# Binary Analysis and Instrumentation Working Group

Jeff Hollingsworth

hollings@cs.umd.edu

Participants:

Drew Bernat, Mike Fagan, Jim Gakarowicz,
Jeff Hollingsworth, Madhavi Krishnan, Matthew Legendre,

Dyn inst

## Memory Instrumentation: Ideas for new interfaces

- Filter memory access instrumentation
  - Instrument only potentially dangerous access - eliminate safer access via static analysis
  - Use data flow analysis to filter memory instructions to only instrument global accesses
- Add more flexibly memory inst point
  - Allow instrumentation code to move inst point around a bit to generate better code
  - Might make sense for other inst. points

## ROSE and Dyninst

- New Dyninst snippet type
  - allow external code generator to be called
  - Need to develop exact interface
- New  Rose Features: Instruction semantics
  - X86 – 64 bit
  - Power PC
  - Floating point instructions
- Using Rose to discover/identify libc functions
  - Use info about system calls to do this

Dyn inst

## Complementary source and binary instrumentation

- Cooperate to tailor instrumentation
  - Source Instrumentation puts code in
  - Binary takes it out
- Use Source Inst. To pad code with no-ops
- Use source inst to trigger pragmas
  - Limit register uses
  - Pass compiler flags

## More analysis for smarter instrumentation

- Floating Point Liveness
- Discover loop bounds
- Add a section to to a binary
  - Allow information (analysis) to be stored with binary

Dyn inst