Projects Seeking Students
UNIX Kernel Compartmentalization

- Soon ("months" time scale?), CheriBSD kernel will build in pure capability mode.
- This is a good start, but probably lots (all?) of the kernel can still get at globals containing very powerful capabilities.
- Use existing and/or new tracing tools, code review, whatever to find where we should draw compartment boundaries, draw them, measure the performance impact, and estimate the security benefits.

Approved for public release; distribution is unlimited. This research is sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL), under contract FA8750-10-C-0237. The views, opinions, and/or findings contained in this article/presentation are those of the author(s)/presenter(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

Projects Seeking Students
Anti-Spectre

- We have some thoughts about CHERI lessening the impact of Spectre.
  - Don't speculate tagged values: bounds available at time of dereference.
  - Augment MMU ASID with "compartment ID."
- Are these enough? What don't they fix or mitigate? How expensive are they in practice?

Projects Seeking Students
Coupling CHERI and Information Flow

- CHERI does object bounds really quite well.
- It does not do things like information flow.
- Try mixing an information flow machine (e.g., PUMP) with CHERI; does it get the best of both? Do the two sides become simpler or smaller or faster as a result of the union?

Projects Seeking Students
Software Optimization Given CHERI

- A whole lot of C uses pointer-and-length pairs. Could we automatically find those? Could we replace them with single capabilities?
- CHERI could plausibly have room for "user-defined capabilities". Are there cases where you'd want such things?
- Revisit `NaN` packing. Unlike integer-pointer architectures, CHERI can't stuff an address inside an IEEE-754 double-sized box, but the reverse is certainly possible on CHERI. How much does it help? Would the sudden vast increase in available metadata bits for values (but fewer for pointers) be useful?

- CHERI capabilities can be sealed with an object type as part of controlled non-monotonicity.
  - Only unseal by control transfer or with an authorizing capability.
- These bits live in capability metadata, and there just are not *that many* bits to be had.
- How many do we really need? Is there a lightweight virtualization strategy? Can we move them out to memory somehow?