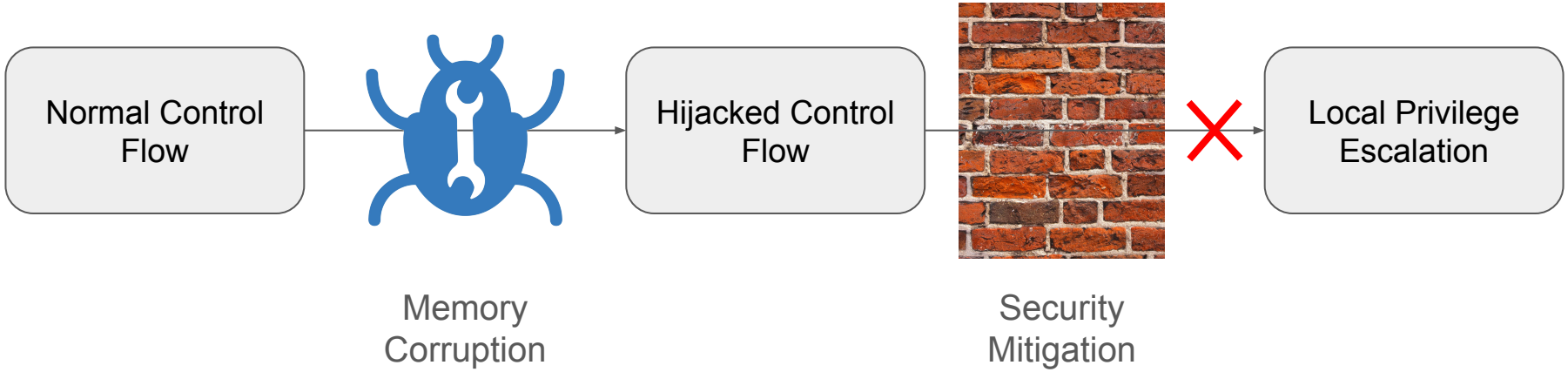
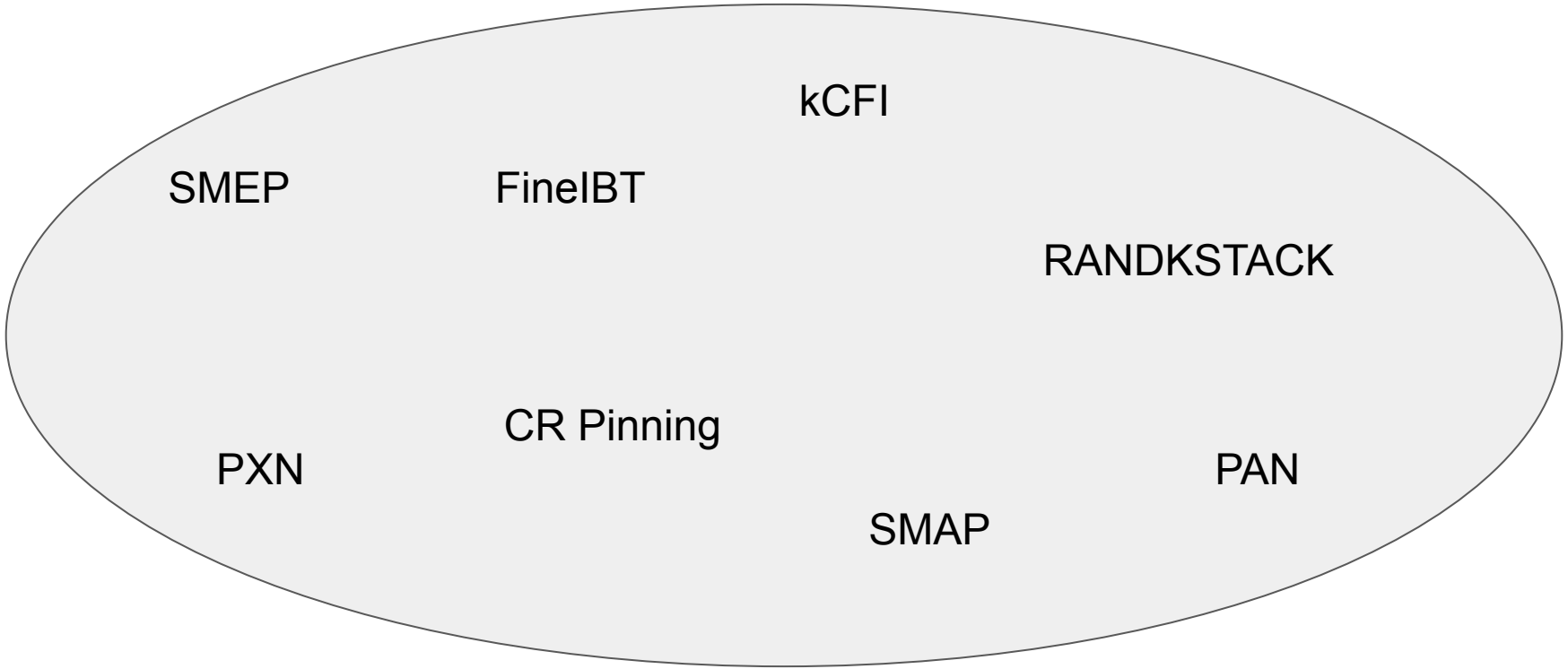
# **System Register Hijacking**: Compromising Kernel Integrity By Turning System Registers Against the System

