Spectroscopic: A practical PB Inception Attack and Implications for Confidential Computing

Kaya Ercihan, Jonathan Müller

ZHAW
School of Engineering

Feb. 03, 2026

Summary

Overview

Overview

- **Literature baseline:** 17 primary sources (9 papers)
  - 7 attack papers covering **9 distinct** side-channel / speculative-execution attacks
  - 2 mitigation-focused papers + vendor/kernel documentation
- **Reproducibility artifact:** end-to-end PoC (*guest userland harness + guest LKM gadget*) with runbook and A/B toggles
- **Communication:** established a repro channel with **AMD PSIRT** and answered follow-up validation questions
- **Clarity:** diagrams of attack flow and component interfaces (sysfs gadget, trampolines, ASM harness, driver)

Outcome: a concrete, auditable implementation blueprint for post-IBPB return steering on Zen 1/2 in a CC-light (SEV/SEV-ES) setting.

Spectre example

Spectre example (bound check bypass)

```c
int main(void) {
    uint8_t array1[] = {1, 2, 3, 4};
    size_t array1_size = sizeof(array1) / sizeof(array1[0]);

    uint8_t array2[256 * 256] = {0};

    int x = 0; //would be user controllable
    volatile uint8_t y = 0;

    if ((size_t)x < array1_size) {
        y = array2[(size_t)array1[x] * 256];
    }
    return 0;
}

void gadget(void){
    printf("Pawned");
}
```

```asm
main:
        push    rbp
        mov     rbp, rsp
        sub     rsp, 65568
        mov     DWORD PTR [rbp-16], 67305985
        mov     QWORD PTR [rbp-8], 4
        lea     rax, [rbp-65552]
        mov     edx, 65536
        mov     esi, 0
        mov     rdi, rax
        call    memset
        mov     DWORD PTR [rbp-12], 0
        mov     BYTE PTR [rbp-65553], 0
        mov     eax, DWORD PTR [rbp-12]
        cdqe
        cmp     rax, QWORD PTR [rbp-8]
        jnb     .L2
        mov     eax, DWORD PTR [rbp-12]
        cdqe
        movzx   eax, BYTE PTR [rbp-16+rax]
        movzx   eax, al
        sal     rax, 8
        movzx   eax, BYTE PTR [rbp-65552+rax]
        mov     BYTE PTR [rbp-65553], al
.L2:
        mov     eax, 0
        leave
        ret
.LC0:
        .string "Pawned"
gadget:
        push    rbp
        mov     rbp, rsp
        mov     edi, OFFSET FLAT:.LC0
        mov     eax, 0
        call    printf
        nop
        pop     rbp
        ret
```

