

FRAPL

Next Generation Reverse Engineering Framework

Alex Hude

Max Bazaliy

October 22-23, 2016



Who we are

Alex Hude

- Melbourne, Australia
- Blackmagic Design
- Hardware, XNU
- Fried Apple team

Max Bazaliy

- Kyiv, Ukraine
- Lookout
- XNU, Linux, LLVM
- Fried Apple team

Modern Reverse Engineering

Static approach

- Disassemblers
- Code analyzers
- Decompilers
- IDA as a choice

Dynamic approach

- Code instrumentation
- Dynamic analyzers
- Debuggers
- Frida as a choice

Static analysis challenges

- Missed context (CPU registers, stack, memory)
- Hard to follow code execution flow (obfuscation)
- Hard to follow data flow (encryption)
- Hard to follow indirect function calls

Debugging challenges

- Anti debugging tricks
- Data loss during restarts
- Execution flow may be changed under debugging
- No way to hook/replace existing code easily

Dynamic instrumentation challenges

- Code disassembly still missed
- High learning curve
- Usually requires to write a lot of code
- Hard to maintain multiple things at a time

A close-up photograph of a woman with long dark hair, her eyes closed and her hand covering her face in a gesture of distress or frustration. The background is a plain, light-colored wall.

I NEED TO WRITE CODE

FOR MY FRIDA HOOKS



What is FRAPL ?

FRAPL

=

Frida scripts + FridaLink

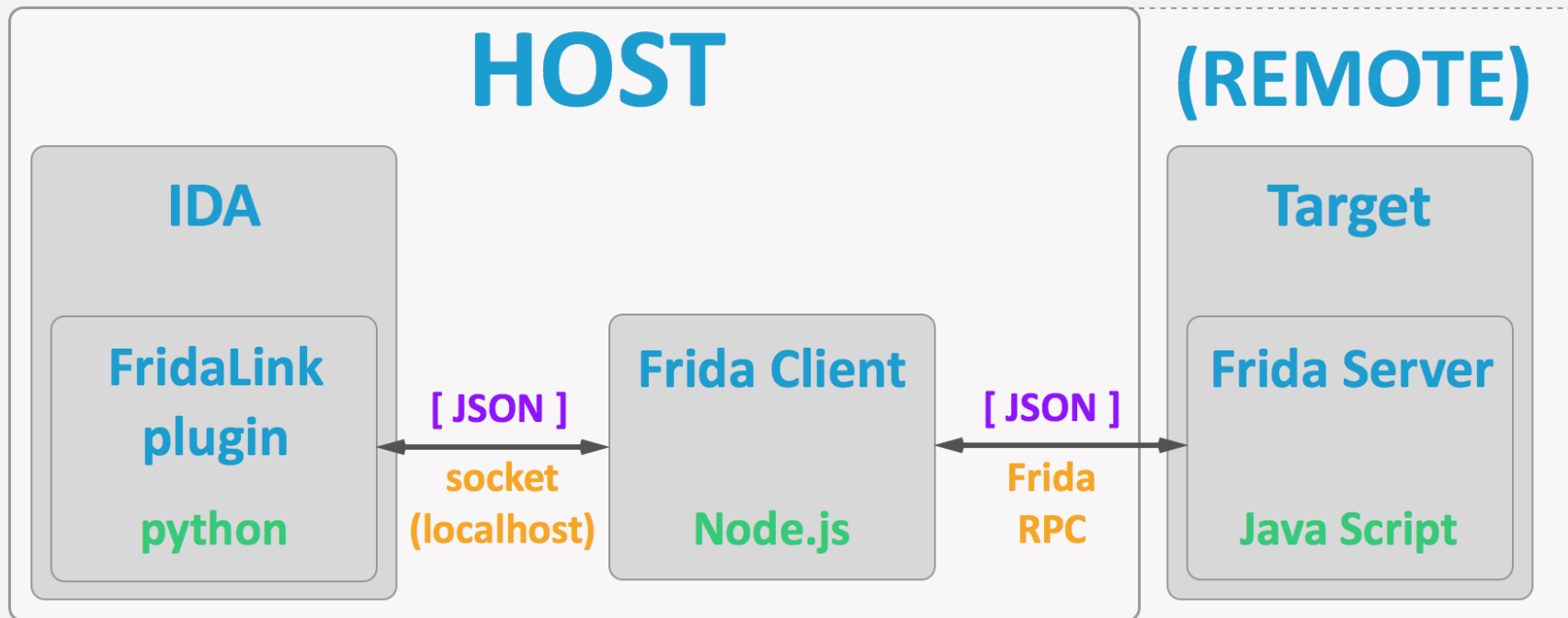
Frida Scripts

- Node.js client (attach, spawn, RPC, script loading)
- Node.js server script (RPC, GCD, iOS/macOS bindings)
- Common operations wrappers (objc hooks etc)
- Utility functions (memory dumps, logging)

