

@HANKBAO

零信（原瀑布IM）

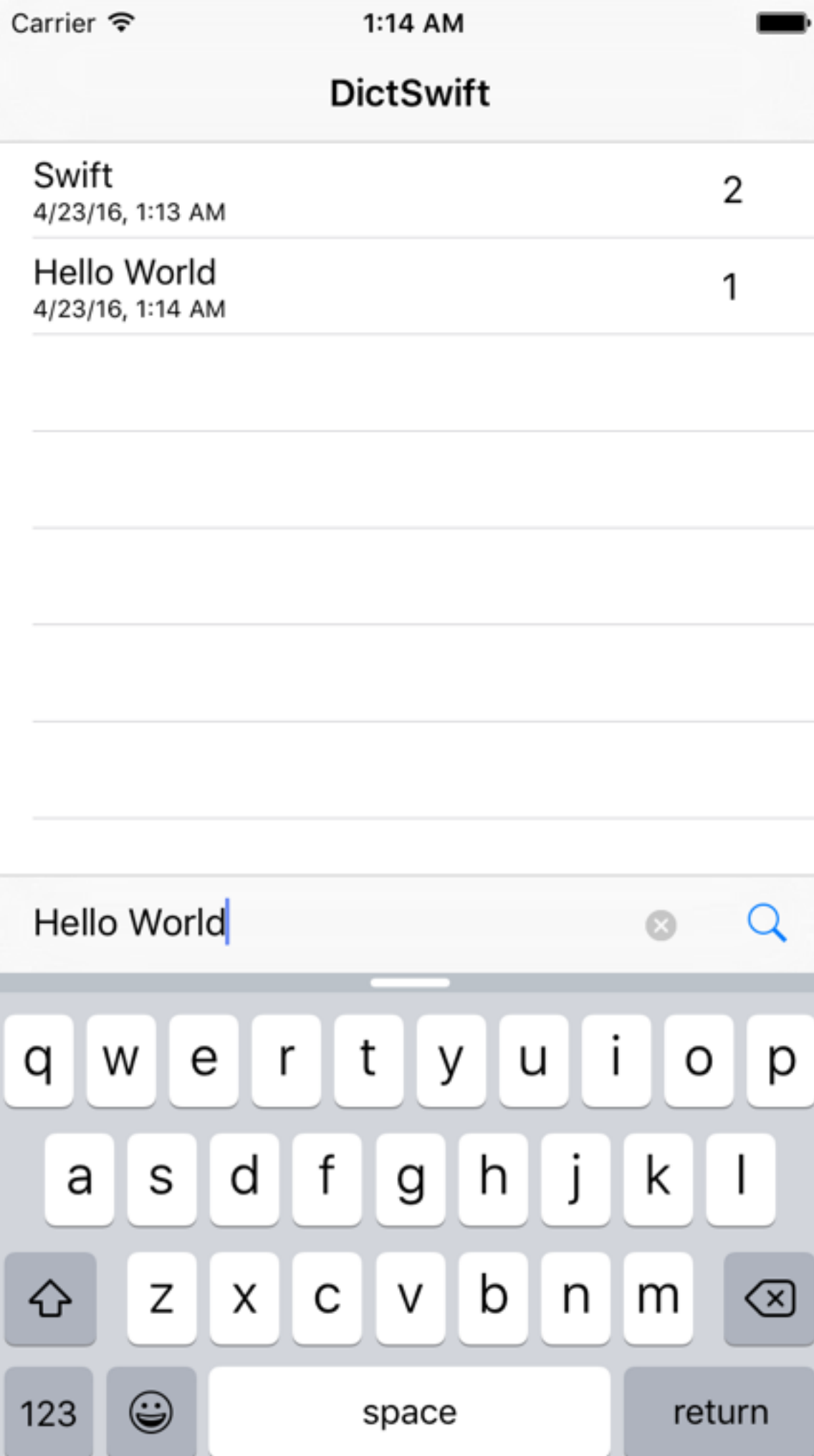
SWIFT 逆向简介

逆向工程的目标

- ▶ 学习
 - ▶ 设计
 - ▶ 实现
 - ▶ 算法
- ▶ 除错
- ▶ 扩展
 - ▶ 插件

APP 插件

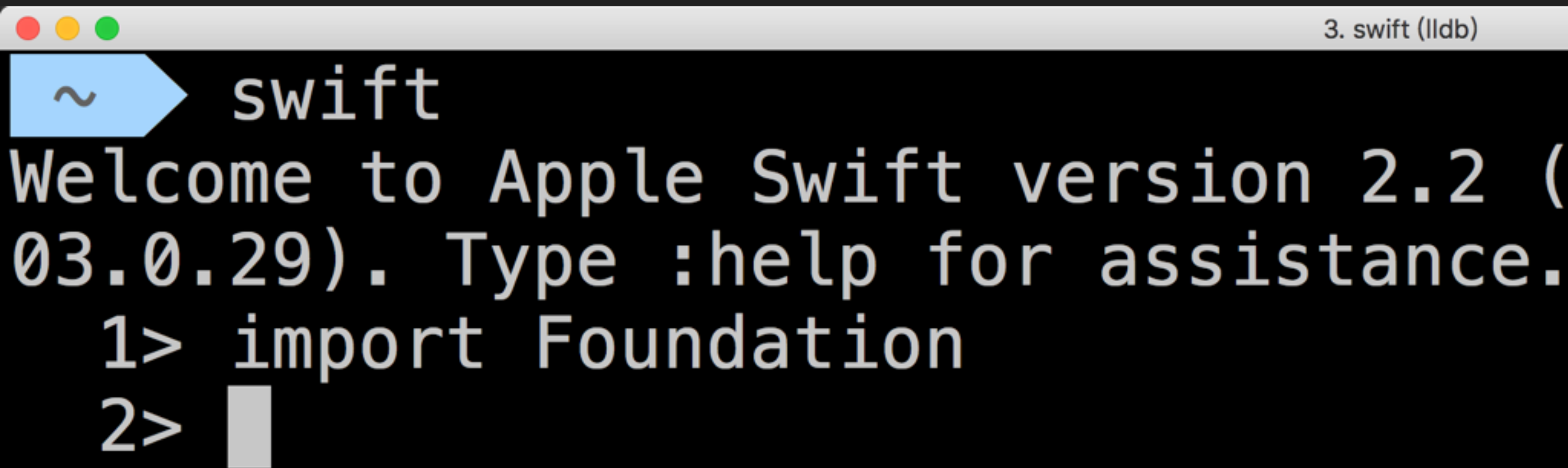




X86_64 调用约定

- ▶ arg1: \$rdi
- ▶ arg2: \$rsi
- ▶ arg3: \$rdx
- ▶ arg4: \$rcx
- ▶ arg5: \$r8
- ▶ arg6: \$r9
- ▶ ret1: \$rax
- ▶ ret2: \$rdx
- ▶ indirect: \$rax (== \$rdi)

SWIFT REPL

A terminal window with a title bar containing three colored circles (red, yellow, green) on the left and the text "3. swift (lldb)" on the right. The terminal content shows a blue prompt character "~" followed by the word "swift". Below this is a welcome message for Apple Swift version 2.2 (03.0.29). The user has entered two commands: "1> import Foundation" and "2>" followed by a cursor.

```
3. swift (lldb)  
~ swift  
Welcome to Apple Swift version 2.2 (03.0.29). Type :help for assistance.  
1> import Foundation  
2> █
```

DEMO: STRUCT

```
11 struct QueryRecord {  
12     let term: String  
13     let date: NSDate  
