KASLR for Freescale BookE32

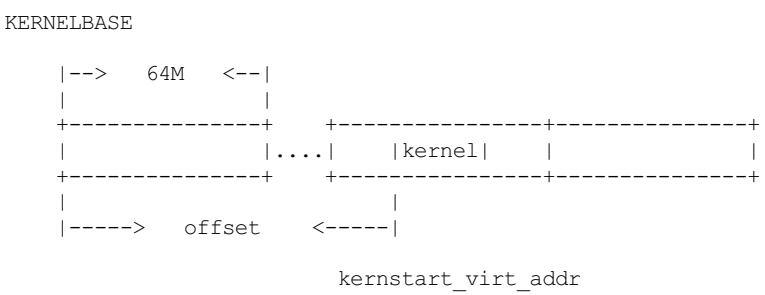
The word KASLR stands for Kernel Address Space Layout Randomization.

This document tries to explain the implementation of the KASLR for Freescale BookE32. KASLR is a security feature that deters exploit attempts relying on knowledge of the location of kernel internals.

Since CONFIG_RELOCATABLE has already supported, what we need to do is map or copy kernel to a proper place and relocate. Freescale Book-E parts expect lowmem to be mapped by fixed TLB entries(TLB1). The TLB1 entries are not suitable to map the kernel directly in a randomized region, so we chose to copy the kernel to a proper place and restart to relocate.

Entropy is derived from the banner and timer base, which will change every build and boot. This not so much safe so additionally the bootloader may pass entropy via the /chosen/kaslr-seed node in device tree.

We will use the first 512M of the low memory to randomize the kernel image. The memory will be split in 64M zones. We will use the lower 8 bit of the entropy to decide the index of the 64M zone. Then we chose a 16K aligned offset inside the 64M zone to put the kernel in:



To enable KASLR, set CONFIG_RANDOMIZE_BASE = y. If KASLR is enabled and you want to disable it at runtime, add "nokaslr" to the kernel cmdline.