Sentry kernel key concepts

A SECURE KERNEL FOR MICRO-CONTROLERS

PART 2: USER-KERNEL EXCHANGE MODEL

Introduction

- **Sentry is a microkernel**
  - designed to minimize attack surface
  - simplify formal verification, and improve system security
- **Critical design choice:**
  - no pointer passing between user space and kernel space
- **Why?**
  - **p**ointers may reference unsafe memory regions
  - leads to attacks or unauthorized access
  - Avoids complex validity checks (bounds, mappings)
  - Simplifies reasoning and formal proofs

the *svc_exchange* concept

- **Sentry defines a per-task fixed memory region**
  - known at compile time
  - called *svc_exchange*
  - used for transferring all non-scalar data between user space and kernel space
- **Workflow**
  - the task writes data into *svc_exchange* before issuing a syscall
  - kernel reads input from svc_exchange during syscall handling
  - For complex outputs, the kernel writes results back into svc_exchange

***Note**: svc_exchange is ephemeral as its contents may be overwritten by the kernel*

Strengths and Constraints

- **Advantages**
  - stronger security: no arbitrary user pointers dereferenced
  - easier formal verification of syscall interface
  - stricter isolation: kernel does not need to permanently map user data, only svc_exchange
- **Limitations**
  - svc_exchange has a fixed, compile-time size
  - requires serialization/deserialization, adding overhead
  - large data transfers need alternative mechanisms (e.g., shared memory regions)

Usage in Sentry and examples

- **All syscalls in Sentry avoid pointer arguments for complex data**
  - they rely exclusively on svc_exchange
- **Examples**
  - Logging data sent from user space to kernel
  - Transferring structured requests with multiple fields
- **Developer considerations**
  - Define data structures carefully for svc_exchange
  - Ensure at compile time the buffer is large enough for intended usage

Note: SVC exchange usage aim to be abstracted through shield library over UAPI

Thank you !

https://github.com/camelot-os/sentry-kernel
https://sentry-kernel.readthedocs.io/en/latest/index.html