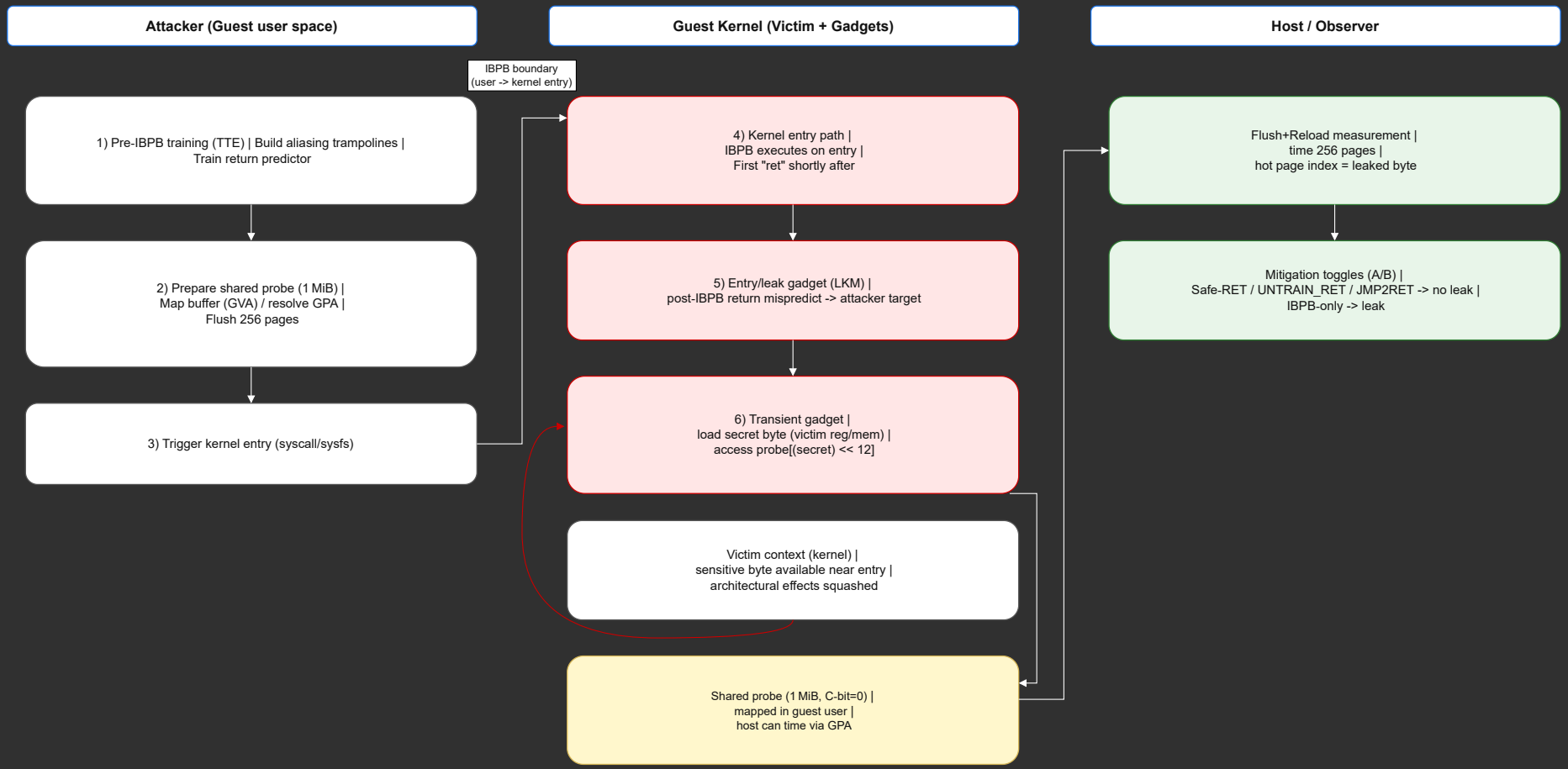
Related works

Related works

Attack name	Post-IBPB	Attack type	Attack model	Preconditions	Target processors	Mitigations	Side channel	CC Impact
RETBLEED [2]	No	Kernel-level speculative code execution targeting the Branch Target Buffer	Unprivileged User Process $\longrightarrow$ Kernel	RSB underflow to BTB fallback (Intel); BTB precedence over RSB (AMD); deep call stacks (Intel); cross-privilege BTB poisoning	AMD (Zen 1, Zen 2), Intel (Kaby Lake, Coffee Lake)	Intel (eIBRS and IBRS), AMD (jmp2ret)	Flush+Reload, Prime+Probe	Bypasses Retpoline; leaks arbitrary kernel memory; intra-guest leakage; host view requires shared side-channel
PB-RRSBA [1]	Yes	Exploit IP-based predictions of the Restricted RSB Alternate (RRSBA) predictor	Cross-process	RSB empty-state; IP-based RRSBA predictor active; IBPB execution; same-thread attacker/victim co-location	Intel (Golden Cove, Raptor Cove)	Enabling the RRSBA_DIS_U chicken bit	Flush+Reload	Leaks from IBPB-protected suid processes; shared library mappings used for cross-process exfiltration
PB-Inception [1]	Yes	Variant of Inception attack targeting Return Stack Buffer predictions	Cross-process	IBPB-on-entry active; TTE/Phantom training pre-IBPB; RSB persistence post-IBPB (AMD Zen 1/2)	AMD (Zen 1, Zen 2)	RSB stuffing	Secret-dependent cache side channel	Bypasses kernel/hypervisor protection (VMExit); host observation requires shared C-bit=0 mapping
PHANTOM [3]	No	Confuse the CPU's branch prediction unit by using asymmetric combinations of training and victim instruction types	Execute unprivileged code on top of a recent Linux kernel	Speculation before instruction decode; cross-privilege BTB aliasing; physically contiguous huge pages for Prime+Probe	AMD (Zen 1, Zen 2, Zen 3, Zen 4)	SuppressBPOnNonBr (AMD MSR), AutoIBRS (AMD Zen 4), IBPB	Flush+Reload, Prime+Probe	Bypasses AutoIBRS/-SuppressBPOnNonBr; enables transient fetch/decode leaks; host visibility needs shared memory