FridaLink

- IDA plugin that implements UI controls to Frida
- Socket protocol between IDA & Frida Client (JSON)
- RPC protocol for between Frida Client Server (JSON)
- FridaLink.js (Frida script)

FridaLink architecture



FridaLink goals

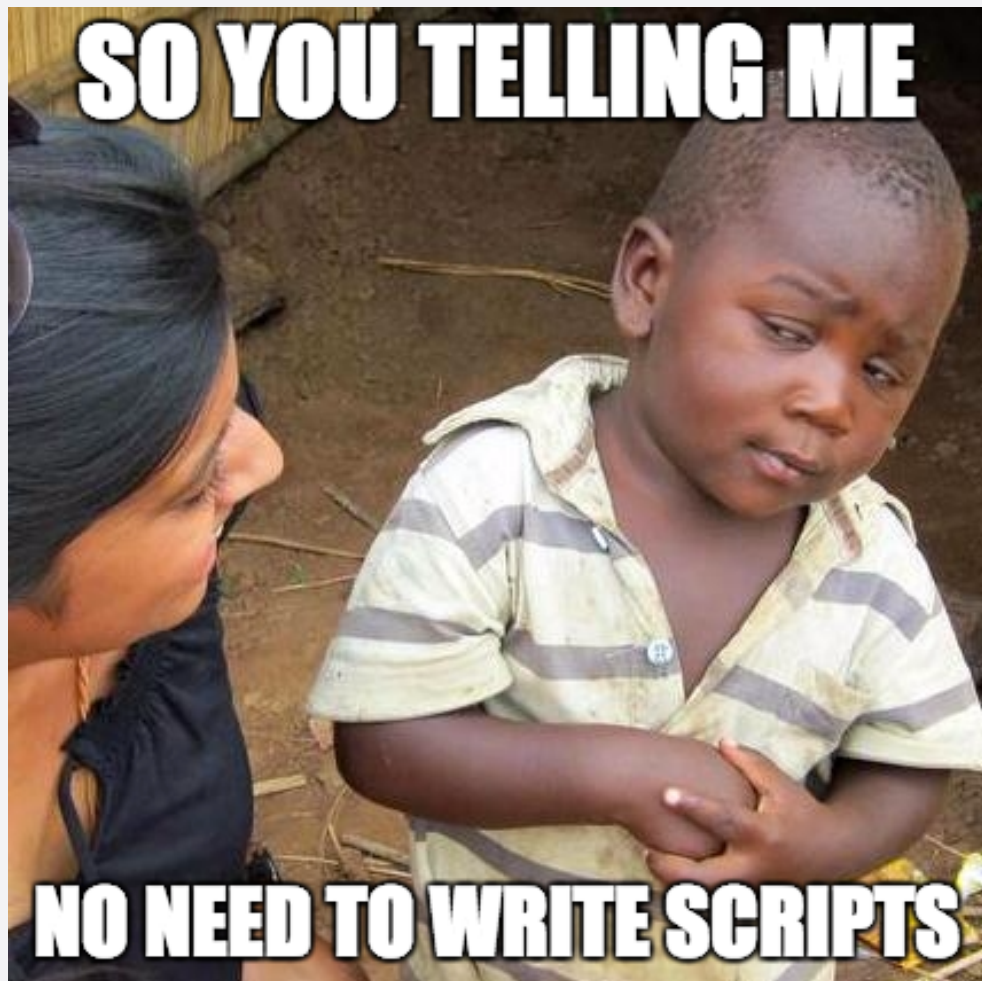
- Bring static analysis info from IDA to Frida
- Use dynamic info from Frida for IDA analysis
- Monitor runtime state directly from IDA
- Control Frida agent directly from IDA

FridaLink features

- Direct function hooks made easy
- Function replacement made easy
- Module loading made easy
- Custom scripts support

FridaLink features

- CPU context monitoring
- Memory monitoring
- SQLite database support
- Helpers and project save/restore



