### Background

- Testing prototype attacks on hardware is an expensive operation.
- To expedite the operation's penalties, software emulation can be used.
- However, prototyping attacks requires modification of the emulator's internal architecture.
- Qemu and Tsim-leon3 do not fit the requirements (the former uses dynamic translation while the other is not open source).

### Research Plan

- Plan out the internal emulator architecture.
- Develop the basic disassembler.
- Develop the main CPU pipeline emulation.
- Analyze and reverse engineer current SPARC simulators (such as the tsim-leon3) for IO devices.

### Goals

- To develop a functional SPARC emulator that is capable of simulating hardware considered important (such as I and D caches, TLB, etc.).
- To develop an organized, modular architecture so internal components can be made interchangeable.

### Research Results

- Developed a modularized 32 bit SPARC disassembler.
- Developed a modularized 32 bit Executable and Linking Format (ELF) parser.
- Developed a SPARC emulator capable of simulating most of the standard, non-coprocessor, non-floating point instructions.
- Reverse engineered a SPARC simulator to determine the locations of certain memory mapped IO devices.

### Fundamental Questions/Challenges

- Can code running on the guest operating system determine the presence of an emulator/virtual machine?
- Is it possible for code running on the guest operating system to exploit a generic vulnerability to discover the presence of malicious hardware?

### Related Work/Interaction with Other Projects

- The SPARC emulator will be used to prototype malicious hardware.