14     var queryCount: Int  
15  
16     init(term: String) {  
17         self.term = term  
18         date = NSDate()  
19         queryCount = 1  
20     }  
21 }
```

```
13> sizeof(QueryRecord)  
$R0: Int = 40  
14> sizeof(String)  
$R1: Int = 24  
15> sizeof(NSDate)  
$R2: Int = 8  
16> sizeof(Int)  
$R3: Int = 8
```

DEMO

```
3> sizeof(String)
$R3: Int = 24
4> :type lookup String
struct String {
  init()
  init(_ _core: Swift._StringCore)
  var _core: Swift._StringCore
}
```

```
2> sizeof(_StringCore)
$R1: Int = 24
3> :type lookup _StringCore
struct _StringCore {
  var _baseAddress: Swift.COpaquePointer
  var _countAndFlags: Swift.UInt
  var _owner: AnyObject?
  init(baseAddress: Swift.COpaquePointer
```


SWIFT-DEMANGLE

```
__TFV9DictSwift1 1 QueryRecordCfT4termSS_S0_  
__TFV9DictSwift1 1 QueryRecordg4termSS  
__TFV9DictSwift1 1 QueryRecordg4dateCSO6NSDate  
__TFV9DictSwift1 1 QueryRecordg10queryCountSi  
__TFV9DictSwift1 1 QueryRecords10queryCountSi  
__TFV9DictSwift1 1 QueryRecordm10queryCountSi
```