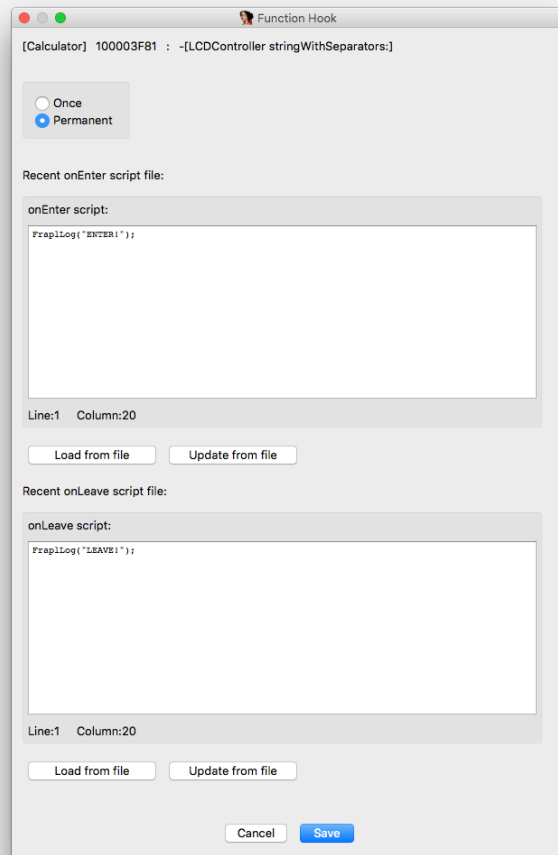
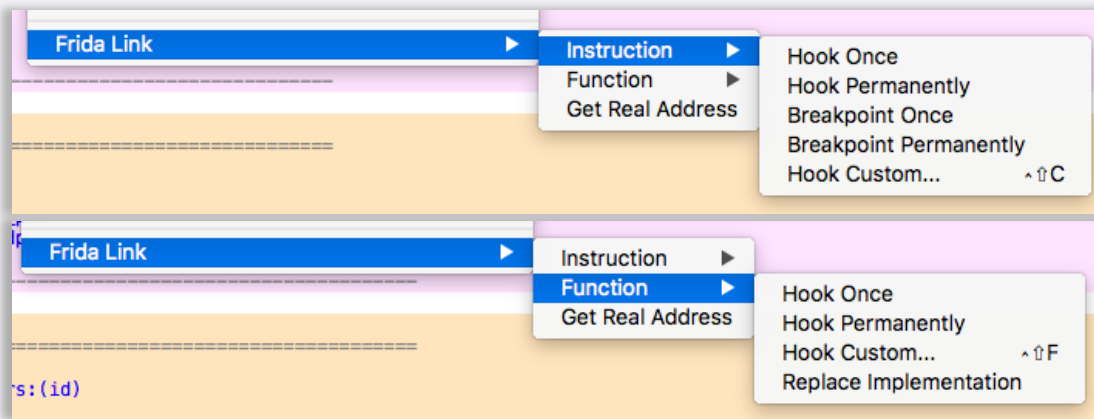
FridaLink - Overall View

The screenshot displays the FridaLink application interface, which is designed to look like a standard macOS desktop. The main window is titled "IDA View-1" and shows the assembly code of the "Calculator" application. The code is color-coded and includes comments in Chinese. A "Find crypt v2" window is open at the top right, showing the "Frida Link" option. A "Load Project" menu is visible on the far right, listing various actions like "Load Module", "Fetch Target Modules", and "Show Import Hooks". A calculator window is overlaid on the assembly code, displaying the number "42". The bottom of the interface features a "FRAPL Log" window showing the application's startup sequence, including the installation of function hooks and the replacement of the "showMemoryIndicator" function. The overall layout is clean and professional, with a focus on providing a comprehensive view of the application's internal structure and the FridaLink integration.

17

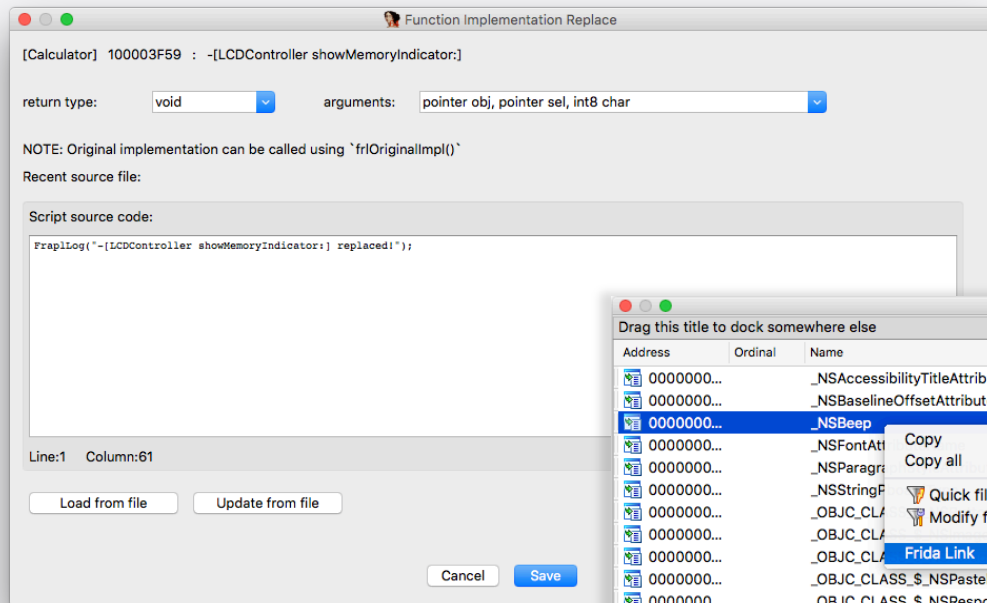
FridaLink – Hooks

- Instruction hooks
- Instruction breakpoints (hook with wait)
- IDB (local) function hooks
- Import function hooks

