# Enforcement techniques

# A wide variety

- Processor privilege levels
- Virtual memory
- Capabilities
- Shielded execution (SGX, TZ, …)
- Trusted computing (TPM)
- Tagged hardware
- Hardware support for
  - Bounds checking (MPX)
  - Exploit mitigation (CFI, NX, …)
- Cryptography
- Type checking
- Static analysis
- Program verification
- Language design
- Run-time monitoring
- Taint tracking
- Program rewriting
- Inlined reference monitors

**Question**: where are enforcement techniques lacking?

# **Question**: where are enforcement techniques lacking?

- Protection against (micro-architectural) side-channels?
  - There are very heuristic compiler based countermeasures
  - More principled software based countermeasures for crypto code
  - Are more principled approaches possible?

# **Question**: where are enforcement techniques lacking?

- Protection against (micro-architectural) side-channels?
  - There are very heuristic compiler based countermeasures
  - More principled software based countermeasures for crypto code
  - Are more principled approaches possible?
- Enforcing liveness properties? (Deepak's talk)
  - Interrupts /scheduling?

# **Question**: Should source code pass additional security information to the compiler?

- (Andrew's talk:) tag-based security monitors at C-level?

# **Question**: Should source code pass additional security information to the compiler?

- (Andrew's talk:) tag-based security monitors at C-level?
- Are source level abstraction or information hiding constructs useful as indications of protection domains?

# **Question**: Should source code pass additional security information to the compiler?

- (Andrew's talk:) tag-based security monitors at C-level?
- Are source level abstraction or information hiding constructs useful as indications of protection domains?
- Do we need some kind of intermediate representation for platform-level enforcement techniques?
  - If so, what would this look like?
  - At LLVM IR level? (Address spaces?)
  - (Perhaps Typed LLVM IR can play a role here?)

# **Question**: Expressivity vs complexity vs performance in enforcement mechanisms

- E.g. Capabilities versus virtual memory
  - Why would capabilities succeed now? (The idea is decades old)
  - What else do we need? (Dominique's talk)
    - Linear capabilities? Attestation?
  - Somebody (Pramod?) mentioned "time-limited" capabilities yesterday?
  - Scalability of capability systems (Pramod's talk today)?

# **Question**: Expressivity vs complexity vs performance in enforcement mechanisms

- E.g. Capabilities versus virtual memory
  - Why would capabilities succeed now? (The idea is decades old)
  - What else do we need? (Dominique's talk)
    - Linear capabilities? Attestation?
  - Somebody (Pramod?) mentioned "time-limited" capabilities yesterday?
  - Scalability of capability systems (Pramod's talk today)?
- For what other protection primitives do we need fast hardware support to build efficient secure compilers?
  - Efficient isolation within one address space? (Deepak/Deian)

# **Question**: Static vs dynamic enforcement mechanisms?

- At source language level, both static and dynamic approaches to enforcement are popular
- At target (low) level, there are mainly dynamic mechanisms
- (Except for some academic work: e.g. Amal's work)

# Other questions from the audience?