACK.

The C language famously does not worry much about the length of strings, which simply extend until the null byte at the end. Fortran, though, likes to know the sizes of the strings it is dealing with. When strings are passed as arguments to functions or subroutines, the GCC

Fortran argument-passing conventions state that the length of each string is to be appended to the list of arguments.

- **Statistics from the 5.2 kernel ? and before** **[7]**

As of this writing, just over 13,600 non-merge changesets have been pulled into the mainline repository for the 5.2 development cycle. The time has come, once again, for a look at where that work came from and who supported it. There are some unique aspects to 5.2 that have thrown off some of the usual numbers.
1,716 developers contributed changes for the 5.2 kernel, 245 of whom made their first contribution during this cycle. Those 1,716 developers removed nearly 490,000 lines of code, which is a lot, but the addition of 596,000 new lines of code means that the kernel still grew by 106,000 lines.

- **Lockdown as a security module** **[8]**

Technologies like UEFI secure boot are intended to guarantee that a locked-down system is running the software intended by its owner (for a definition of "owner" as "whoever holds the signing key recognized by the firmware"). That guarantee is hard to uphold, though, if a program run on the system in question is able to modify the running kernel somehow. Thus, proponents of secure-boot technologies have been trying for years to provide the ability to lock down many types of kernel functionality on secure systems. The latest attempt posted by Matthew Garrett, at an eyebrow-raising version 34, tries to address previous concerns by putting lockdown under the control of a Linux security module (LSM).
The lockdown patches have a long and controversial history; LWN first wrote about them in 2012. Opposition has come at all kinds of levels; some developers see lockdown as a way of taking control of systems away from their owners, while others see it as ultimately useless security theater. There does appear to be some value, though, in making a system as resistant to compromise as possible, so these patches have persisted and are often shipped by distributors. Disagreement over more recent versions of the lockdown patch set were focused on details like whether lockdown should be tied to the presence of secure boot or integration with the integrity-measurement infrastructure.

One outcome from the most recent discussion was a concern that the lockdown patches were wiring too much policy into the kernel itself. The kernel has long had a mechanism for pushing security-policy decisions out to user space ? the security-module mechanism. So it arguably makes sense to move lockdown decision-making into an LSM; that is indeed what the more recent versions of the patch set do.

First, though, there is the problem of initialization. LSMs exist to apply policies to actions taken by user space, so as long as the LSM infrastructure is running by the time user space starts, everything is fine. Lockdown, though, must act earlier: it needs to be able to block the action of certain types of command-line parameters and must be functional even before a security policy can be loaded. So the patch set starts by creating a new type of "early security module" that is initialized toward the beginning of the boot process. At this point, the module can't do much ? even basic amenities like kmalloc() are not available ? but it's enough to register its hooks and take control.

Development Linux SUSE

Source URL: http://www.tuxmachines.org/node/125542

Links:
[1] http://www.tuxmachines.org/taxonomy/term/145
[2] http://www.tuxmachines.org/taxonomy/term/63
[3] http://www.tuxmachines.org/taxonomy/term/117
[4] https://lwn.net/Articles/792053/
[5] https://lwn.net/Articles/791855/
[6] https://lwn.net/Articles/791393/
[7] https://lwn.net/Articles/791606/
[8] https://lwn.net/Articles/791863/