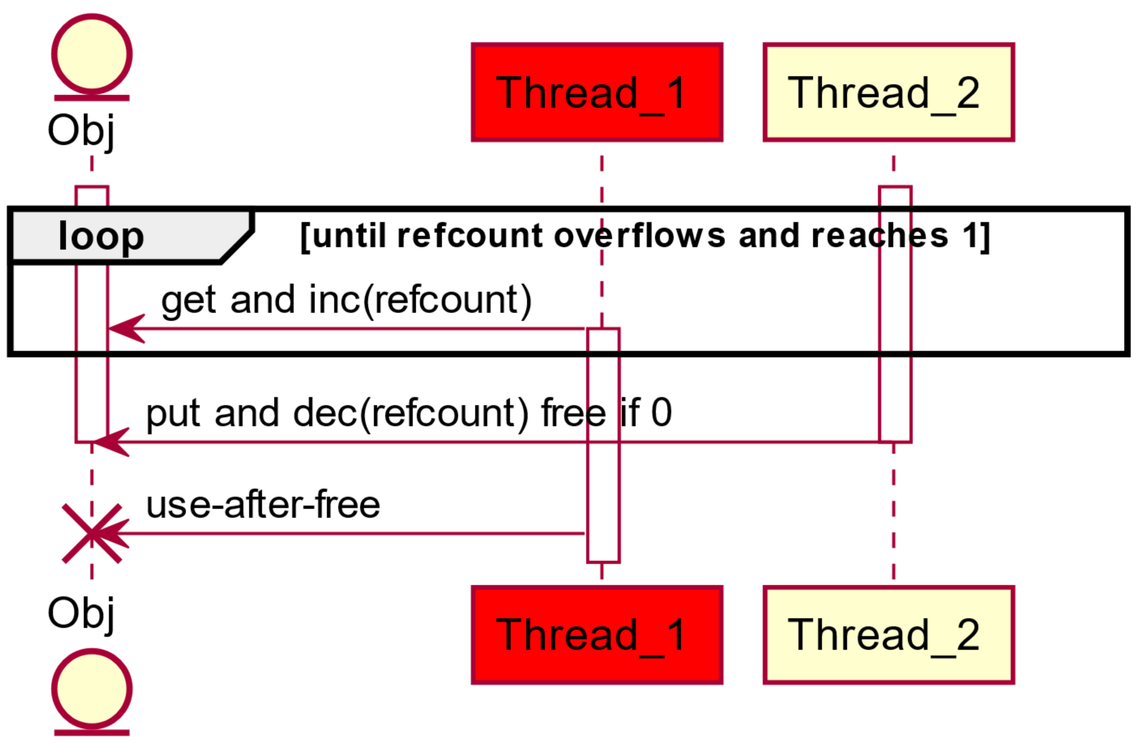
Kernel Hardening
Secure Systems Group, Aalto University

Hans Liljestrand, Elena Reshetova, N. Asokan

Linux Kernel Memory Safety

Reference Counters in the kernel

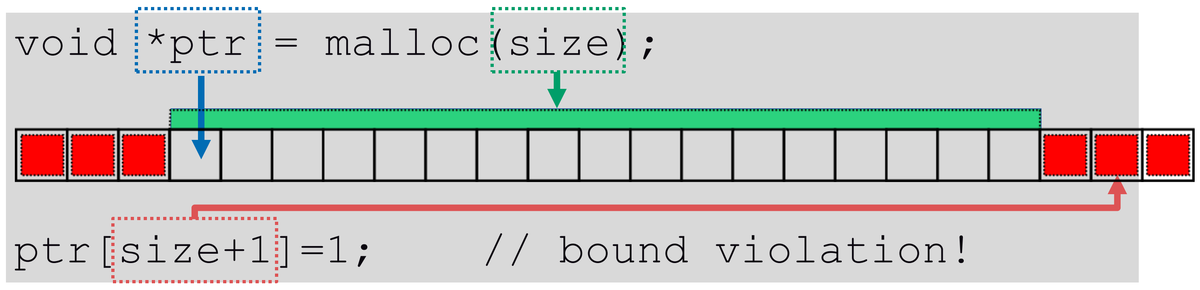
- The kernel does not have high-level objects and garbage collection; It uses counters to track object use and ensure safe destruction.
- Refcounts are implemented as atomic integers (e.g. atomic_t) and can thus overflow causing faulty deletion and use-after-free.



Preventing Refcount Overflows

- We prevent overflows by saturating the reference counter before overflow, thereby reducing the use-after-free to a memory leak.
- Proposed PaX/Grsecurity-based solution of hardening underlying atomic types:
  - Secure-by-default.
  - High maintenance overhead.
  - x86 race-condition in protection.
- Contributed to eventual refcount_t design:
  - Restricted API promotes safe use.
  - Generic implementation, still changing.
- Working on kernel-wide adoption:
  - 233 patches submitted by us.
  - ~60 patches landed upstream.

Bounds checking with Intel MPX



- The Intel Memory Protection Extension (MPX) is a runtime pointer bounds checking mechanism.
- MPX adds new instructions and registers, but needs software support by compiler and OS.
- No source code changes needed to enable, and mixing with non-MPX code is supported.

Challenges for in-kernel MPX

- Hardware designed for ring 0 and 3, but current software supports only user space.
- Excessive memory use.
- User space MPX relies on Page Faults for on-demand memory allocation.

MPX-in-kernel implementation

- Eliminates memory requirements by using kernel memory management metadata for pointer bounds.
- Implemented as in-kernel support code and GCC-plugin for kernel-specific instrumentation.
- Currently working proof of concept with support for modular use via file-specific flags.

Aalto University
Collaborative Research Institute for Secure Computing