Rethinking late kernel module patching

- **A module may or may not be loaded**
  - It must be patched on load and before being executed
- **Current solution**
  - No module dependencies
  - Hooks in kernel module loader so that a module is patched on load and unpatched on its removal
  - Arch-specific code (late relocations, alternatives, parainstructions, jump labels, …)
  - Fragile and unmaintainable in the long term

Rethinking late module patching

- **Introduce module dependencies**
  - It would load unneeded modules
  - “Half-load” them?
- **Per-object live patches**
  - Atomic replace infrastructure changes could cause troubles
  - Per-object consistency?
- **“Blue sky” idea**
  - Livepatch only loaded modules
  - Replace .ko on disk
  - Blacklist vulnerable versions
  - Complexity moved elsewhere?