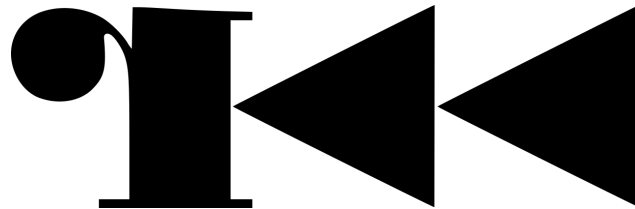


**CUTTER**

A graphical user interface for radare2 - by Antide Petit

# whoami

@xarkes\_



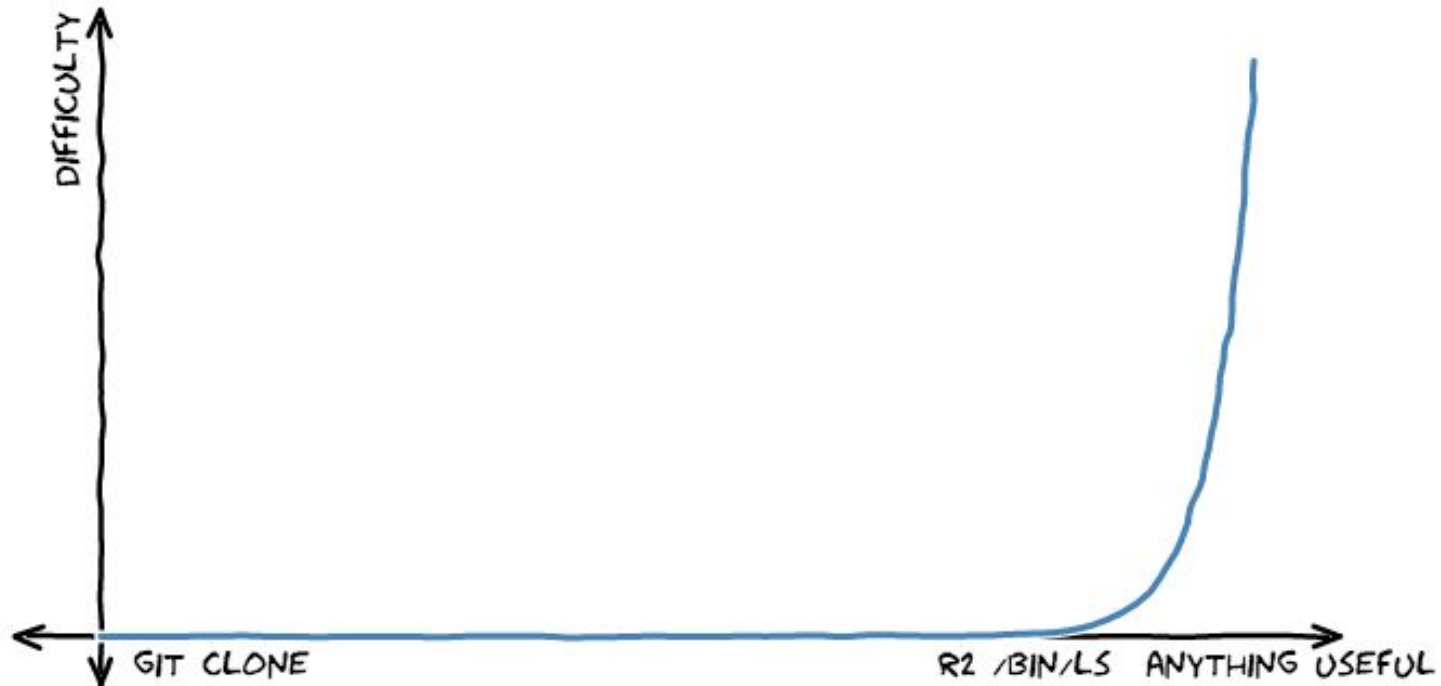
# Origins of Cutter

- Iaito
- Qt/C++
- Developed for a while by only one person (Hugo Teso)
- Took a few years to become open source
- After it became open source, no more maintenance



