

Inception: AMD Zen 4 Return Address Security Microarchitectural Audit

Phantom Speculative Execution, Training in Transient Execution (TTE), and Branch History Table Aliasing

EXECUTIVE ABSTRACT: Discovered by researchers at ETH Zurich (CVE-2023-20569), Inception demonstrates how AMD Zen 3 and Zen 4 microarchitectures can be coerced into leaking cross-privilege secrets. By combining Phantom Speculative Execution with Training in Transient Execution (TTE), an unprivileged attacker misdirects the CPU's Return Address Stack (RAS) without address collisions, completely bypassing AMD's hardware-level return address mitigations.

Exploit Vector	Targeted CPU Structure	Collision Prerequisite	Leakage Rate	Hardware Mitigation Required
Spectre v2 (BTI)	Branch Target Buffer (BTB)	Virtual Address Aliasing	10 - 50 kB/s	IBRS / Enhanced IBRS
Retbleed (CVE-2022-29900)	Return Address Stack (RAS)	RAS Underflow / Call Depth	20 - 40 kB/s	Safe RET / JMP Trampolines
Inception (CVE-2023-20569)	BTB + RAS (Recursive TTE)	None (Recursive injection)	15 - 30 kB/s	Microcode SPEC_CTRL update
Zenbleed (CVE-2023-20593)	Physical Register File (RAT)	Zero (Direct architectural leak)	30 kB/s	Microcode FP Control Bit

1. Phantom Speculative Execution: Branches Created from Nothing

In traditional branch misprediction attacks, an attacker manipulates an existing conditional branch or indirect jump instruction in the victim's binary.

- The Phantom Hazard: Inception discovers that AMD Zen 3/4 branch predictors can be trained to believe a branch instruction exists at an arbitrary address where no branch instruction exists in reality.
- Branch Target Buffer (BTB) Poisoning: The CPU's front-end decoder encounters non-branch opcodes, yet the branch prediction unit speculatively redirects the instruction pointer to a phantom target, executing transient instructions before decoding completes.

2. Training in Transient Execution (TTE)

Prior transient execution models assumed branch predictors could only be trained by instructions that commit architecturally. Inception breaks this assumption:

- Speculative Branch Training: Zen 4 updates branch history registers and RAS predictors during *mispredicted, transient execution paths* that are subsequently rolled back.
- Recursive RAS Poisoning: An attacker can nest speculative phantom branches to continuously overwrite the Return Address Stack with attacker-controlled gadget addresses.

3. Dismantling AMD Hardware Return Address Security (eIBRS)

AMD introduced hardware mitigations in Zen 3 and Zen 4 designed to clear or isolate branch history across privilege boundaries (such as User to Kernel or Guest VM to Hypervisor).

- RAS Underflow Circumvention: When the RAS underflows, Zen CPUs fall back to predicting return targets using the BTB. Inception trains the BTB during transient execution to inject malicious return targets that survive the privilege transition.
- Cross-VM Extraction: An unprivileged VM user extracts /etc/shadow hashes, root passwords, and RSA private keys from the hypervisor at up to 30 kB/s.

4. Silicon Remediation: Microcode Fences and Auto-IBPB

Remediating Inception requires updating processor microcode to flush branch prediction arrays on privilege domain transitions (Automatic Indirect Branch Predictor Barrier - Auto-IBPB) or inserting serializing instruction barriers, incurring a 2% to 6% CPU performance penalty in heavy syscall workloads.

STRATEGIC ARCHITECTURAL TAKEAWAYS

1. Inception (CVE-2023-20569) combines Phantom Speculative Execution with Training in Transient Execution.
2. Enables unprivileged attackers to poison the Return Address Stack (RAS) without address collisions.
3. Bypasses AMD hardware eIBRS protections; remediated via microcode Auto-IBPB flushing at a 2-6% workload cost.