Hex-Rays Decompiler. What's new in 2009?

2009/01/29

Maintenance release of the decompiler.
We decided to release an intermediate version with all accumulated improvements and bugfixes while still continuing to work on the floating point support.

- + added block comments
- + added support for function comments
- + added command to hide cast operators from the output (hotkey is \)
- + added warnings. they can be suppressed by WARNS1 parameter in hexrays.cfg
- + derived types are saved into the database: now decompilation results should not differ between sessions
- + improved the speed of batch decompilation: better handling of decompilation queue
- + references to " r" and " s" special stack frame members are renamed to "retaddr" and "savedregs"
- + mach-o: __cls_refs and __message_refs sections are handled in a special way
- + added special handling for elf .got sections
- + added special handling for mach-o __pointers section
- + reserved words can not be used as local variable names. the function name is rejected too
- + if the user requested a bitfield for a constant, use it even if some bits have no corresponding symbolic names; these bits are printed in hex
- + added rule: *(x+y) => x[y]
- + sdk: added get_line_item() to facilitate mapping of line items
- + sdk: added hxe_text_ready and hxe_print_func events
- bugfix: expressions with pointers to arrays were suboptimal; if such types were used for function arguments, the output would be wrong
- bugfix: small (4 byte) arrays could be used as scalars in the output, which was wrong
- bugfix: if instruction bytes were patched without reanalyzing them, the decompiler could interr
- bugfix: error message for 16bit applications was wrong
- bugfix: decompiler could create fastcall functions with __int64 register arguments
- bugfix: tail call to unknown function could lead to interr
- bugfix: the decompiler could crash in some exceptional circumstances
- bugfix: fixed a typo that could lead to a crash
- bugfix: if a funcarg was also represeting the return value, the function prototype could be modified during analysis
- bugfix: could interr on a linux syscall if it was the last instruction of the function
- bugfix: if the user modified a structure field type, it would be detected by the decompiler
- bugfix: mul64 recognizer could create an interr