Bibliography

[1] The MITRE Corporation. *CVE-2014-0160*. `https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160`. 2013.

[2] Trevor Jim et al. “Cyclone: A Safe Dialect of C”. In: *Proceedings of the General Track of the Annual Conference on USENIX Annual Technical Conference*. ATEC ’02. USA: USENIX Association, June 2002, pp. 275–288. ISBN: 1880446006.

[3] George C. Necula, Scott McPeak, and Westley Weimer. “CCured: Type-Safe Retrofitting of Legacy Code”. In: *Proceedings of the 29th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL ’02. Portland, Oregon: Association for Computing Machinery, Jan. 2002, pp. 128–139.

[4] Archibald Samuel Elliott et al. “Checked C: Making C Safe by Extension”. In: *2018 IEEE Cybersecurity Development (SecDev)*. Cambridge, MA, USA, Sept. 2018, pp. 53–60.

[5] Thomas Bourgeat et al. “MI6: Secure Enclaves in a Speculative Out-of-Order Processor”. In: *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*. MICRO ’52. Columbus, OH, USA: Association for Computing Machinery, Oct. 2019, pp. 42–56.

[6] Marno van der Maas and Simon W. Moore. “Protecting Enclaves from Intra-Core Side-Channel Attacks through Physical Isolation”. In: *Proceedings of the 2nd Workshop on Cyber-Security Arms Race*. CYSARM’20. Virtual Event, USA: Association for Computing Machinery, Nov. 2020, pp. 1–12.

[7] Maurice V. Wilkes and Roger M. Needham. *The Cambridge CAP Computer and Its Operating System*. Elsevier, Jan. 1979.

[8] William B. Ackerman and William W. Plummer. “An implementation of a multiprocessing computer system”. In: *SOSP ’67: Proceedings of the First ACM Symposium on Operating System Principles*. New York, NY, USA: ACM, 1967, pp. 5.1–5.10.

[9] Dmitry Evtyushkin et al. “BranchScope: A New Side-Channel Attack on Directional Branch Predictor”. In: *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*. ASPLOS ’18. Williamsburg, VA, USA: Association for Computing Machinery, Mar. 2018, pp. 693–707.

[10] Krste Asanović and David A. Patterson. *Instruction Sets Should Be Free: The Case For RISC-V*. Tech. rep. UCB/EECS-2014-146. University of California at Berkeley, Electrical Engineering and Computer Sciences, Aug. 2014.

[11] Editors Andrew Waterman and Krste Asanović. *The RISC-V Instruction Set Manual*. Document Version 20191213. Volume I: User-Level ISA. RISC-V Foundation. Dec. 2019.

[12] Editors Andrew Waterman and Krste Asanović. *The RISC-V Instruction Set Manual*. Document Version 20190608-Priv-MSU-Ratified. Volume II: Privileged Architecture. RISC-V Foundation. June 2019.

[13] Robert M. Tomasulo. “An Efficient Algorithm for Exploiting Multiple Arithmetic Units”. In: *IBM Journal of Research and Development* 11.1 (1967), pp. 25–33.

[14] David A. Patterson and John L. Hennessy. *Computer Organization and Design, RISC-V Edition: The Hardware/Software Interface*. 6th. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2017. ISBN: 9780128122754.

[15] John L. Hennessy and David A. Patterson. *Computer Architecture: A Quantitative Approach*. 6th. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2017. ISBN: 9780128119068.

[16] David M. Gallagher et al. “Dynamic Memory Disambiguation Using the Memory Conflict Buffer”. In: *Conference on Architectural Support for Programming Languages and Operating Systems*. San Jose, CA, USA, Oct. 1994.

[17] Martin Schwarzl et al. *Speculative Dereferencing of Registers: Reviving Foreshadow*. Aug. 2020. arXiv: 2008.02307.

[18] Yuval Yarom and Katrina Falkner. “FLUSH+RELOAD: A High Resolution, Low Noise, L3 Cache Side-Channel Attack”. In: *USENIX Security Symposium*. San Diego, CA: USENIX Association, Aug. 2014, pp. 719–732.

