PreFence: A Fine-Grained and Scheduling-Aware Defense Against Prefetching-Based Attacks

Till Schlüter, Nils Ole Tippenhauer (CISPA)

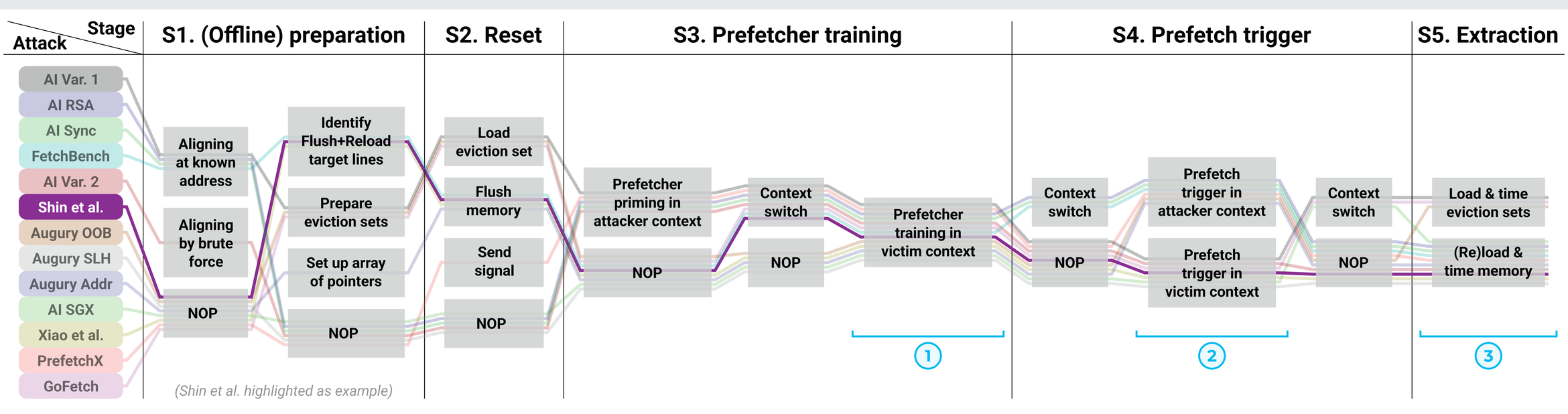
1. Prefetcher Attacks

Prior work uncovered **side-channel vulnerabilities in hardware data prefetchers** that put user data at risk. However, corresponding defenses have not been studied systematically before.

2. No Practical Defense So Far

No effective and efficient defense has been presented so far. The most effective defense is to disable prefetching permanently, which is impractical due to its **high performance cost** for all processes.

3. Attack Systematization



4. Systematization Findings

We identify three mandatory attack stages:

1. **Training in the victim context,** transferring secrets into the prefetcher's state.
2. **Triggering in the victim or attacker context,** transferring secrets into the cache state.
3. **Cache side-channel extraction,** transferring secrets into architectural state.

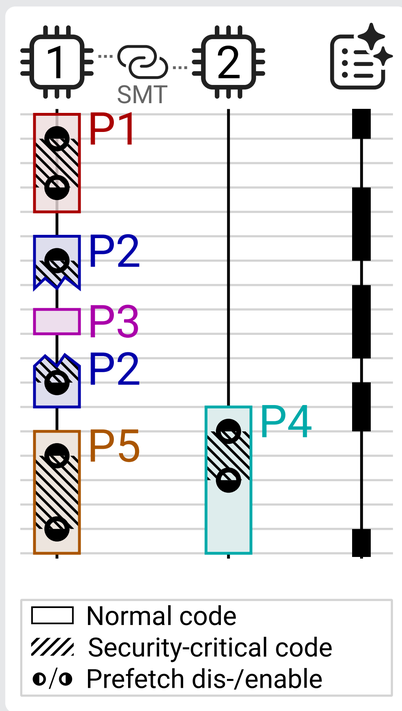
Preventing any of these stages prevents the entire class of prefetcher attacks.

5. PreFence Design

PreFence enables processes to **disable prefetching temporarily** per core to prevent training (1).

