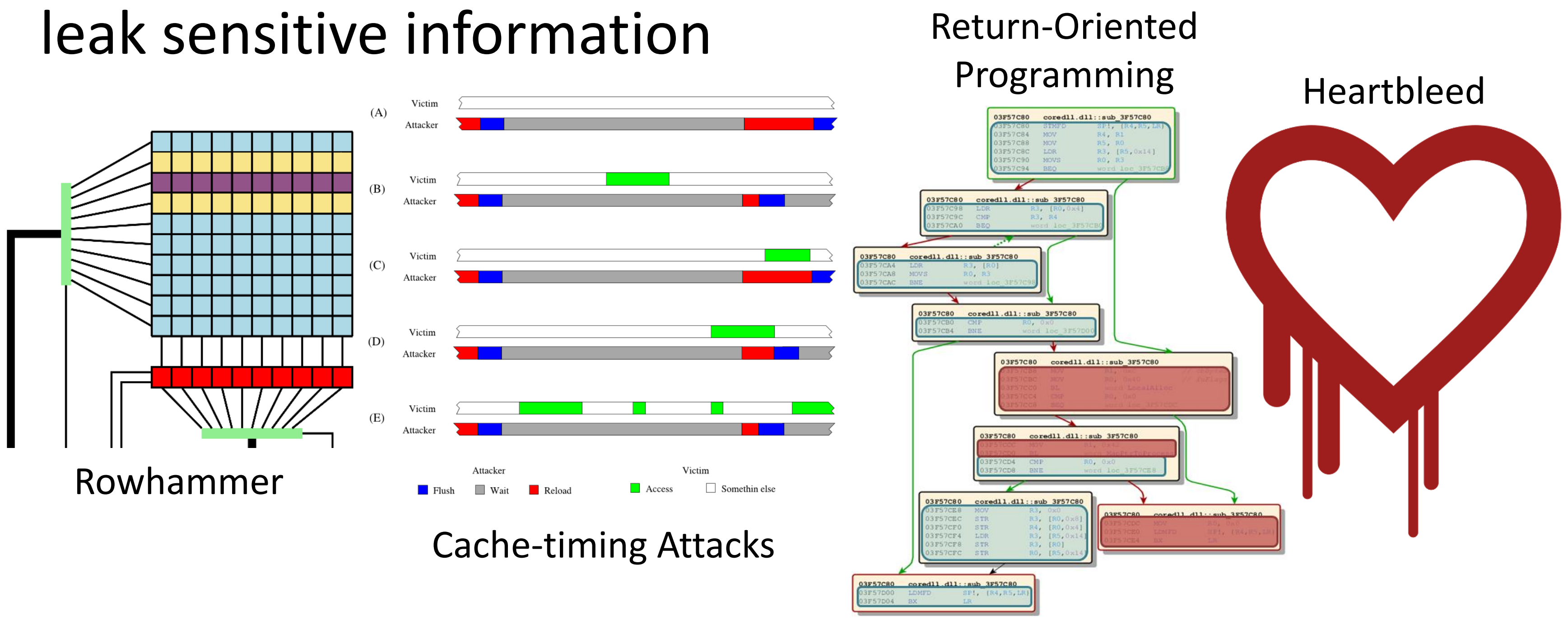
Secure Popcorn: Using Machine Boundaries To Harden Applications

Robert Lyerly[ϕ], Sergey Bratus[§] and Binoy Ravindran[ϕ]
[ϕ]Systems Software Research Group, ECE Department, Virginia Tech
{rlyerly, binoy}@vt.edu
[§]Institute for Security, Technology, and Society, CS Department, Dartmouth College
sergey@cs.dartmouth.edu

