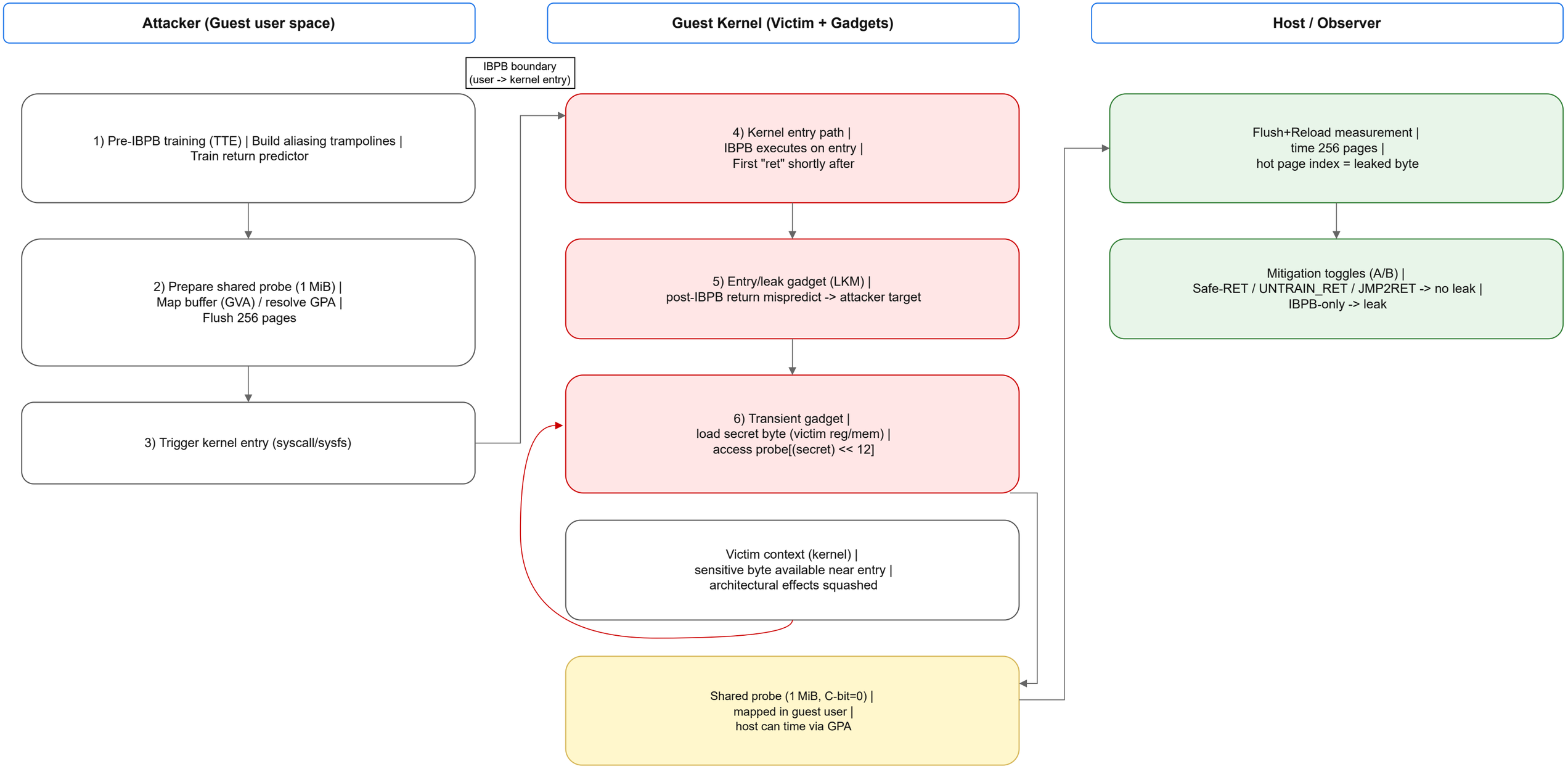
Attacker (Guest user space)
Guest Kernel (Victim + Gadgets)
Host / Observer
IBPB boundary
(user -> kernel entry)
1) Pre-IBPB training (TTE) | Build aliasing trampolines |
Train return predictor
2) Prepare shared probe (1 MiB) |
Map buffer (GVA) / resolve GPA |
Flush 256 pages
3) Trigger kernel entry (syscall/sysfs)
4) Kernel entry path |
IBPB executes on entry |
First "ret" shortly after
5) Entry/leak gadget (LKM) |
post-IBPB return mispredict -> attacker target
6) Transient gadget |
load secret byte (victim reg/mem) |
access probe[(secret) << 12]
Victim context (kernel) |
sensitive byte available near entry |
architectural effects squashed
Shared probe (1 MiB, C-bit=0) |
mapped in guest user |
host can time via GPA
Flush+Reload measurement |
time 256 pages |
hot page index = leaked byte
Mitigation toggles (A/B) |
Safe-RET / UNTRAIN_RET / JMP2RET -> no leak |
IBPB-only -> leak