Attack flow & Reporting

Spectroscopic attack flow

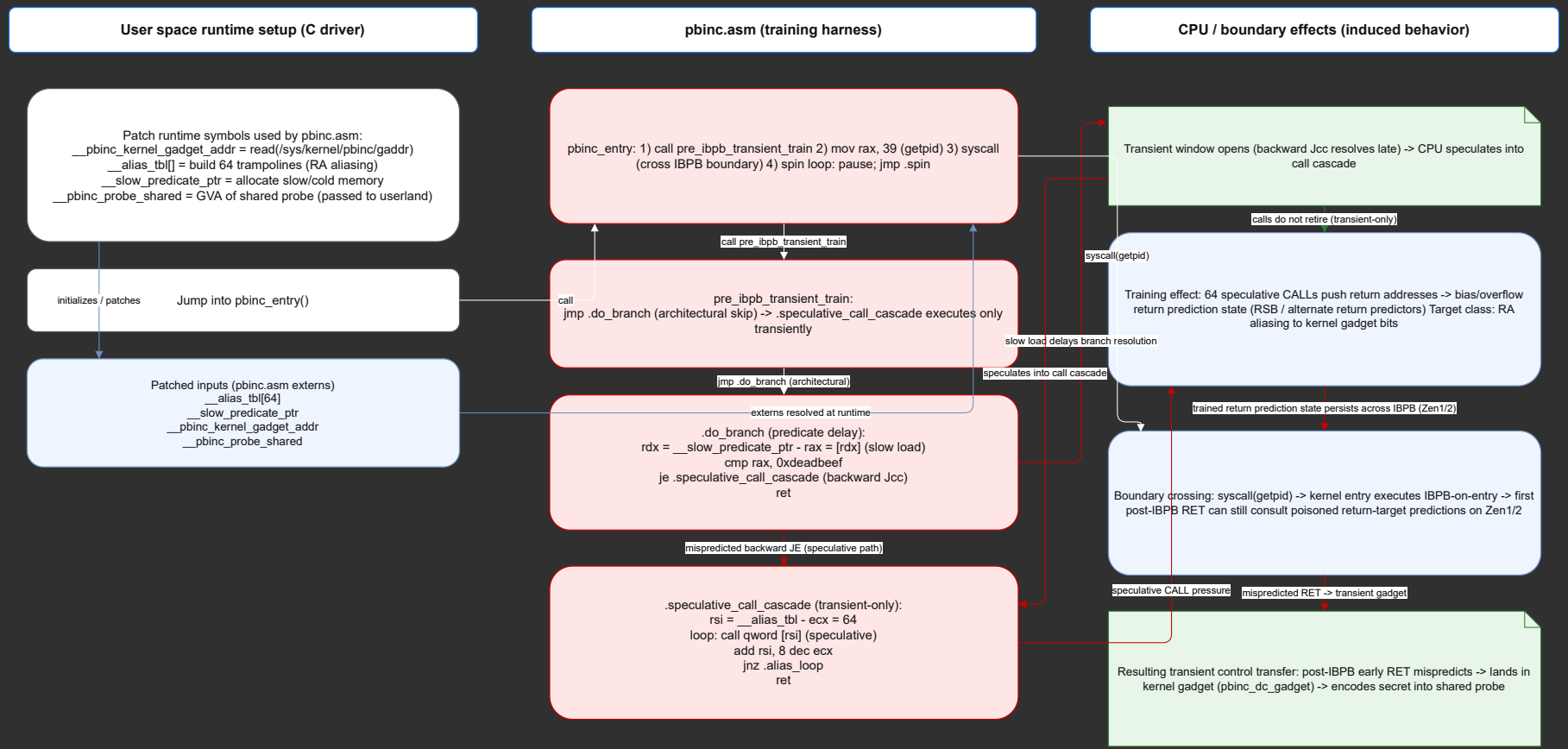


Report to AMD

