

Confusion about process states and invariants

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2004-06/0112.html>

From: Robert Watson (rwatson_at_FreeBSD.org)

Date: 06/26/04

Date: Sat, 26 Jun 2004 12:38:43 -0400 (EDT)

To: arch@FreeBSD.org

Over the last two weeks, I've seen several reports of panics relating to code making incorrect assumptions about process state, generally relating to the "p_ucred" pointer in new and dying processes. In particular, a number of pieces of code assume that if a process is reachable by the all process list (or other process lists), p_ucred will be valid and non-NULL if the process lock is held on the process. This results in possible NULL pointer dereferences in the PRS_NEW state, and also during the tear-down in kern_wait(). At first glance, the easy answer would appear to be "check for p_ucred to be NULL", but I'm actually of the opinion that I'd prefer we have the non-NULL p_ucred invariant actually hold true. This would permit security checks to be performed properly during those windows. I'm not very familiar with our process state and locking, but if someone with a more qualified background in that area could comment on the current issue, that would be useful.

FYI, two of the reported problems were in sysctl_kern_proc() and linprocfs. One of those has been patched by checking p_ucred for NULL; the other has not.

Robert N M Watson FreeBSD Core Team, TrustedBSD Projects
robert@fledge.watson.org Principal Research Scientist, McAfee Research

freebsd-arch@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-arch>

To unsubscribe, send any mail to "freebsd-arch-unsubscribe@freebsd.org"