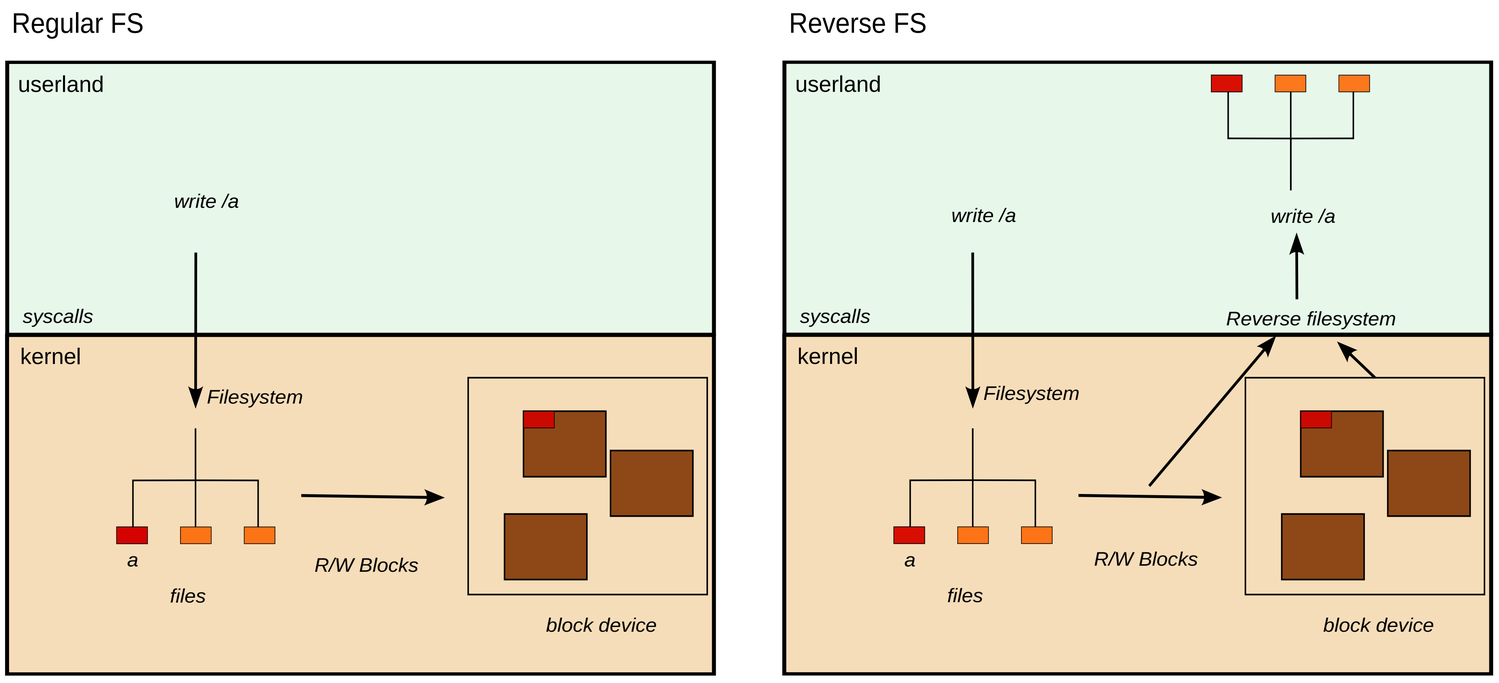
Socarrat: Building Cost-Effective Secure WORM Devices Following the Reverse File System Approach

Gorka Guardiola Múzquiz and Enrique Soriano-Salvador
Universidad Rey Juan Carlos, Spain
{gorka.guardiola,enrique.soriano}@urjc.es

Abstract: **WORM (Write Once Read Many)** devices allow data to be written once and read unlimited times, making them essential for logging and regulatory-compliant storage. Implementing secure WORM devices is complex (traditionally using paper printers or optical devices, currently using complex distributed ledger technology). Moreover, Cyberattacks pose a risk, as attackers with elevated privileges can modify or delete WORM data. We propose a **cost-effective local solution** using a small external single board device with OTG capabilities, like a Raspberry Pi, to create a USB WORM storage system for append-only log files that is tamper-evident and provides forward-integrity, based on the reverse file system approach to enable the **Continuous Printer Model (CPM): What is** *printed* **can't be** *unprinted***.**