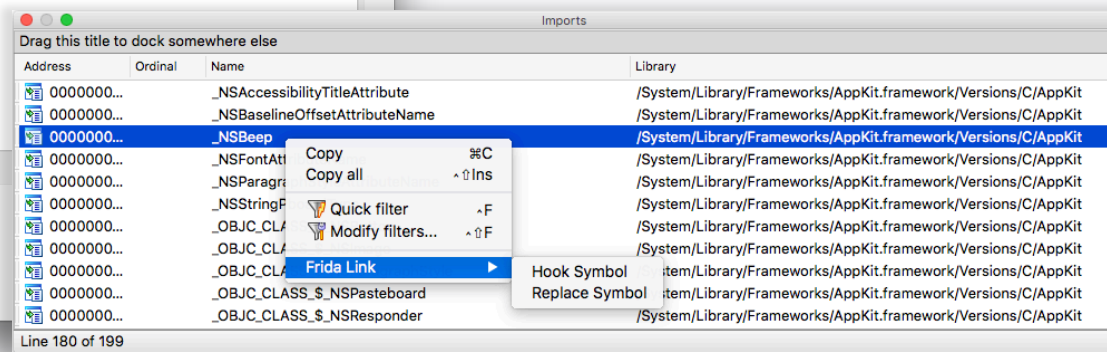


FridaLink – Function Replacement

Replace local function

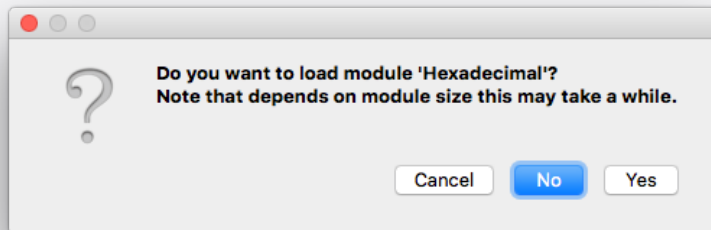


Replace Import function



FridaLink – Module Loading

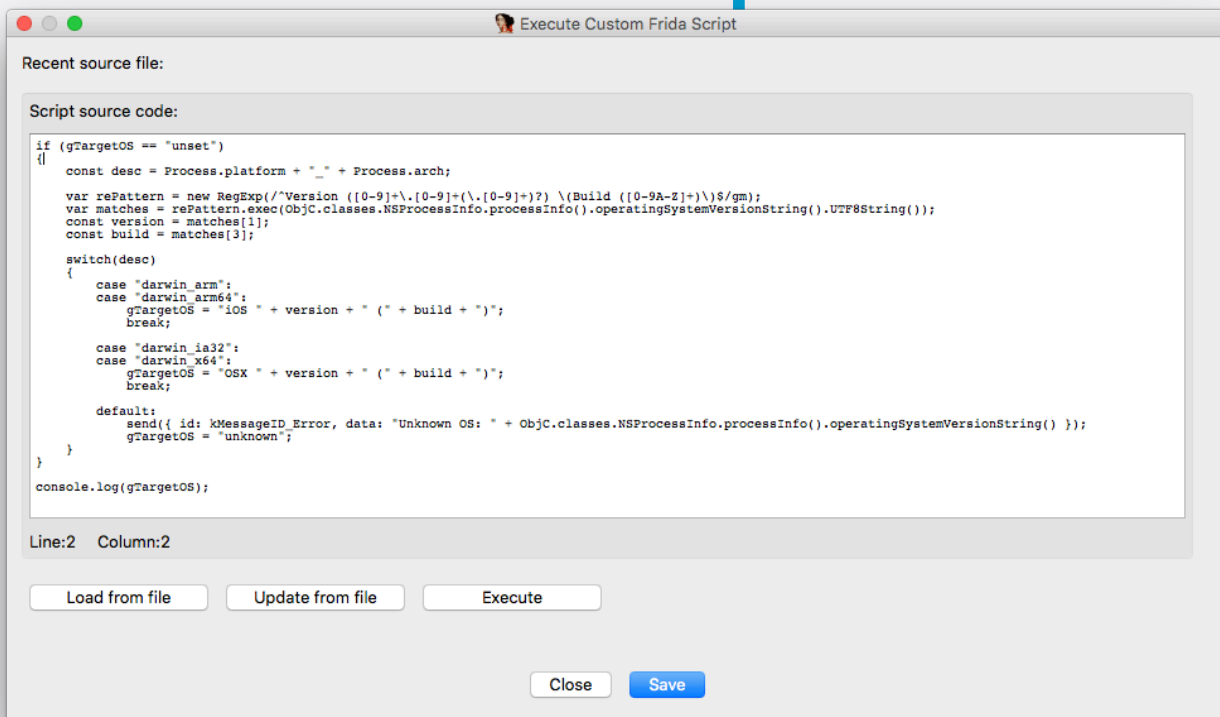
- Automatic (on backtrace)
- Manual



Module	Base	Path	Size
Calculator	0x10dfe1000	/Users/Alex/Projects/Calculator/Calculator.app/Contents/MacOS/Calculator	86016 (0x15000)
Cocoa	0x7fff9ea95000	/System/Library/Frameworks/Cocoa.framework/Versions/A/Cocoa	4096 (0x1000)
SpeechDictionary	0x10e008000	/System/Library/PrivateFrameworks/SpeechDictionary.framework/Versions/A/SpeechDictionary	569344 (0x8B000)
SpeechObjects	0x10e0cd000	/System/Library/PrivateFrameworks/SpeechObjects.framework/Versions/A/SpeechObjects	139264 (0x22000)
Calculate	0x7fff9117c000	/System/Library/PrivateFrameworks/Calculate.framework/Versions/A/Calculate	77824 (0x13000)
ApplicationServices	0x7fff93d77000	/System/Library/Frameworks/ApplicationServices.framework/Versions/A/ApplicationServices	4096 (0x1000)
QuartzCore	0x7fff8f1ae000	/System/Library/Frameworks/QuartzCore.framework/Versions/A/QuartzCore	1896448 (0x1CF000)
Foundation	0x7fff9c67d000	/System/Library/Frameworks/Foundation.framework/Versions/C/Foundation	3493888 (0x355000)
libobjc.A.dylib	0x7fff962ed000	/usr/lib/libobjc.A.dylib	3588096 (0x36C000)
libSystem.B.dylib	0x7fff9ccc4000	/usr/lib/libSystem.B.dylib	8192 (0x2000)

Line 1 of 223

FridaLink – Custom Scripts



Execute custom script dialog

FridaLink – CPU Context Monitoring

CPU context

Stack

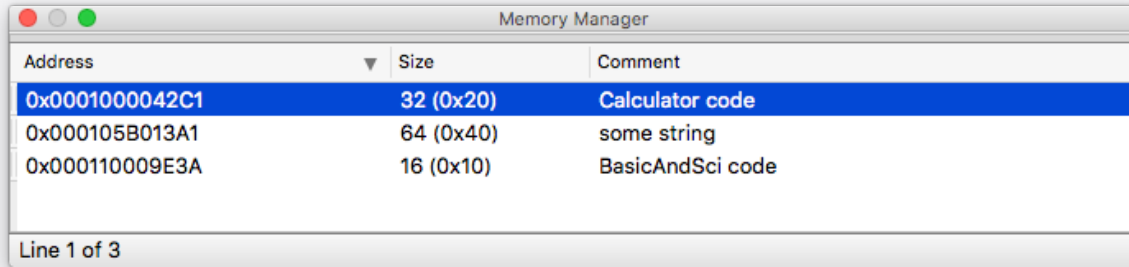
```
Stack for [ 0x100003F92: mov r14, rdx ]
00007FFFA10DFB8 0000000105AF4F92 Calculator:--[LCDController stringWithSeparators:]
00007FFFA10DFC0 0000000000000028
00007FFFA10DFC8 00007FFFB7D6D4C0 libobjc.A.dylib:objc_msgSend
00007FFFA10DFD0 03002EF113DE5F42
00007FFFA10DFD8 00007FB36BA1BEA0
00007FFFA10DFE0 0000000105B0D718 Calculator:___objc2_class <offset _OBJC_METACLASS_$_LCDController, \