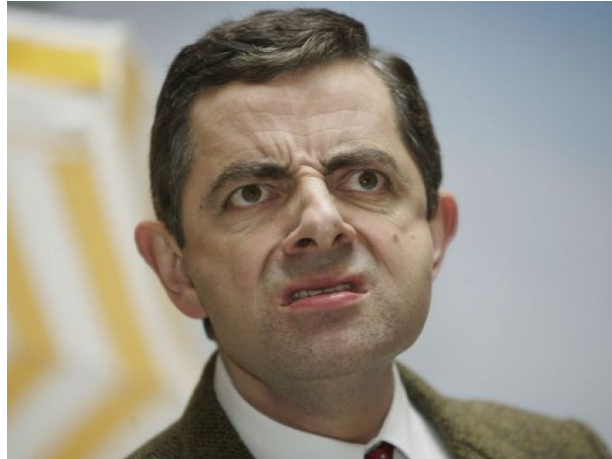
*“Cutter is not aimed at existing radare2 users. It instead focuses on those whose are not yet radare2 users because of the learning curve, because they don't like CLI applications or because of the difficulty/instability of radare2.”*

# R2 LEARNING CURVE



# Old code...

- Many useless or unusable features (from a Reverse Engineer point of view)
- Displayed graphs with HTML (Qt WebEngine)



# ... to better code

- Big refactoring
- Removed HTML graphs
- Added a lot of new features



# What is it like?

The screenshot displays the Cutter debugger interface, which is used for analyzing and debugging binaries. The main window is divided into several panes:

- Functions:** A list of functions found in the binary, with `decryption_function` selected.
- Graph (decryption\_function):** A control flow graph showing the execution flow of the selected function. It includes nodes for variable declarations, stack frame setup, and various instructions like `movsx`, `cmp`, and `jmp`.
- Disassembly:** A pane showing the assembly code corresponding to the graph. It includes comments and hex addresses for each instruction.
- Console:** A pane at the bottom left showing the output of the debugger, displaying `Welcome to Cutter!`.
- Sections:** A pane at the bottom right showing a table of sections and a pie chart representing the binary's structure.

**Sections Table:**

Name	Size	Address	EndAddress	Entropy
.data	8704	0x0041b000	0x0041d200	3.28480039
.rdata	30208	0x00413000	0x0041a600	5.08320213
.reloc	5120	0x00432000	0x00433400	6.47993944
.rsrc	78336	0x0041e000	0x00431200	7.86195980

**Function registers info:**

- A: esp ebp of sf zf pf cf eax eip ecx
- I: esp ebp eip dx
- N: dx
- R: esp ebp eax eip ecx edx of sf cl

**X-Refs to current address:**

Address	Instruction
0x004012d2	<code>movsx edx, word [arg_ch]</code>
0x004012d6	<code>cmp dword [local_4h], edx</code>
0x004012d9	<code>jge 0x4012f5</code>



# Features

- Basic reverse engineering widgets
  - Disassembly
  - Graph
  - Hexdump
  - Sections/Functions/Strings

# Features

- Analysis options
- Jupyter notebook (python scripting)

# Demo



# Internal structure

- (Almost) Everything is a QDockWidget
  - Can be moved around
  - Can be docked
  - Can be windowed
  - => Very flexible
- (Almost) Everything uses radare2 json output
  - The commands are more stable than the current API
  - Json is easy to parse

# Projects (TODO)

- Move reverse widgets to an external library
  - Will help maintenance with x64dbg
  - Will give the possibility for anyone to use the existing rather than reinventing the wheel
- Collaborative platform (dependant on r2)

# Projects (WIP)

- Python plugins
- Plugin manager
- More asynchronous tasks
- Debugger support

# GSoC

- [@filipe\\_casal](#)
- Debugging platform



Google  
Summer of Code

# How to get it?

- Releases are published on GitHub
- <https://github.com/radareorg/cutter/releases>



# Documentation?

- Contribution guidelines are available in the repo
- No complete documentation (the software moves fast)
- Feel free to ask help on IRC or Telegram (#cutter on Freenode)

# Can I use it?

- Some people use it (already) in their daily job
- Follow [@megabeets\\_](#) tutorials



# Conclusion

- A lot of improvements in a year
- A lot of remaining work
- Is used in real world reverse engineering

**Thank you!**



@radareorg  
@r2gui

<https://github.com/radareorg/cutter>