Upcoming Features in Dyninst and its Components

Bill Williams
Paradyn Project

Paradyn / Dyninst Week
College Park, Maryland
March 26-28, 2012
What’s New in Dyninst

Dyninst 7.0.1
- ProcControl, Stackwalker not reintegrated
- DataflowAPI early prototype
- Static CFG model
- No PPC64 or BlueGene rewriting
- Non-standard configure, packaging

Dyninst 8.0
- ProcControl, Stackwalker fully integrated on all platforms
- DataflowAPI includes all analyses used by Dyninst
- Dynamic CFG model
- PPC64 and BlueGene rewriting
- More standards-compliant distribution
Dyninst and the Components

= Existing Component

= Proposed

AST

SymtabAPI
Parsing API
Patch API

Instruction API
DataFlow API

Stackwalker API
ProcControl API

Code Gen

Process
ProcControlAPI: Ports

• Windows port
  • Complex things become simple (e.g. library load)
  • Simple things become complex (e.g. stopping process)

• BlueGene port
  • Everything is asynchronous
  • Everything is at large scale
ProcControlAPI: Interface Extensions

• RPCs revisited
  • Sometimes you really want a blocking operation

• Process groups
  • Iteration, once again, the bane of scalability

• Detach-on-the-fly support
DataflowAPI

- **New features**
  - Liveness analysis

- **Existing features**
  - Stack height analysis
  - Symbolic evaluation
  - Slicing

- **Application inside Dyninst**
  - Code generation
  - BG/P return instruction analysis
Dynamic CFG Applications

- Malware analysis
- Binary modification
- Dynamically loaded libraries
- Unparsed indirect control flow
Dynamic CFG

• Observe updated CFG via callbacks
  • New code
  • Modified code
  • Overwritten code

• Using modified CFG is optional
  • Instrumentation works normally
SymtabAPI and orthogonality

Old model (pre-PPC64 Linux)
- **Symtab depends on:**
  - Binary format
  - ...and address width

- **Symbols in ELF binaries point to:**
  - Functions
  - Variables

New model (including PPC64)
- **Symtab depends on:**
  - Binary format
  - Address width
  - ABI, including:
    - Architecture
    - OS

- **Symbols in ELF binaries point to:**
  - Functions
  - Variables
  - Pointers to functions
Packaging

- Separate `make, make install`
- Improved detection of dependencies
- Improved modularity
- Working with RedHat to make a standards-compliant RPM
New in MRNet 4.0

- Performance improvements
- Platform pruning
  - No more AIX, Solaris official support
- New multithreaded, lightweight back end
Questions?