00007FFFA10DFE8 00007FB36E8EFA10
00007FFFA10DFF0 0000000000000000
00007FFFA10DFF8 0000000000000002
00007FFFA10E000 00007FB36BA1BEA0
00007FFFA10E008 0000000105B013A1 Calculator:db 'stringWithSeparators:',0
00007FFFA10E010 00007FFFA10E090
00007FFFA10E018 00007FFFA10E058
00007FFFA10E020 0000000000000098
00007FFFA10E028 00007FB36BA1C0A0
00007FFFA10E030 03002EF113DE5F42
00007FFFA10E038 0000000000000000
00007FFFA10E040 0000000000000286
00007FFFA10E048 00007FB36E857800
00007FFFA10E050 0000000105B013A1 Calculator:db 'stringWithSeparators:',0
00007FFFA10E058 00007FFFA10E0F0
00007FFFA10E060 00007FFFA10E5A00
00007FFFA10E068 0000000000000098
00007FFFA10E070 00007FB36BA1BEA0
00007FFFA10E078 03002EF113DE5F42
00007FFFA10E080 00007FFFA10E090 libobjc.A.dylib:objc_msgSend
00007FFFA10E088 0000000000000028
00007FFFA10E090 00007FFFA10E0F0
00007FFFA10E098 000000010D5CA413
00007FFFA10E0A0 00007FFFB7D6D4C0 libobjc.A.dylib:objc_msgSend
00007FFFA10E0A8 0000000000000030
00007FFFA10E0B0 00007FF96E93AD CoreFoundation:CreateStringFromFileSystemRepresentationByAdding
00007FFFA10E0B8 00007FB36A592A30
00007FFFA10E0C0 03002EF113DE5F42
00007FFFA10E0C8 00007FB36BA1BEA0
00007FFFA10E0D0 00007FFFB7D6D4C0 libobjc.A.dylib:objc_msgSend
00007FFFA10E0D8 0000000000000030
00007FFFA10E0E0 00007FFFB7D6D4C0 libobjc.A.dylib:objc_msgSend
00007FFFA10E0E8 0000000000000028
00007FFFA10E0F0 00007FFFA10E110
00007FFFA10E0F8 0000000105AF5312 Calculator:--[LCDController setLCDStringValue:input:]
```

```
CPU Context-1
Drag this title to dock somewhere else
CPU context for [ 0x100003F92: mov r14, rdx ]
rax: 0x0 r10: 0x7fb36c003610
rbx: 0x98 r11: 0x105b0d718
rcx: 0x3002ef113de5f42 r12: 0x7fb36ba1bea0
rdx: 0x7fb36ba1c0a0 r13: 0x3002ef113de5f42
rsi: 0x105b013a1 r14: 0x7fff87d6d4c0
rdi: 0x7fb36ba1bea0 r15: 0x28
rbp: 0x7ffff5a10e090 pc: 0x105af4f92
rsp: 0x7ffff5a10e058 rip: 0x105af4f92
r8: 0x2 sp: 0x7fff5a10e058
r9: 0x0
```