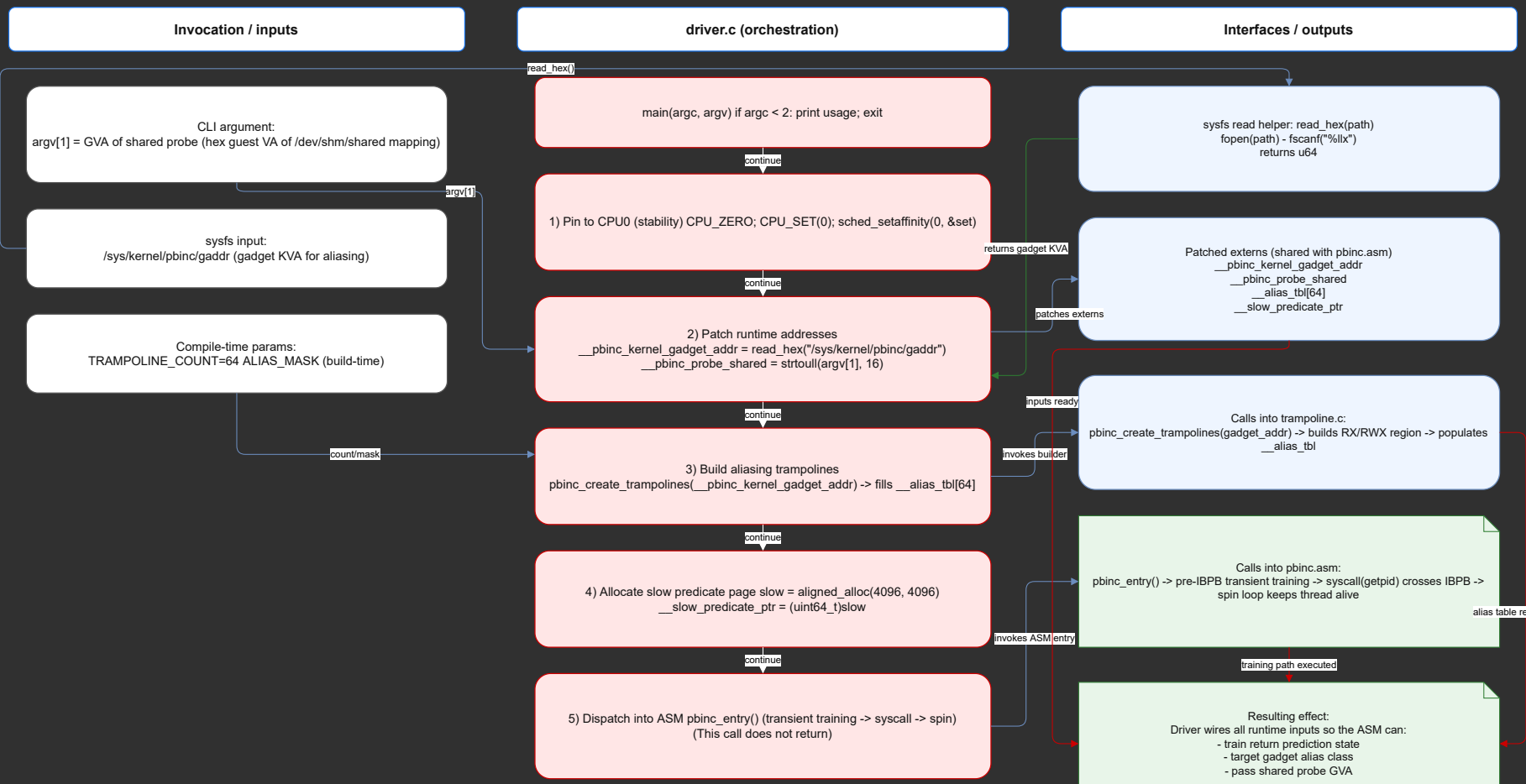
We reported the exploit and are currently in contact with AMD