Problem

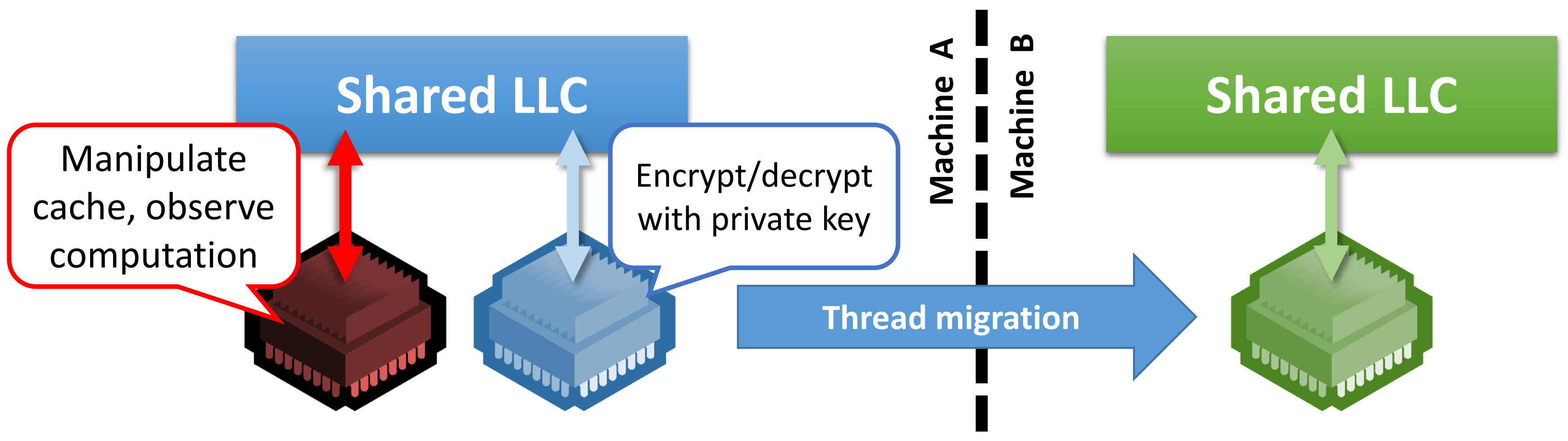
- Current security mechanisms are too coarse-grained, provide weak mitigations or incur heavy run-time costs
  - **SELinux** – inter-process “bag of permissions”, who can access which files
  - **ASLR** – load-time virtual address space layout randomization
  - **Control/data-flow integrity** – ensure control flow/memory operations use legitimate target memory addresses (enormous instrumentation)
- Exploits circumvent mechanisms to co-opt execution & leak sensitive information



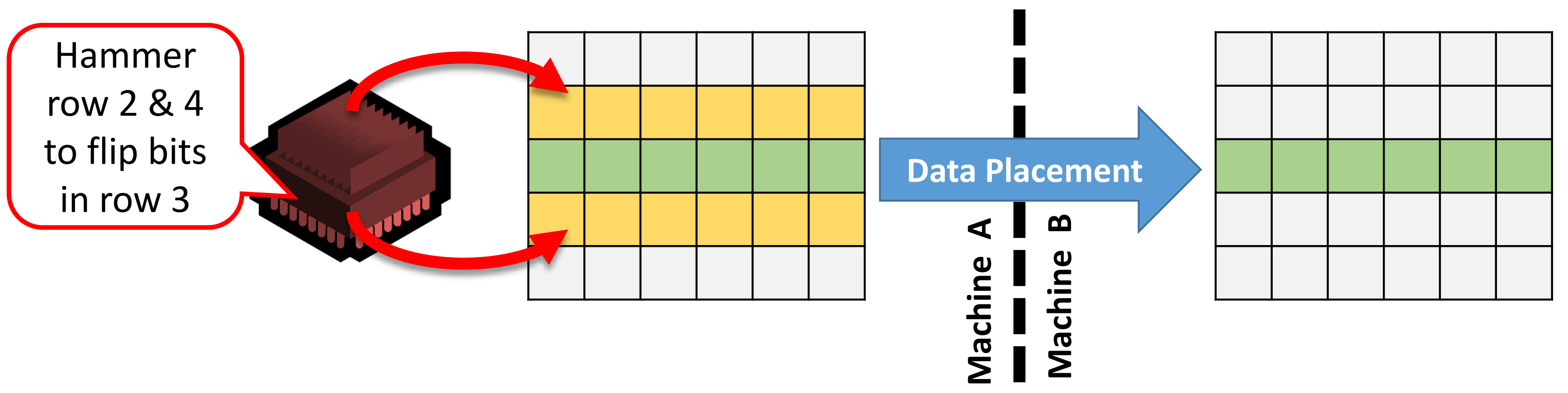
- **How do we provide better inter-/intra-process security?**

Key Idea #1: Isolation across Machine Boundaries

- Eliminate cache timing attacks by physically separating privileged compute on different machines
  - Use ELFbac’s phase transitions to drive thread migration, e.g., “entering crypto phase, migrate to new machine”
  - Popcorn OS overlays shared memory illusion on top of separate physical memory regions, removes sharing of physical last-level cache


Shared LLC
Manipulate cache, observe computation
Encrypt/decrypt with private key
Machine A
Machine B
Thread migration
Shared LLC

- Prevent memory crosstalk bit flips (and potential privilege escalation) by physically isolating access control state



