UVNet Universal Decompiler (uvudec)

John McMaster

January 18, 2010

Abstract

Toolsuite for analysis of executable binaries. There are several competitors to this project such as Hex-Rays/IDA and REC. Hex-Rays is very expensive and, being commercial software, does not allow strong collaborative development. REC (Reverse Engineering Compiler) supports several architectures and is of decent quality, but is closed source, so also does not allow collaborative development.

1 uvudec Executable

Describes theory behind the uvudec executable, the core of this project.

1.1 Command line summary

```
uvudec version 0.2.0.UVUDEC_VER_BUILD
Copyright 2009 John McMaster <JohnDMcMaster@gmail.com>
Portions copyright GNU (MD5 implementation)

Usage: ../bin/uvudec <args>
Args:
--verbose: verbose output.  Equivilent to --verbose=3
--verbose=<level>: set verbose level.  0 (none) - 3 (most)
--verbose-init: for selectivly debugging configuration file reading
--verbose-analysis: for selectivly debugging code analysis
--verbose-processing: for selectivly debugging code post-analysis
--verbose-printing: for selectivly debugging print routine
--config-language=<language>: default config interpreter language (plugins may require specific)
python: use Python
javascript: use javascript
--addr-min=<min>: minimum analysis address
--addr-max=<max>: maximum analysis address
--addr-exclude-min=<min>: minimum exclusion address
--addr-exclude-max=<max>: maximum exclusion address
--addr-comment: put comments on addresses
--addr-label: label addresses for jumping
--analysis-only[=<bool>]: only do analysis, don't print data
--analysis-address=<address>: only output analysis data for specified address
--flow-analysis=<type>: how to trace jump, calls
linear: start at beginning, read all instructions linearly, then find jump/calls (default)
trace: start at all vectors, analyze all segments called/branched recursivly
--opcode-usage: opcode usage count table
--analysis-dir=<dir>: create skeleton data suitible for stored analysis
--input=<file name>: source for data
--output=<file name>: output program (default: stdout)
```

```
--debug=<file name>: debug output (default: stdout)
--print-jumped-addresses=<bool>: whether to print information about jumped to addresses (*1)
--print-called-addresses=<bool>: whether to print information about called to addresses (*1)
--useless-ascii-art: append nifty ascii art headers to output files
--help: print this message and exit
--version: print version and exit

Special files: -: stdin
<bool>:
true includes case insensitive "true", non-zero numbers (ie 1)
false includes case insensitve "false", 0

*1: WARNING: currently slow, may be fixed in future releases
```

1.2 Basic usage

Example: ./uvudec embedded.bin Will print the disassembly/decompile from the image. Will try to automatically guess architecture based off of heuristics. Currently the heuristic algorithm is blazing fast at O(0) and will assume you are using 8051. This is trivial to fix, but I've been too lazy.

1.3 Advanced

2 Appendix

Misc items

2.1 Thanks

Kudos to...

- Rob Escriva for providing hosting, help with GIT, and this Latex template from our RPI HPC project
- Alex Radocea for enhanced Python support
- RCOS for the pizza n' stuff. Yummy!