Backtrace

```
Backtrace-1
Backtrace for [ 0x100003F81: --[LCDController stringWithSeparators:] ]
1: Calculator 0x000100001610 + 0x000083 --[LCDController showValue]
2: Calculator 0x0001000042C1 + 0x000051 --[LCDController setLCDStringValue:input:]
3: BasicAndSci 0x000110009E3A + 0x0000CC --[CalculatorEngine userEnteredDigit:]
4: BasicAndSci 0x000110002860 + 0x000054 --[BasicAdvancedController handleDigit:]
5: libsystem_trace.dylib 0x7FFF8604C02F + 0x00004B _os_activity_initiate
6: AppKit 0x7FFF82CFC8BD + 0x0001CC --[NSApplication sendAction:to:from:]
7: AppKit 0x7FFF82D0AF88 + 0x000056 --[NSControl sendAction:to:]
8: AppKit 0x7FFF82D0AE85 + 0x000083 _26--[NSCell _sendActionFrom:]_block_invoke
9: libsystem_trace.dylib 0x7FFF8604C02F + 0x00004B _os_activity_initiate
10: AppKit 0x7FFF82D0ADD5 + 0x000090 --[NSCell _sendActionFrom:]
11: libsystem_trace.dylib 0x7FFF8604C02F + 0x00004B _os_activity_initiate
12: AppKit 0x7FFF82D08A05 + 0x000A85 --[NSCell trackMouse:inRect:ofView:untilMouseUp:]
13: AppKit 0x7FFF82D51CE8 + 0x0002E8 --[NSButtonCell trackMouse:inRect:ofView:untilMouseUp:]
14: AppKit 0x7FFF82D07917 + 0x00029D --[NSControl mouseDown:]
15: AppKit 0x7FFF8325ABB7 + 0x0018B2 --[NSWindow _handleMouseDownEvent:isDelayedEvent:]
16: AppKit 0x7FFF8325D379 + 0x0000D4 --[NSWindow _reallySendEvent:isDelayedEvent:]
17: AppKit 0x7FFF82C9C438 + 0x000205 --[NSWindow sendEvent:]
```

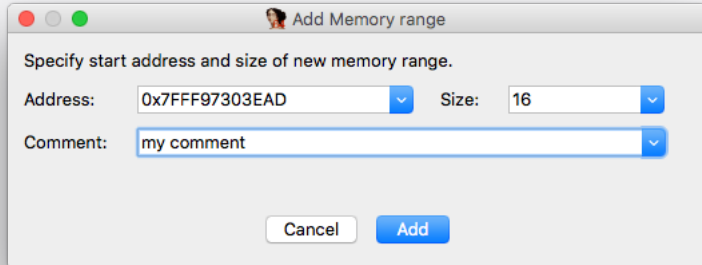
FridaLink – Memory Monitoring



Address	Size	Comment
0x0001000042C1	32 (0x20)	Calculator code
0x000105B013A1	64 (0x40)	some string
0x000110009E3A	16 (0x10)	BasicAndSci code

