

Automatic device driver isolation protects against bugs in operating systems

August 22 2022, by Mary Fetzter



Credit: Pixabay/CC0 Public Domain

An operating system's kernel acts as the translator between user and machine. To improve the reliability of a kernel, developers can isolate the operating system's device drivers and prevent a failure in one component from affecting other components. Isolation, however, requires impractical amounts of human effort.

A team of researchers, led by G. Gary Tan and Trent Jaeger, professors of computer science and engineering at Penn State, set out to develop a framework that could automate and reduce the amount of manual work needed for device driver isolation in the presence of challenging kernel patterns.

The researchers presented their framework at the [16th USENIX Symposium on Operating Systems Design and Implementation](#) (OSDI '22), which took place July 11–13 in Carlsbad, California. OSDI is a premier conference in operating systems research and brings together professionals from academic and industrial backgrounds to discuss the design, implementation and implications of systems software, according to its website.

The operating system kernel controls and coordinates all hardware and software in the computer. Device drivers allow the kernel to interact with hardware without knowing the component's details. For example, when a user directs their computer to print a document, the kernel invokes certain interface functions provided by a printer driver, which processes the data and sends the job to the printer.

According to Tan, to effectively isolate device drivers and maintain kernel-driver communication, developers needed to inspect the large and complex communication interface between a driver and the kernel and decide what data needed to be synchronized by examining all the interactions between the driver and the kernel. They also needed to handle challenging synchronization patterns such as data concurrency, writing thousands of lines of code to keep operations smooth.

"Isolation is an effective technique for improving reliability in [software systems](#), such as the kernel, but relying on human effort to isolate drivers is unrealistic, so we set out to develop a framework to automate the process," Tan said. "With isolation, failure in one component is

constrained within its own domain; bugs in one component cannot directly affect the rest of the system. This significantly improves reliability."

They developed KSplit, an approach that can both automatically analyze the shared driver-kernel state and compute any of the synchronization needs for just this shared state. The program also identifies areas that need manual intervention, reducing the manual workload to minimal coding and data marshaling.

"Before KSplit, the process was labor-intensive and prone to error," Tan said. "With KSplit, most of these complicated isolation tasks are handled automatically."

The researchers evaluated the solution with 354 device drivers on nine subsystems in a Linux kernel. For one complex driver, KSplit required manual updates to 53 of 2,476 lines of code, which were automatically generated interface specifications, and 19 additional changes to the driver's code. That's less than 3% of the manual work required without KSplit.

"The KSplit analysis of the 354 drivers shows a similar fraction of manual work is expected, demonstrating that KSplit is a practical tool for automating key tasks to enable driver isolation," Jaeger said.

"However, the framework mainly focuses on improving system reliability. We plan to extend KSplit to harden the interface between the kernel and a driver, with a security guarantee for the [kernel](#) even in the presence of a malicious driver."

According to Jaeger, the set of static analyses provided by KSplit, such as computing necessary data among different system parts, can be extended to software that requires a precise analysis of data access information, highlighting the potential for protecting against intentional

[security breaches](#) or other attacks.

Provided by Pennsylvania State University

Citation: Automatic device driver isolation protects against bugs in operating systems (2022, August 22) retrieved 4 May 2024 from <https://techxplore.com/news/2022-08-automatic-device-driver-isolation-bugs.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.