Contribution ID: **138** Type: **not specified**

Revocable: a mechanism for preventing ”classic” use-after-free bugs

A “classic” Use-After-Free (UAF) can occur when resources tied to hot-pluggable devices are accessed after the device has been removed. For example, an open file descriptor may hold references to such resources; if the device is unplugged, subsequent file operations on that descriptor can trigger an UAF. This talk, a follow-up to a previous presentation[1], explores an approach to this challenge.

We will present “revocable”[2], a new kernel mechanism for resource management. A revocable allows a resource provider (e.g., a device driver) to invalidate access to a resource from a consumer (e.g., a character device) when the underlying device is no longer available. Once a resource is revoked, any further attempts to use it will fail gracefully, thus preventing the UAF.

We will discuss the design and implementation of the revocable mechanism and its application in the ChromeOS Embedded Controller drivers to fix a real-world UAF bug. We hope to also start a discussion on how this generic mechanism could be adopted by other drivers to handle similar resource lifecycle issues.

[1] https://lpc.events/event/17/contributions/1627/
[2] https://lore.kernel.org/chrome-platform/20250820081645.847919-1-tzungbi@kernel.org/T/#u

Primary author: SHIH, Tzung-Bi

Presenter: SHIH, Tzung-Bi

Session Classification: Kernel Summit Track

Track Classification: Kernel Summit Track