Line 1 of 3

Memory manger



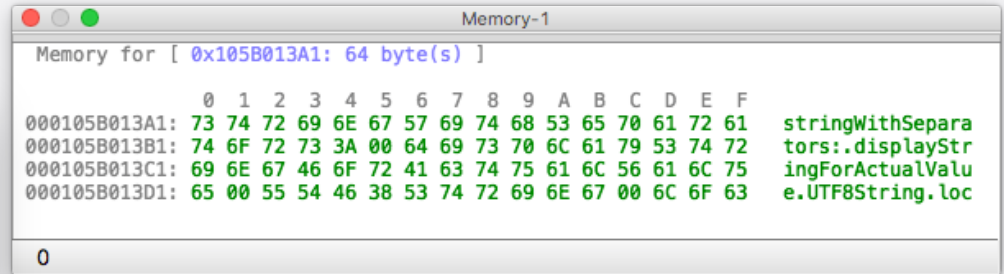
Add Memory range

Specify start address and size of new memory range.

Address: Size:

Comment:

Add new memory watchpoint



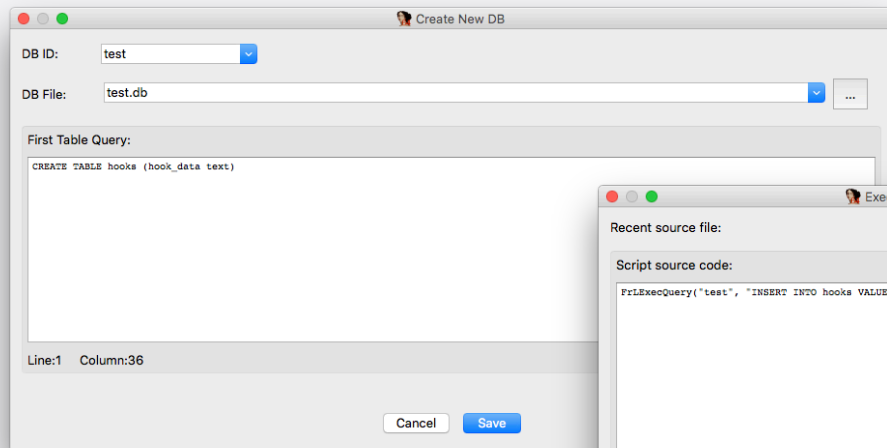
Memory for [0x105B013A1: 64 byte(s)]

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
000105B013A1:	73	74	72	69	6E	67	57	69	74	68	53	65	70	61	72	61	stringWithSepara
000105B013B1:	74	6F	72	73	3A	00	64	69	73	70	6C	61	79	53	74	72	tors:.displayStr
000105B013C1:	69	6E	67	46	6F	72	41	63	74	75	61	6C	56	61	6C	75	ingForActualValu
000105B013D1:	65	00	55	54	46	38	53	74	72	69	6E	67	00	6C	6F	63	e.UTF8String.loc