Application-part visualizations

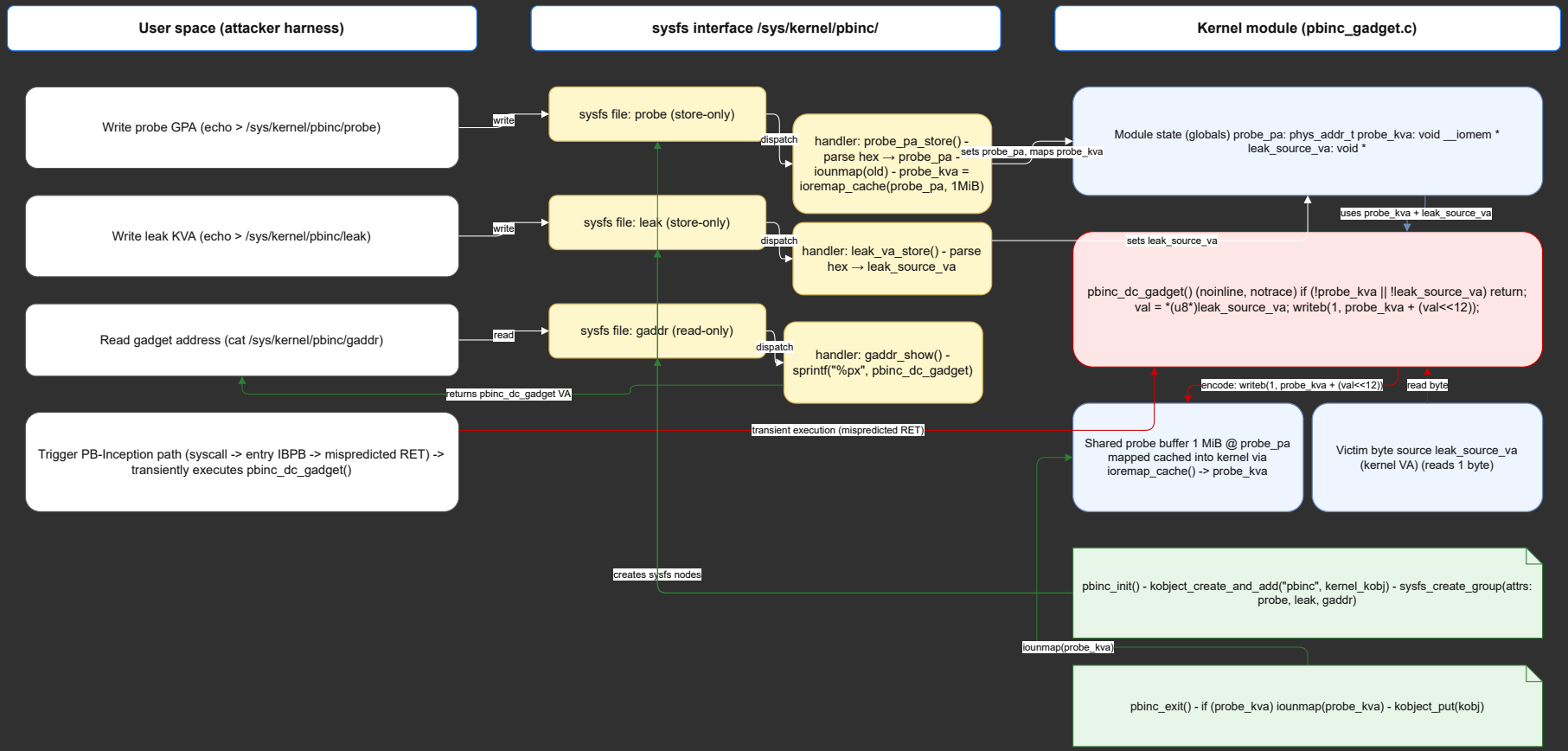
Assembly-Level Gadget Walkthrough



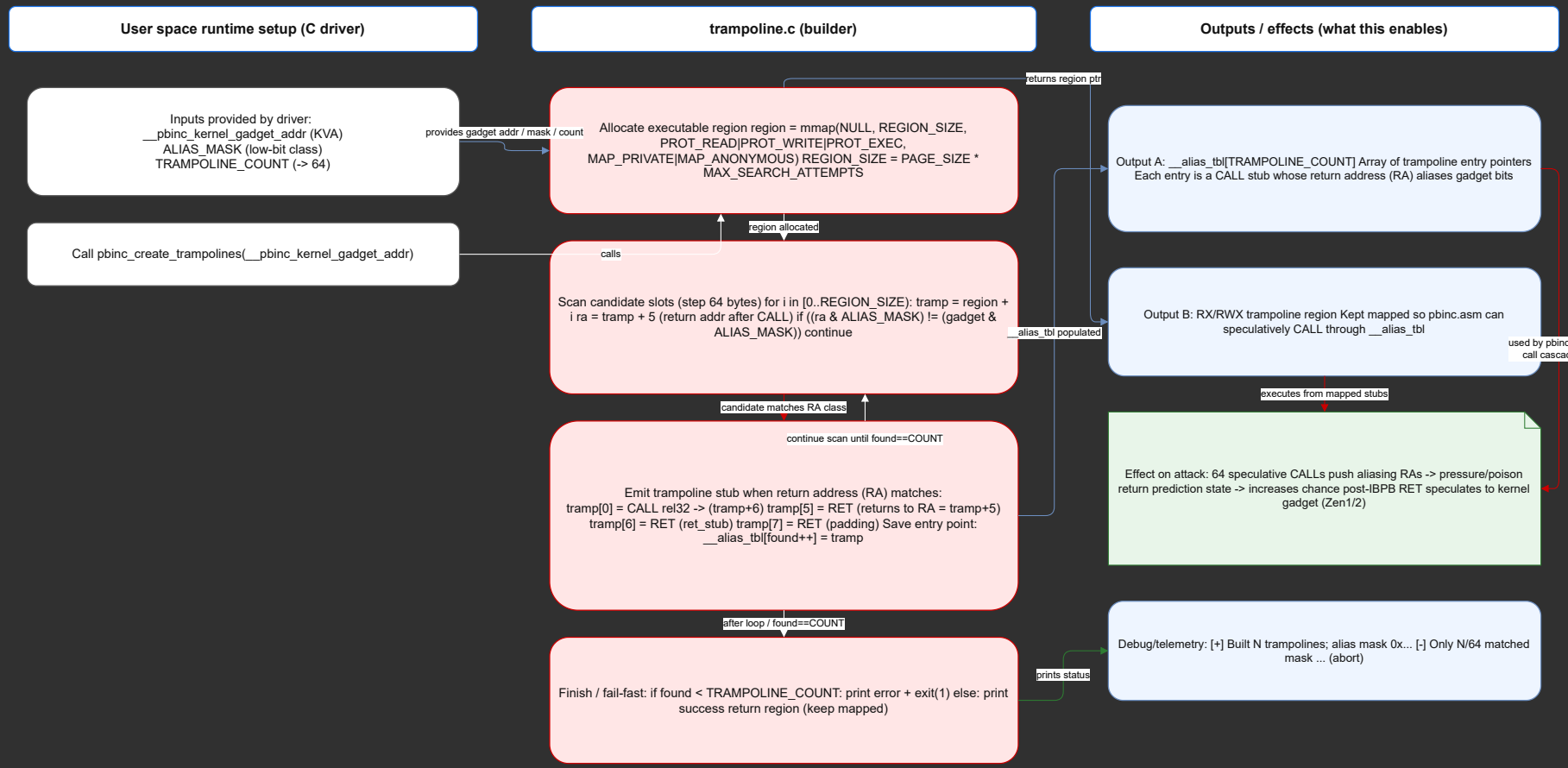
Assembly-Level Gadget Walkthrough



Assembly-Level Gadget Walkthrough



Assembly-Level Gadget Walkthrough



Thanks for your attention!

Sources I

1. Wikner, J. & Razavi, K. *Breaking the Barrier: Post-Barrier Spectre Attacks.* in *IEEE Symposium on Security and Privacy* (2025). https://comsec.ethz.ch/wp-content/files/ibpb_sp25.pdf.
2. Wikner, J. & Razavi, K. *RETbleed: Arbitrary Speculative Code Execution with Return Instructions.* in *31st USENIX Security Symposium* (2022). https://www.usenix.org/system/files/sec22-wikner.pdf.
3. Wikner, J., Trujillo, D. & Razavi, K. *Phantom: Exploiting Decoder-detectable Mispredictions.* in *56th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '23)* (2023). https://comsec.ethz.ch/wp-content/files/phantom_micro23.pdf.