Untrusted Hardware Causes Double-fetch Problems in the I/O Memory

Kai Lu, Pengfei Wang, Gen Li, Xu Zhou

Lu K, Wang PF, Li G et al. Untrusted hardware causes double-fetch problems in the I/O memory. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 33(3): 587–602 May 2018. DOI 10.1007/s11390-018-1842-3

Motivation

- USB autoplay attack
- BadUSB (Blackhat 2014)
  - performed attacks by writing malicious code onto USB control chips
  - brought the risks from what the devices carry to the core of how they work
  - The underlying issue is the inability to guarantee and verify the functionality and integrity of the connected devices.

Hardware Double Fetch

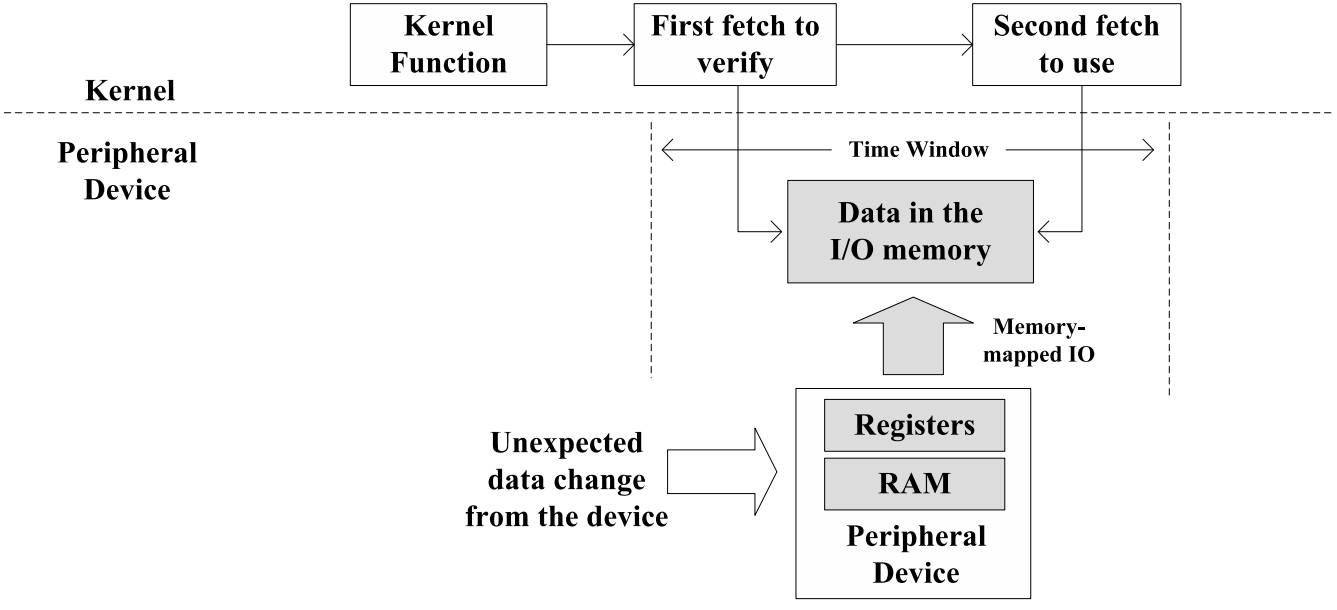

Figure 1: Illustration of How A Hardware Double Fetch Happens

- Operating systems control peripheral devices by reading from and writing to the device registers via memory-mapped I/O.
- Compromised hardware could flip the data between two reads of the "same" I/O memory data.

Static Pattern-matching Approach

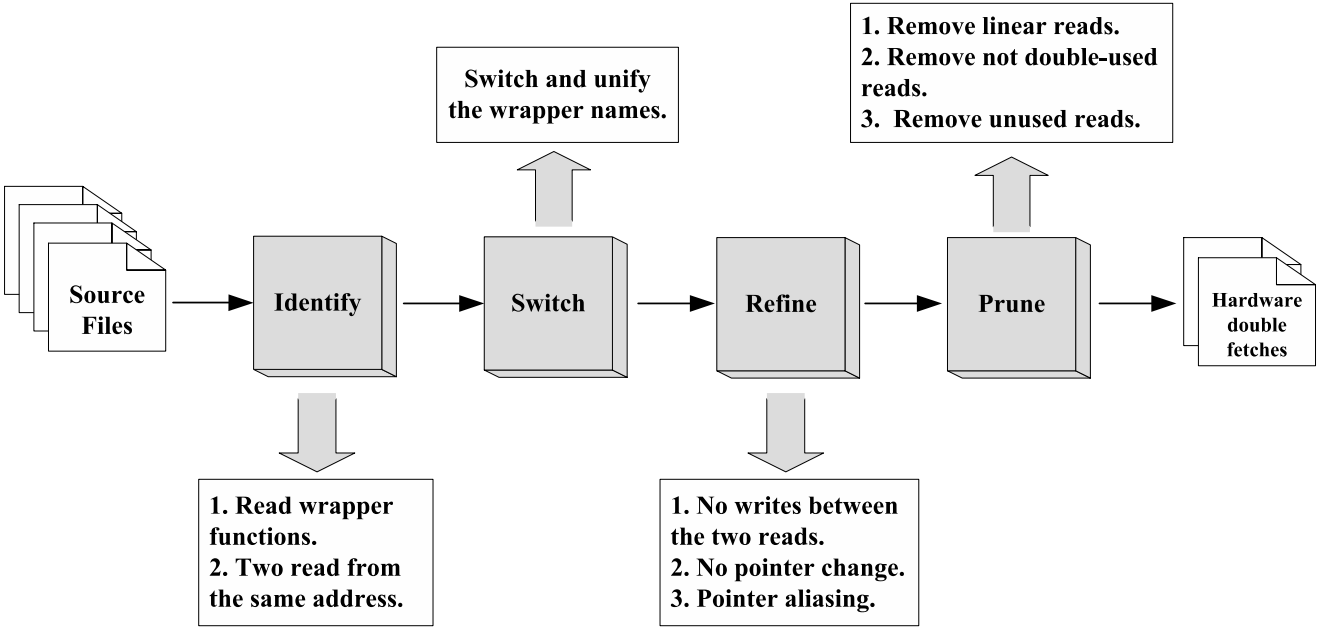

Figure 2: Overview of the Static Pattern-Matching Approach

- Could analyze the complete kernel (including all drivers) without relying on the corresponding hardware
- Open sourced: https://github.com/wpengfei/hardware_df

Results

Table 1: Identified Hardware Double Fetches Results

Types	Categories	Occurrences &Percentage	True Bugs
Status Regs	Common Check	59 (16.3%)	0
	Loop Check	80 (22.2%)	0
	Wait Check	81 (22.4%)	0
	Stable Check	18 (5.0%)	0
Configure Regs	Configure Check	29 (8.0%)	0
Data Regs	Check and Use	68 (18.8%)	3
Device Mem	Block Check	1 (0.3%)	1
Special	Flush Write	17 (4.7%)	0
	Double Valid	6 (1.7%)	0
	Delay	2 (0.6%)	0
Total	–	361 (100%)	4

- Confirmed vulnerabilities:
  - CVE-2017-8831、CVE-2017-9984
  - CVE-2017-9985、CVE-2017-9986

- Applied to Linux kernel-4.10.1
- took approximately 28 minutes
- got 361 occurrences of hardware double fetches from 178 candidate files out of 42,417 source files (.c or .h files) of the entire Linux kernel.