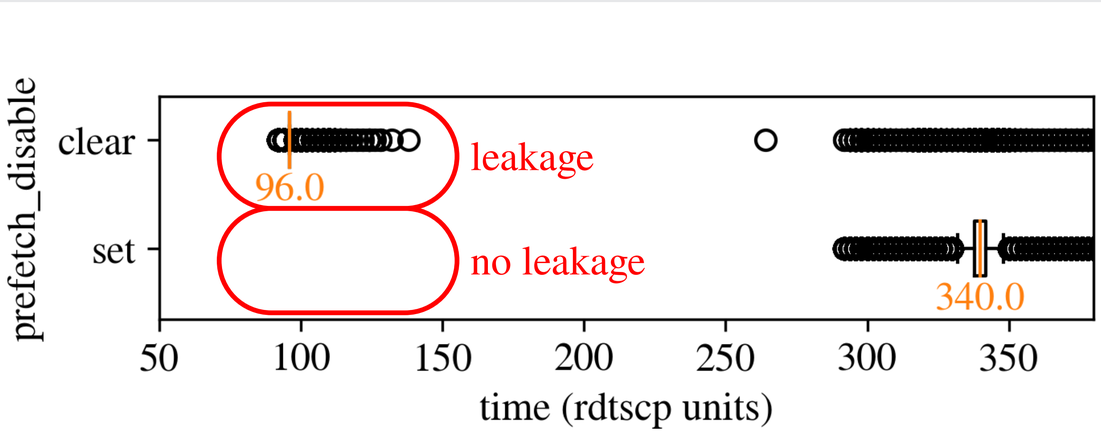
Processes send **system calls** to announce when they execute security-critical code.

We **extend the scheduler** to let it manage the prefetcher activation state, preventing attacks across processes and cores.



6. PreFence Is Effective

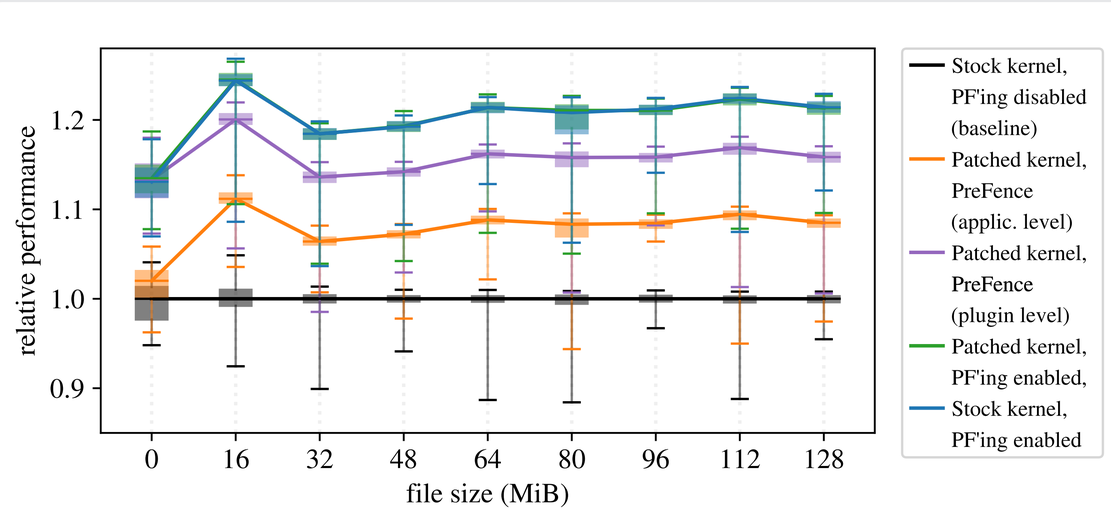
We show that PreFence is effective by successfully **preventing attacks** from prior work, for example the shared library attack by Shin et al. (CCS 2018).



Efficacy: PreFence mitigates the shared library attack by Shin et al., where the prefetcher is triggered by memory accesses to shared data and leaks secret-dependent access patterns into the cache state. PreFence prevents this successfully.

7. PreFence Is Efficient

PreFence has **negligible impact** on non-critical code and performs better than permanent disabling for critical workloads.



Efficiency: The performance of PreFence depends on how it is applied to the code of the workload. Permanent disabling (black line) is most expensive.

IEEE European Symposium on Security and Privacy (EuroS&P) 2025

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY