Zack’s Kernel News

Full freedom

For awhile, Linux has provided certain abilities to modules based on their license. If they set a certain variable to indicate that they are GPLed, then they are given greater freedom within the kernel. Binary-only, or otherwise non-free modules, are not given this access.

The existence of the ‘License variable’ is controversial, as are the legality or illegality of binary-only modules; however, some proprietary vendors have managed to bypass the license variable, either willfully or inadvertently, in a very interesting way.

LinuxAnt is one such vendor, using the following string in their license variable: “GPL\0for files in the \“GPL\” directory; for others, only LICENSE file applies”. Note the ‘\0’ after the initial ‘GPL’. Since \0 signifies the end of a string in C, this causes the kernel to only see the ‘GPL’, and to discard the rest of the string.

The effect is that LinuxAnt’s driver is given full access to the kernel, in spite of the fact that not all of it’s files are free. Some developers do believe LinuxAnt added the ‘\0’ intentionally to bypass the check, but regardless, in future versions Linux will no longer be susceptible to that kind of back door. Still, it is interesting to watch the development of this ‘feature’, and to see what potential exploits against it turn up.

Counting the cost

The ELSA (Enhanced Linux System Accounting) project has begun as an attempt to extend the existing process accounting mechanisms. One of the ways ELSA will address this problem will be by clustering processes together into ‘banks’ of related processes, that could be tallied up and analyzed as a group. In theory, other things besides processes could also be included in these banks, making it possible to draw sophisticated conclusions from the data gathered.

The problem of tracking the use of system resources is one that is crucial to operating system design. While most users don’t have much need of tracking their own behavior, the statistics of the behavior of many users is quite valuable in the creation of a number of algorithms in the kernel. The virtual memory subsystem is an example of a kernel feature that can benefit from system accounting.

The questions of which aspects of a running system belong in RAM or swap, and when the transitions should be made, as well as what to do when those resources grow scarce are difficult to answer. This is because they are dependent on how a particular system is used. Appropriate behavior on one system could be wrong on another.

One more feature

Marcelo Tosatti’s attempt to close the 2.4 tree off from further additional features is proceeding slowly, as he struggles to balance the needs of users with the needs of Linux development to stabilize the 2.4 and 2.6 trees and move on to new developments in 2.7. Most recently he accepted Serial ATA (SATA) into the kernel, a bold and somewhat controversial move, considering that 2.6 will also support SATA.

The argument for inclusion is that much new hardware has SATA by default. The 2.6 tree has not yet taken over the user-base is due to the various radical changes that took place during the 2.5 development cycle, including new module code, new tools for compilation, and other features that just make it harder to just drop in the 2.6 kernel as a replacement for a working 2.4 kernel.

Marcelo has begun to repeat himself, each time saying, “this is the last new feature to go into 2.4”.

Marcelo’s task is difficult, because he must balance the need to keep the 2.4 tree reliable as the primary Linux kernel, while accepting some features he regards as essential, and rejecting other features that for one reason or another are not so essential.

INFO

The Kernel Mailing List comprises the core of Linux development activities. Traffic volumes are immense 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 that take on this impossible task is Zack Brown.

Our regular monthly column keeps you up to date on the latest discussions and decisions, selected and summarized by Zack. Zack has been publishing a weekly digest, the Kernel Traffic Mailing List for several years now, reading just the digest is a time consuming task.

Linux Magazine now provides you with the quintessence of Linux Kernel activities straight from the horse’s mouth.

Hiding data

Guillaume Lacôte has started work on a way to obfuscate data at the device-mapper level, so that potential attackers would have a hard time interpreting any data available to them. This involves a compression algorithm that doesn’t actually need to compress anything.

Compression tends to make data appear random, because patterns are identified and removed as part of the process. Guillaume also plans to interleave random data.

He hopes that an attacker would give up rather than struggle to identify the random data. There’s some support for an idea of this kind, although it is unclear that the level of obfuscation can be obtained using his methods.

Modules made easy

I’m always happy to see documentation crop up in the kernel world, and recently Sam Ravnborg wrote a HOWTO on building external modules for a kernel that has already been compiled. The kbuild configuration system that was so controversial in the 2.5 series, provides a number of features to make this easier.

For building external modules, a user must have a compiled kernel plus the full source; after which he lists a batch of ‘make’ commands to accomplish tasks, such as to build or install modules, or to clean object files out of the directories.