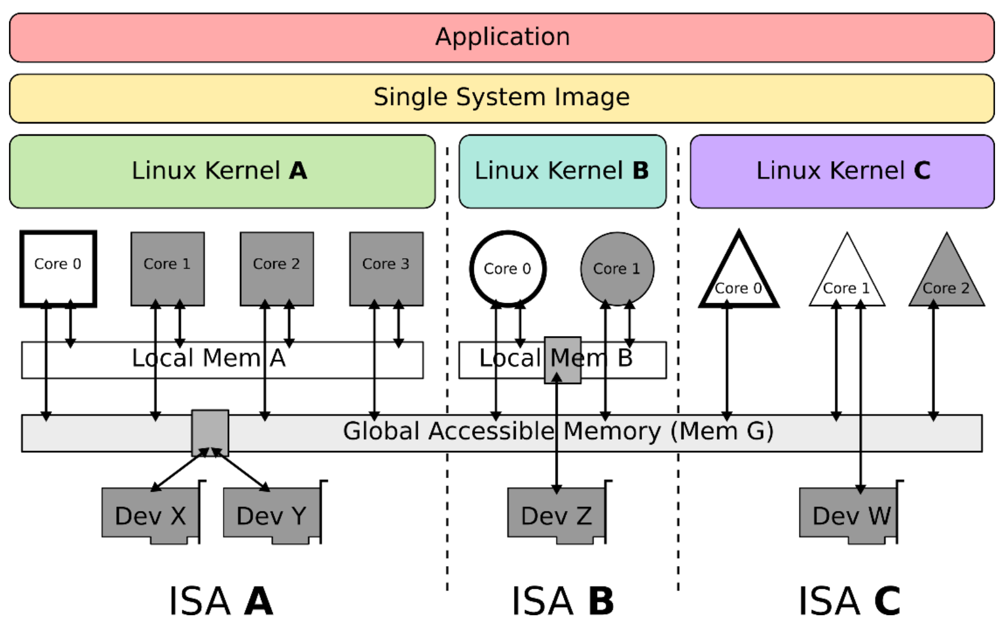
- **Secure Popcorn allows transparently placing critical data in physically isolated memory, nullifying traditional information leakage/side-channel attacks**

Conclusion

- New exploits and side-channel attacks circumvent state-of-the-art security policy and mitigation mechanisms
- Secure Popcorn utilizes system software innovations for strong physical isolation and continuous randomization
  - Popcorn Linux transparently executes C/C++ shared memory applications across heterogeneous-ISA machines
  - ELFbac enforces programmer intent by utilizing ELF section metadata
- Significantly enhance application security without requiring any developer effort

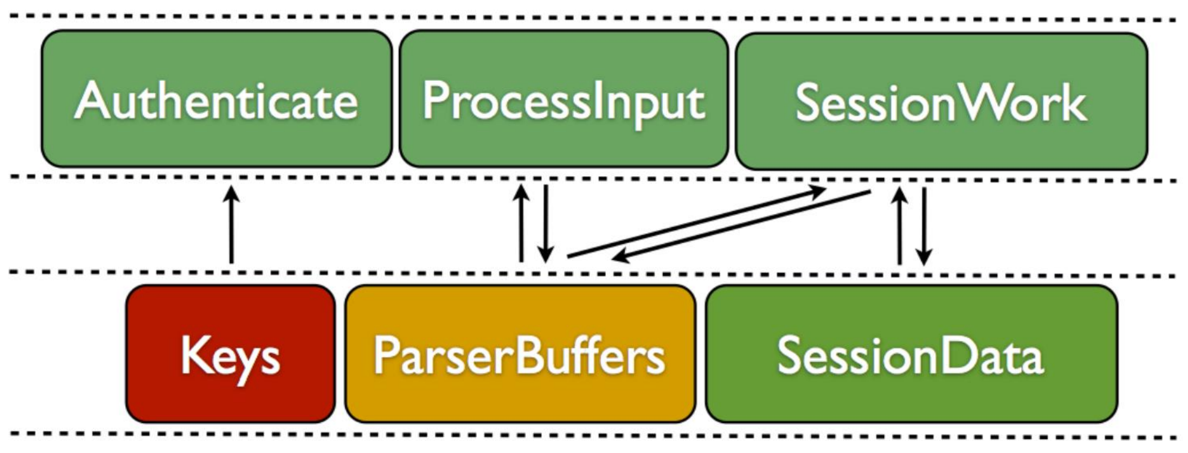
Solution: Secure Popcorn Linux

- Secure Popcorn: an OS, compiler and runtime for secure application execution across machine boundaries
  - **Popcorn Linux** – compiler/OS/runtime for transparently executing C/C++ shared memory applications across physically distinct heterogeneous-ISA machines



- OS provides per-thread execution migration across machine boundaries
  - Thread context, code/data pages, file descriptor metadata (network, filesystem)
  - Migrate between AArch64 and x86-64
- Compiler builds multi-ISA binaries
  - Custom virtual address space layout (code & data symbols), aligned across ISAs
- Runtime converts stack/registers between ISA-specific formats during migration