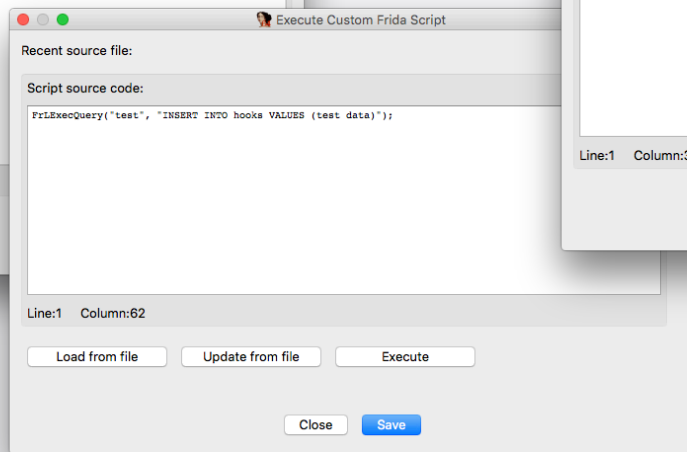
0

Memory content

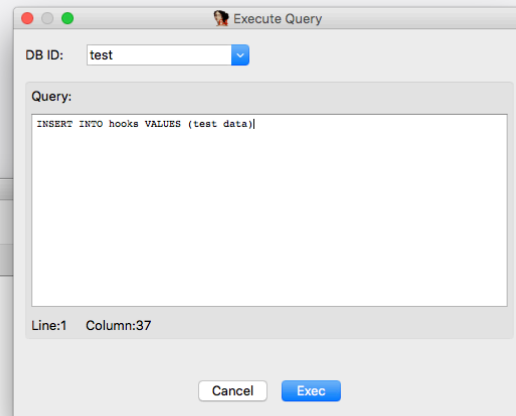
FridaLink – SQLite Support



Set up DB

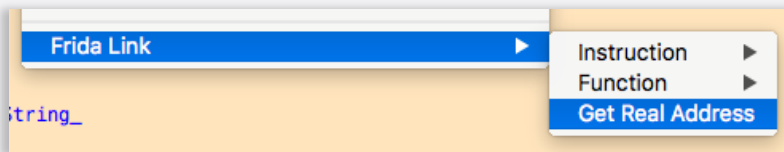
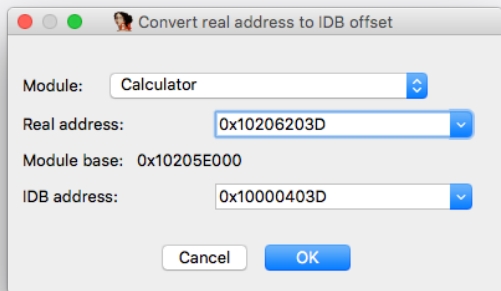


Load script



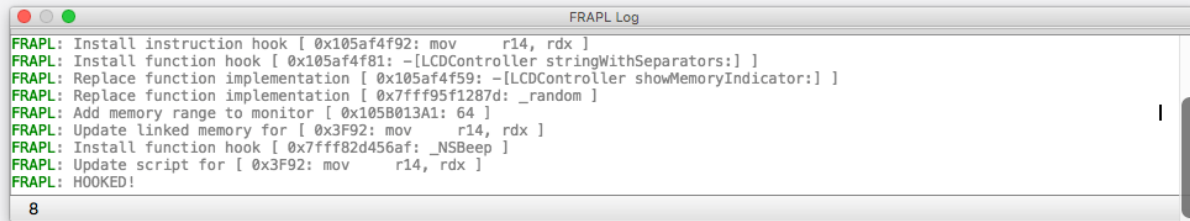
Query execution

FridaLink – Helpers and more



```
FRAPL: [ Calculator ] 0x100004043 => 0x1004B8043 cmp    rbx, rax
FRAPL: [ BasicAndSci ] 0x1100056DC => 0x107E1F6DC mov    rsi, cs:selRef_substringWithRange_0
```

Address converter



FRAPL logs

macOS Application Demo

```
ZSH Profile (iMac) (node)
~ > Projects > FRAPL ./create_project.sh -f ~/Projects/iTunes ; cd ~/Projects/iTunes
~ > Projects > iTunes node ./client.js -l -c theme_example.json -n iTunes ./server.js
FRAPL: establish FridaLink automatically
FRAPL: starting mode set to attach by name
FRAPL: target location set to local
FRAPL: bind export from FrAFridaLink.js
FRAPL: script source is loaded
FRAPL: process 'include' directives
FRAPL:   include('FRAPL/FrACommon.js')
FRAPL:   include('FRAPL/FrAServerCore.js')
FRAPL:   include('FRAPL/FrAGCD.js')
FRAPL:   include('FRAPL/FrAdlfcn.js')
FRAPL:   include('FRAPL/FrAUtils.js')
FRAPL:   include('FRAPL/FrAFridaLink.js')
FRAPL: attaching to target by name...
FRAPL: server script created
FRAPL: message listener set
FRAPL: server script loaded
FRAPL: FridaLink established
FRAPL: Module list request complete
FRAPL: Delete all memory ranges from monitor
FRAPL: Remove all FridaLink instruction hooks
FRAPL: Remove all FridaLink function hooks
```

iOS Application Demo

```
ZSH Profile (iMac) (node)
~ > Projects > FRAPL > ./create_project.sh -f ~/Projects/iTunes ; cd ~/Projects/iTunes
~ > Projects > iTunes > node ./client.js -l -c theme_example.json -r -p $(frida-ps -U | grep Calculator | awk '{print $1}') ./server.js
FRAPL: establish FridaLink automatically
FRAPL: starting mode set to attach by PID
FRAPL: target location set to remote
FRAPL: bind export from FrAFridaLink.js
FRAPL: script source is loaded
FRAPL: process 'include' directives
FRAPL:   include('FRAPL/FrCommon.js')
FRAPL:   include('FRAPL/FrServerCore.js')
FRAPL:   include('FRAPL/FrAGCD.js')
FRAPL:   include('FRAPL/FrAdlfcn.js')
FRAPL:   include('FRAPL/FrUtils.js')
FRAPL:   include('FRAPL/FrAFridaLink.js')
FRAPL: attaching to target by PID...
FRAPL: server script created
FRAPL: message listener set
FRAPL: server script loaded
FRAPL: FridaLink established
FRAPL: Module list request complete
FRAPL: Delete all memory ranges from monitor
FRAPL: Remove all FridaLink instruction hooks
FRAPL: Remove all FridaLink function hooks
```



WEN ETA RELES ???

eta son

Future plans

- Kernel support
- Windows support ?
- Android support ?
- Hack the planet!

@getorix

@mbazaliy