[19] Claudio Canella et al. “A Systematic Evaluation of Transient Execution Attacks and Defenses”. In: *Proceedings of the 28th USENIX Conference on Security Symposium*. SEC’19. Santa Clara, CA, USA: USENIX Association, Aug. 2019, pp. 249–266.

[20] Jo Van Bulck et al. “LVI: Hijacking Transient Execution through Microarchitectural Load Value Injection”. In: *2020 IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA, 2020, pp. 54–72.

[21] Robert N. M. Watson et al. *Capability Hardware Enhanced RISC Instructions (CHERI): Notes on the Meltdown and Spectre Attacks*. Tech. rep. UCAM-CL-TR-916. University of Cambridge, Computer Laboratory, Feb. 2018. URL: `https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-916.pdf`.

[22] Paul Kocher et al. “Spectre Attacks: Exploiting Speculative Execution”. In: *IEEE Symposium on Security and Privacy*. San Francisco, CA, USA, May 2019.

[23] Esmaeil Mohammadian Koruyeh et al. “Spectre Returns! Speculation Attacks Using the Return Stack Buffer”. In: *Proceedings of the 12th USENIX Conference on Offensive Technologies*. WOOT’18. Baltimore, MD, USA: USENIX Association, Aug. 2018.

[24] Giorgi Maisuradze and Christian Rossow. “Ret2spec: Speculative Execution Using Return Stack Buffers”. In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’18. Toronto, Canada: Association for Computing Machinery, Jan. 2018, pp. 2109–2122.

[25] Jan Horn. *speculative execution, variant 4: speculative store bypass*. `https://bugs.chromium.org/p/project-zero/issues/detail?id=1528`. Feb. 2018.

[26] Stephan Van Schaik et al. “RIDL: Rogue In-Flight Data Load”. In: *IEEE Symposium on Security and Privacy*. San Francisco, CA, USA, May 2019.

[27] Moritz Lipp et al. “Meltdown: Reading Kernel Memory from User Space”. In: *Commun. ACM* (May 2020), pp. 46–56.

[28] Jo Van Bulck et al. “Foreshadow: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution”. In: *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, 991–1008.

[29] Intel Corporation. *Intel® Software Guard Extensions Developer Guide*. `https://software.intel.com/content/www/us/en/develop/documentation/sgx-developer-guide/top.html`. Sept. 2016.

[30] Intel Corporation. *Deep Dive: Intel Analysis of L1 Terminal Fault*. Tech. rep. 2018. URL: `%5Curl%7Bhttps://software.intel.com/security-software-guidance/advisory-guidance/l1-terminal-fault%7D`.

[31] Ofir Weisse et al. *Foreshadow-NG: Breaking the Virtual Memory Abstraction with Transient Out-of-Order Execution*. Tech. rep. 1.0. Aug. 2018, p. 7. URL: `https://foreshadowattack.eu/foreshadow-NG.pdf`.

[32] Arm Limited. *Cache Speculation Side-channels*. Tech. rep. 2.5. 2020, p. 21. URL: `https://developer.arm.com/support/arm-security-updates/speculative-processor-vulnerability`.

[33] Intel Corporation. *Intel Analysis of Speculative Execution Side Channels*. Tech. rep. 4.0. 2018, p. 16. URL: `https://www.intel.com/content/www/us/en/architecture-and-technology/intel-analysis-of-speculative-execution-side-channels-paper.html`.

[34] Vladimir Kiriansky and Carl Waldspurger. *Speculative Buffer Overflows: Attacks and Defenses*. 2018. arXiv: `1807.03757 [cs.CR]`.

[35] Dag Arne Osvik, Adi Shamir, and Eran Tromer. “Cache Attacks and Countermeasures: The Case of AES”. In: *Proceedings of the 2006 The Cryptographers’ Track at the RSA Conference on Topics in Cryptology*. CT-RSA’06. San Jose, CA: Springer-Verlag, 2006, pp. 1–20.

[36] Arm Limited. *Arm v8.5-A CPU updates*. `https://developer.arm.com/support/arm-security-updates/speculative-processor-vulnerability`. Version 1.4. June 2019.

[37] Robert N. M. Watson et al. *Capability Hardware Enhanced RISC Instructions: CHERI Instruction-Set Architecture (Version 8)*. Tech. rep. UCAM-CL-TR-951. University of Cambridge, Computer Laboratory, Oct. 2020. URL: https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-951.pdf.