**Jennifer Miller**, Manas Ghandat, Kyle Zeng, Hongkai Chen, Abdelouahab (Habs) Benchikh, Tiffany Bao, Ruoyu Wang, Adam Doupé, Yan Shoshitaishvili


# Linux Kernel Control Flow Hijacking



# Mitigations



# Back in 2017…

Wednesday, May 10, 2017

## Exploiting the Linux kernel via packet sockets

Guest blog post, posted by Andrey Konovalov

c. Overflow the block and overwrite retire_blk_timer field. Make retire_blk_timer->func point to native_write_cr4 and make retire_blk_timer->data equal to the desired CR4 value.

d. Wait for the timer to be executed, now we have SMEP & SMAP disabled on the current core.

# “Security-Sensitive System Registers”

Mitigation Enforcement

- `cr0`, `cr4`, `EFLAGS`, `MSR_EFER`

Architectural Structure Information

- `cr3`, `IDTR`, `GDTR`, `GSBASE`

# System Register Hijacking

We propose System Register Hijacking (SRH) as a class of exploitation techniques which involve writes to Security-Sensitive System Registers via Control Flow Hijacking in order to bypass mitigations or expand attacker capabilities.

# System Register Hijacking Techniques

**`cr4`**: Bypass SMEP/SMAP

**`cr0`**: Bypass Write Protect

**`EFLAGS.AC`**: Temporarily Bypass SMAP

**`GSBASE`**: Bypass FineIBT

… see the paper for more

# SWAPGS Stack Pivoting

A **universal stack pivoting gadget** and **FineIBT bypass** on x86-64 Linux.

The technique reuses code from syscall entrypoints:

- `endbr64` ➜ `swapgs` ➜ `mov rsp, gs:xxxx`

Entrypoints are architecturally *required* to start with `endbr64` under IBT.

# Outcomes

**FineIBT Paranoid** was introduced by Linux kernel developers, which adds a CFI check on the caller-side, mitigating our bypass.

A **specification change** was made to Intel's upcoming feature, Flexible Return and Event Delivery (FRED), to no longer require `endbr64` at FRED entrypoints.

Read the Paper:

https://tinyurl.com/srhpaper


# Thank You!

Jennifer Miller
jmill@asu.edu
blog.zolutal.io
@zolutal.bsky.social

Arizona State University

sefcom
security engineering for future computing