- **ELFbac** – virtual memory access control driven by ELF binary metadata
- ELF section metadata describes how parts of application interact
  - Code and associated data have exclusive relationships describing programmer intent
- Application phases see subset of page table entries, page faults drive phase transitions



- **Use existing ELF ABI as policy to partition intra-process computation (and associated data) into isolated physical domains**
- **Inter-ISA/machine migration mitigates usefulness of info leakages**

Key Idea #2: Runtime Randomization

- Migrate between ISAs to thwart attacks hand-crafted for a particular ISA’s function activation (stack & registers) layout
  - Migrate randomly or at ELFbac phase boundaries

Attacker-crafted x86-64 gadget

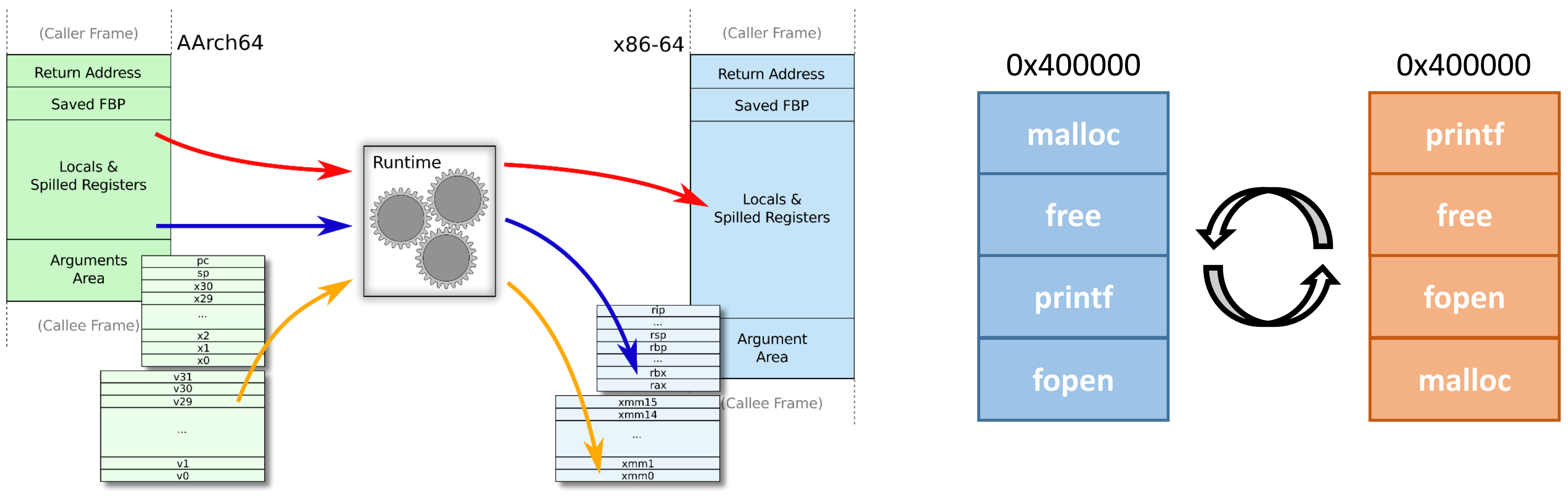
```
…
pop %ebx
xor %eax, %eax
mov %edx, -0x16(%ebp)
mov $0x3, (%esp)
mov %eax, 0x4(%esp)
…
```

Policy requests thread migration

Unknown execution stream on AArch64

```
…
add x0, x2, x3
ldr x20, x19, [sp],#64
ret
???
…
```

- Randomize code & data layout (including function activations) during inter-ISA state transformation or any migration



- **Inter-ISA migration limits the ability of attackers to chain together gadgets and gives a limited lifetime to the usefulness of any leaked memory layout information**

References

- “Exploiting the DRAM rowhammer bug to gain kernel privileges”, Seaborn and Dullien, Black Hat 2015
- “Flush+Reload: A High Resolution, Low Noise, L3 Cache Side-Channel Attack”, Yarom and Falkner, USENIX Security Symposium, 2014
- “Heartbleed Bug”, http://heartbleed.com/
- “Return-Oriented Programming”, Prandini and Ramilli, IEEE Security and Privacy, 2012.
- “Breaking the Boundaries in Heterogeneous-ISA Datacenters”, Barbalace et. al, ASPLOS 2017
- “Intra-Process Memory Protection for Applications on ARM and x86: Leveraging the ELF ABI”, Bratus, Bangert and Koo, BlackHat USA 2016