[38] Jonathan Woodruff et al. "CHERI Concentrate: Practical Compressed Capabilities". In: *IEEE Transactions on Computers* 68.10 (2019), pp. 1455–1469.

[39] Brooks Davis et al. *CheriABI: Enforcing valid pointer provenance and minimizing pointer privilege in the POSIX C run-time environment*. Tech. rep. UCAM-CL-TR-932. University of Cambridge, Computer Laboratory, Apr. 2019. URL: https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-932.pdf.

[40] Hongyan Xia et al. "CheriRTOS: A Capability Model for Embedded Devices". In: *2018 IEEE 36th International Conference on Computer Design (ICCD)*. Orlando, FL, USA: IEEE Computer Society, Oct. 2018, pp. 92–99.

[41] David Kaplan, Jeremy Powell, and Tom Woller. *AMD SEV-SNP: Strengthening VM Isolationwith Integrity Protection and More*. Tech. rep. Advanced Micro Devices Inc., Jan. 2020. URL: https://www.amd.com/system/files/TechDocs/SEV-SNP-strengthening-vm-isolation-with-integrity-protection-and-more.pdf.

[42] Abraham Gonzalez et al. "Replicating and Mitigating Spectre Attacks on a Open Source RISC-V Microarchitecture". In: *Third Workshop on Computer Architecture Research with RISC-V*. Phoenix, AZ, USA, June 2019.

[43] Christopher Celio, David A. Patterson, and Krste Asanović. *The Berkeley Out-of-Order Machine (BOOM): An Industry-Competitive, Synthesizable, Parameterized RISC-V Processor*. Tech. rep. UCB/EECS-2015-167. University of California at Berkeley, Electrical Engineering and Computer Sciences, June 2015.

[44] Anh-Tien Le et al. "Experiment on Replication of Side Channel Attack via Cache of RISC-V Berkeley Out-of-Order Machine (BOOM) Implemented on FPGA". In: *Fourth Workshop on Computer Architecture Research with RISC-V (CARRV 2020)*. Valencia, Spain, May 2020.

[45] Arm Limited. *Vulnerability of Speculative Processors to Cache Timing Side-Channel Mechanism*. `https://developer.arm.com/support/arm-security-updates/speculative-processor-vulnerability`. 2020.

[46] Sizhou Zhang et al. "Composable Building Blocks to Open up Processor Design". In: *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. Fukouka, Japan, Oct. 2018, pp. 68–81.

[47] Zhen Hang Jiang and Yunsi Fei. "A novel cache bank timing attack". In: *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. Irvine, CA, USA, Nov. 2017, pp. 139–146.

[48] Hovav Shacham. "The Geometry of Innocent Flesh on the Bone: Return-into-Libc without Function Calls (on the X86)". In: *Proceedings of the 14th ACM Conference on Computer and Communications Security*. CCS '07. Alexandria, Virginia, USA: Association for Computing Machinery, 2007, pp. 552–561.

[49] Arm Limited. *Arm Architecture Reference Manual Supplement Morello for A-profile Architecture*. DDI0606. Arm Limited. Sept. 2020.

Appendix A

Full C Attack

```
 /*
 * Author: Franz Fuchs
 *
 * Spectre-PHT proof of concept version
 *
 * spec_funct first checks the array bounds
 * and then loads the value determined by the
 * index. By training the Pattern History Table
 * with 16 calls to the function with valid indexes,
 * we trick Toooba in speculatively executing
 * the loads even though the index is out of bounds.
 */

#ifdef __CHERI_PURE_CAPABILITY__
#include "pure_cap.h"
#endif

#define MEM_SIZE 16384
#define MEM_SIZE_DW MEM_SIZE/8
#define STACK_SIZE 2048
#define STACK_SIZE_DW STACK_SIZE/8
#define PROBE_SIZE 2048
#define PROBE_SIZE_DW PROBE_SIZE/8
#define SEC_ARR_SIZE 128
#define SEC_ARR_SIZE_DW SEC_ARR_SIZE/8
#define FLUSH_ARR_SIZE 16384
```