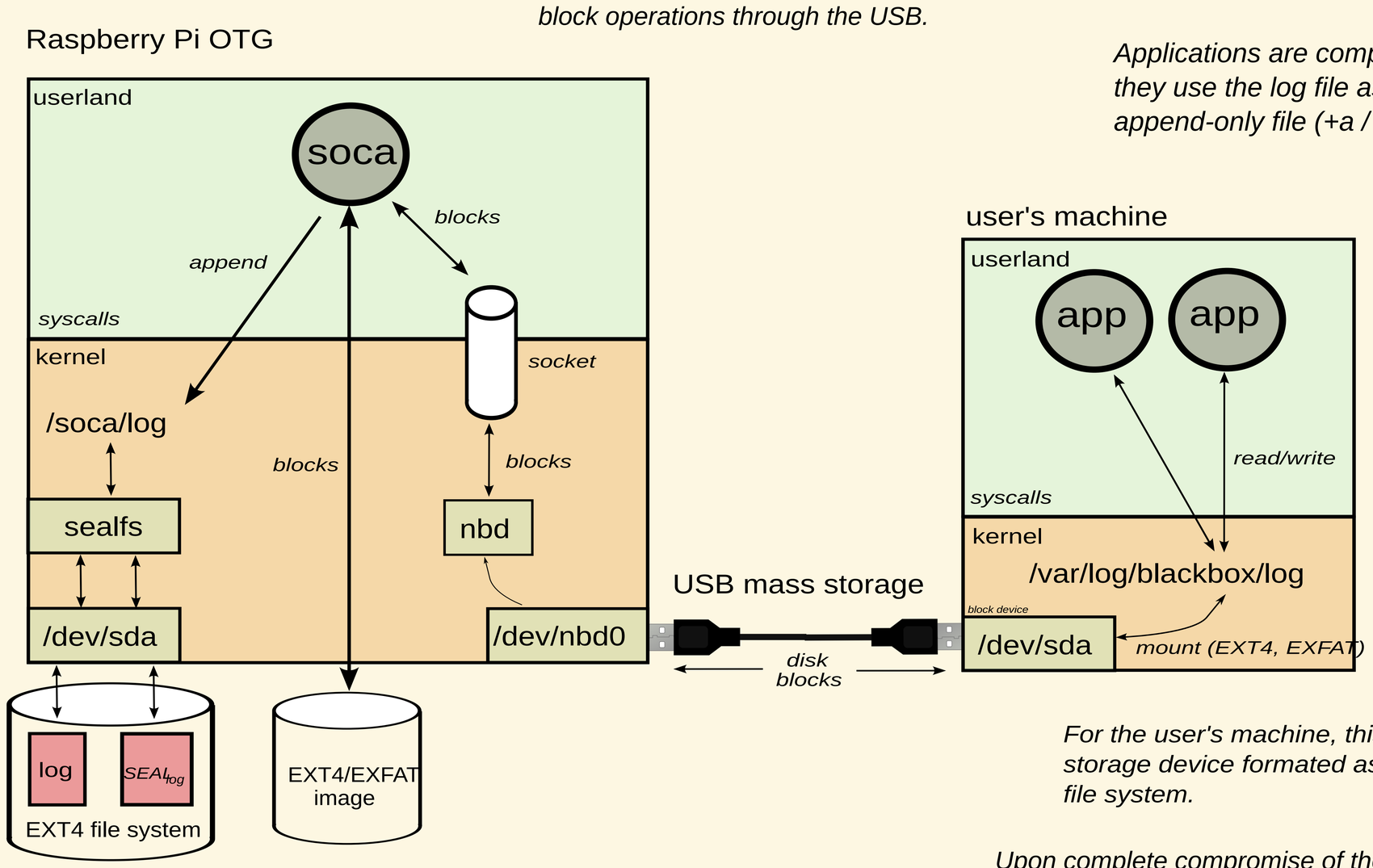
Reverse FS approach:
Analyzing the blocks that are written to the storage device (i.e. the block device) in order to infer the file system operations executed at the upper layers (i.e. the file system).

Socarrat general scheme

soca is a Go userspace program that implements the reverse FS and infer EXFAT/EXT4 operations from the written blocks.

The NBD protocol is used to receive the block operations through the USB.

Applications are completely agnostic: they use the log file as a common append-only file (+a / O_APPEND).



SealFS is a previous tamper-evident File System used to secure log files [1]

For the user's machine, this is a common USB storage device formated as an EXFAT or EXT4 file system.

Only appended data is written to the real, authenticated log file that can be verified by an auditor → Any modification will be detected.

Non valid data (i.e. anything but appended data for the log) is written to the file system image that is served to the user's machine: the attacker thinks she can modify the committed data in the log (but she can't) → Honeypot properties.

Upon complete compromise of the user's machine, the only attack surface to modify the data commited to the log is the USB link.

References: [1] Guardiola-Múzquiz, G., Soriano-Salvador, E. SealFSv2: combining storage-based and ratcheting for tamper-evident logging. Int. J. Inf. Secur. 22, 447–466 (2023). https://doi.org/10.1007/s10207-022-00643-1

This work is funded under the Proyectos de Generación deConocimiento 2021 call of Ministry of Science and Innovation ofSpain co-funded by the European Union, projectPID2021-126592OB-C22 CASCAR/DMARCE.