Radare2 + Frida

Better Together

/by @trufae + @oleavr

What is radare2?

- Advanced free/open/libre hexadecimal editor with disassembler, debugger, ..
- Multi-platform, multi-architecture, works on any POSIX system and Windows
- Provides libraries, apis, bindings and scripting to use all the features
- Command-line interface (with visual and embedded web server interfaces)
- Each module can be extended with plugins
- r2pipe is the recommended way to script r2 from ANY language
- Easy to integrate with existing tools

Bonus

- Cutter is the official graphical user interface
- r2pm is the package manager

What is Frida?

- Dynamic instrumentation toolkit
  - Debug live processes
- Scriptable
  - Execute your own debug scripts inside another process
- Multi-platform
  - Windows, macOS, Linux, iOS, Android, QNX
- Highly modular, JavaScript is optional
- Open Source

What is in common?

- OpenSource
- Focus on reverse engineering
- Runs on many operating systems and architectures
- Able to read and write memory
- Support for debugger features
- Able to disassemble code
- Search/inspect memory

We can observe both tools have things in common, but some are solved better in r2 and others in frida.

In addition the authors of both tools work in the same company (NowSecure)

- So why not merge them?

r2frida!

- Better Together!

References

Frida

- 10 yo project by Ole Andre
- https://twitter.com/oleavr
- https://frida.re
- https://github.com/frida/frida
- #frida
- https://t.me/fridadotre

Radare2

- 13yo project by pancake
- https://rada.re
- https://twitter.com/trufae
- https://github.com/radare/radare2
- #radare
- https://t.me/radare

NowSecure:
https://www.nowsecure.com/

r2con

- Barcelona first week of September
- 3rd edition
- ~200 ppl
- ~50€ ticket
- 4 day conference
- 2 day training + 2 day talks

r2frida

How to install it?

- r2pm -i r2frida

How to use it?

- r2 frida://

Attach to any local process or remote frida-server via USB or TCP.

- Pid
- Spawn Path
- Host:Port
- USBID/pid

* No frida required (it’s self contained inside the plugin)

IO is the base

r2frida is implemented as an IO plugin, so the basic feature is the ability to read and write memory from the target process.

- open/close o,o-
- read/write x,w

We can load Frida information as flags into r2

- Maps \dm
- Symbols, Imports \is \ii
- Class, Methods \ic

r2frida Commands

- r2 io plugins have a cmd interface
- Those can be accessed with the =! Or \ prefix.
- !!! allows to register autocompletions
- \? For help

Evaluating Code inside the Agent

We can run js code in the target process.

- `\ console.log(123)`

To load a local script into Frida-agent.

- `\. Test.js`

Many languages transpile to Javascript:

- `Typescript`

r2frida plugins

- Add more commands (accessible via \ or =! prefix)
- Hook the IO (read, write operations)
- Run r2 commands from the agent-side
- Use the Javascript Frida API

How to register them?

- `\. File.js` --- run script
- if the script calls to r2frida.pluginRegister()
- `\.-name` --- to unregister

Calling r2 from Frida

R2Frida exposes the r2pipe interface into the Frida Javascript API, so you can call r2 commands from frida.

- As long as Frida can call-back r2, we can make Frida script create flags, rename functions, add backtraces, draw graphs in r2 after tracing code.

DEMO TIME

DEMO (1)

- while(1) { print(“%d\n”, value); sleep(1); }
- Attach with r2frida
- Modify the assembly code by hand in realtime

DEMO (2)

- Load gnuboy.js r2frida plugin
- Reading in-process memory
- Works in gameboy/qemu/androidemulator/vbox
- Using the additional ‘gb’ command you can toggle the gb IO instead of the plain process IO
- Show the source of the plugin

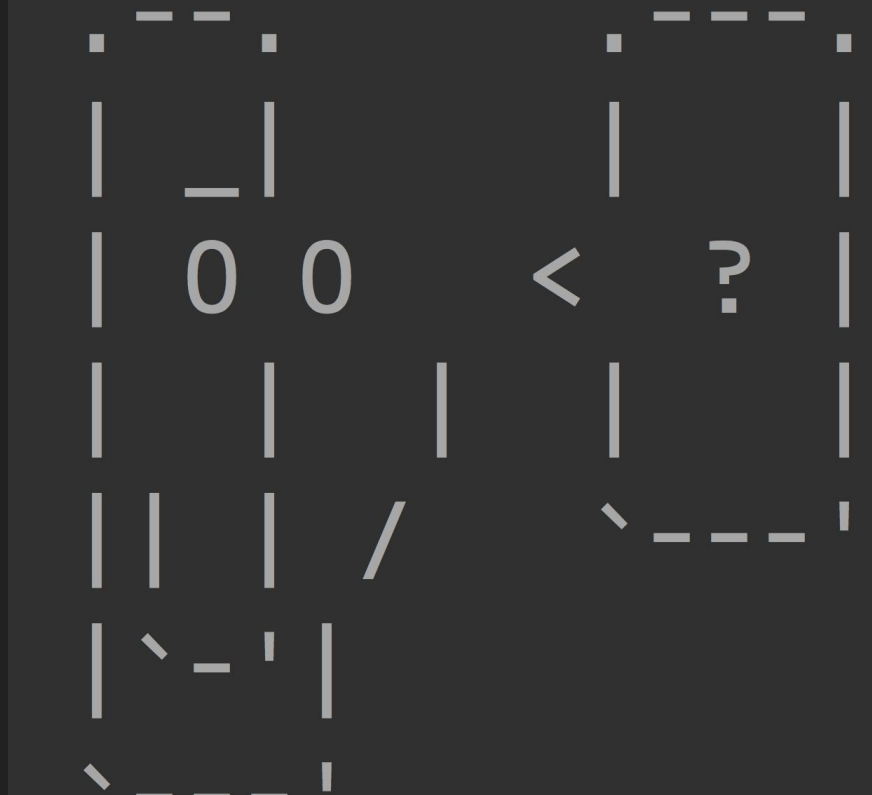
DEMO (3)

Hack the Fox game

- Loading scripts to run code in a remote iOS app:
  - Scan the heap to find the Objective-C object representing the game character
  - Put it on fire by calling an instance method
  - Scan the heap to find the game state object
  - Set number of flowers collected by calling an instance a method. Win the game!

Questions?

or gtfo



Thanks for Watching