radare >1.0 && <3

--pancake @ 2k8

Overview

- Stable 1.0 released in 2008-10-08
- Free book released too
- Ported to x86,mips,arm,ppc and osx,sunos,linux,bsd,w32
- Define the basics of the core and capabilities of the framework
- Lot of work has been done since now
- Needs some refactoring and love
- Extreme development thru hackatons
- Looking for users and betatesters
- Brainstorming!

Walking to 2.0...

- Find, implement and optimize common use cases

- Cleanup, refactorizing

- More flexibility

- Modular GUI

- Split up into libraries

We need...

- Users and betatesters

- Crazy ideas

Scripting

- Full turing-machine for radare scripting

- Integration for .NET and Vala

- Javascript and other dynamic langs will come fast

- Integration with Parrot (perl6vm) adding support for a large list of languages (perl6, php, ruby, python, scheme, TCL, Javascript, lolcode, ...)

- Interactive exo-scripting for python and so on

Projects

- Seamless static vs debug environments
- Autoreloading and multi-host syncronization of metadata
- Compress them using Gzip
- Enhace import/export of metadata
- Store/restore debugger process state from the core

Search engine

- Backward (reverse) search support
- Enhaced search+replace and file-based finds
- Added pluggable algorithms for finding arch-specific function definitions
- Signature generation and matching
- Full encoding support (using libiconv)

Plugins

- Redesign and refactor the plugin interface

- Define the core-plugin communication API

- Modularize plugin loading by using descriptor plugins exporting capabilities but not functionalities

- Enables faster loading

- Allows to use radare as a library

Console

- Speedup console access
- Optional ncurses frontend
- Fully replace readline with dietline
- Better MVC design

GUI

- Pluggable GTK widgets implemented as radare plugins

- Disassembler, Hexdump gtk widget

- Standalone frontends for rahash, rasm ...

- Native thread-based frontend for the Core

- Ubuntu/w32/OSX users need it

- We need an icon too :) designers and proposals are welcome!

Graphs

- Also for data structures
- Ease the interface and add more interaction
- Multi-color trace graphs
- Integration with radiff for graphical code analysis diffing
- Ease export/import/manipulation of graphs
- Group nodes by local-var usage
- Better layout algorithms

Disassembler

- Full multiarch pseudo decompilation (pas.c)

- Funroll display mode

- Speed up and refactorize of the disassembler modules

- Scripting based extensions for the disassembler to implement unknown instructions or full new architectures in python or so

- Nested structure viewer/manipulation

- Interactive assembler and opcode manipulation

- Applied IRA, GStreamer and UNIX concepts to decompile natively using a pipeline and a set of single-task modules

Injection

- Add more shellcode snippets for injection
- Define API for code injection for static and dynamic environments
- Enhacements for the native assembler
- Use IRA concepts to crosscompile assembly to multiple architectures

Debugger

- Fix breakpoints and extend them for tracing facilities
- Better threading support
- Reduce {arch,os,dbg}-specific code
- Native DWARF support
- !lib !call !fork !int helpers for code injection
- Replay trace executions (chronicles integration?)

FPU

- Native floating point support in the core
- Pseudocode and pseudodecompilation for floating point opcodes
- Emulation support
- Debugger integration and extend support for non-x86 architectures

Random ideas

- Emulate filesystems
- Kernel debugger (User mode linux)
- Better uber-cpu debugger (bochs-python)
- Integration with ERESI user and kernel debugger
- gst-inject for GStreamer introspection and code injection
- ...

Cya!

(Sorry, this time there's no hurting image :)

Questions? Ideas? Feedback? Patches? Beer?

http://radare.nopcode.org/