\$ xcrun swift-demangle

__TFV9DictSwift1 1 QueryRecordg4termSS

__TFV9DictSwift1 1 QueryRecordg4termSS --->

DictSwift.QueryRecord.term.getter : Swift.String

HOPPER DEMANGLE

- ▶ <https://github.com/keith/hopper-swift-demangle>
- ▶ <https://github.com/Januzellij/hopperscripts>

DEMO

```
                                ; DictSwift.QueryRecord.term.getter : Swift.String
__TFV9DictSwift11QueryRecordg4termSS:
000000001000031a0      push      rbp
000000001000031a1      mov       rbp, rsp
000000001000031a4      push      r15
000000001000031a6      push      r14
000000001000031a8      push      rbx
000000001000031a9      push      rax
000000001000031aa      mov       r15, qword [ds:rdi]
000000001000031ad      mov       r14, qword [ds:rdi+8]
000000001000031b1      mov       rbx, qword [ds:rdi+0x10]
000000001000031b5      mov       rdi, rbx                                ; argument "
000000001000031b8      call     imp___stubs__swift_unknownRetain
000000001000031bd      mov       rax, r15
000000001000031c0      mov       rdx, r14
000000001000031c3      mov       rcx, rbx
000000001000031c6      add       rsp, 0x8
000000001000031ca      pop       rbx
000000001000031cb      pop       r14
000000001000031cd      pop       r15
000000001000031cf      pop       rbp
000000001000031d0      ret
                                ; endp
```

SWIFT NATIVE 调用约定

- ▶ arg1: \$rdi
- ▶ arg2: \$rsi
- ▶ arg3: \$rdx
- ▶ arg4: \$rcx
- ▶ arg5: \$r8
- ▶ arg6: \$r9
- ▶ ret1: \$rax
- ▶ ret2: \$rdx
- ▶ ret3: \$rcx
- ▶ indirect: \$rax (== \$rdi)

DEMO

```
10
11 protocol QueryURLConvertible {
12     var zt_queryURL: NSURL? { get }
13 }
14
15 extension String: QueryURLConvertible {
16     var zt_queryURL: NSURL? {
```

DEMO

```
11> var qurl: QueryURLConvertible = ""  
qurl: String = ""  
12> sizeof(QueryURLConvertible)  
$R0: Int = 40  
13> sizeofValue(qurl)  
$R1: Int = 40
```

DEMO

```
18> func addrOf<T>(inout v: T) {  
19.     withUnsafePointer(&v) { print($0) }  
20. }  
21> addrOf(&curl)  
0x00000001004fcf40  
  
22> :x/5xg 0x00000001004fcf40  
0x1004fcf40: 0x0000000101c00800 0x0000000000000000  
0x1004fcf50: 0x0000000000000000 0x00000001002724c8  
0x1004fcf60: 0x00000001004fc180
```


DEMO

```
22> :ima lookup -a 0x0000000101c00800
    Address: $__lldb_expr11[0x0000000101c00800] ($__lldb_expr11.
    __cstring + 0)
    Summary:
22> :ima lookup -a 0x00000001002724c8
    Address: libswiftCore.dylib[0x00000000002674c8] (libswiftCo
re.dylib.__DATA.__const + 81656)
    Summary: libswiftCore.dylib`type metadata for Swift.String
22> :x/xg 0x00000001004fc180
0x1004fc180: 0x00000001004fb8e0
22> :x/i 0x00000001004fb8e0
    0x1004fb8e0: 55  pushq  %rbp
22> :ima lookup -a 0x00000001004fb8e0
    Address: $__lldb_expr7[0x00000001004fb8e0] ($__lldb_expr7.
    _text + 160)
    Summary: $__lldb_expr7`protocol witness for __lldb_expr_4.Q
    ueryURLConvertible.zt_queryURL.getter : Swift.Optional<__ObjC.NSU
    RL> in conformance Swift.String : __lldb_expr_4.QueryURLConvertib
    le in __lldb_expr_6 at repl6.swift
```


DEMO

```
90 private func showTerm(convertible: QueryURLConvertible) {
91     guard let url = convertible.zt_queryURL else { return }
92
93     let libViewController =
94         SFSafariViewController(URL: url, entersReaderIfAvailable: true)
95     presentViewController(libViewController, animated: true, completion: nil)
96 }
97
```

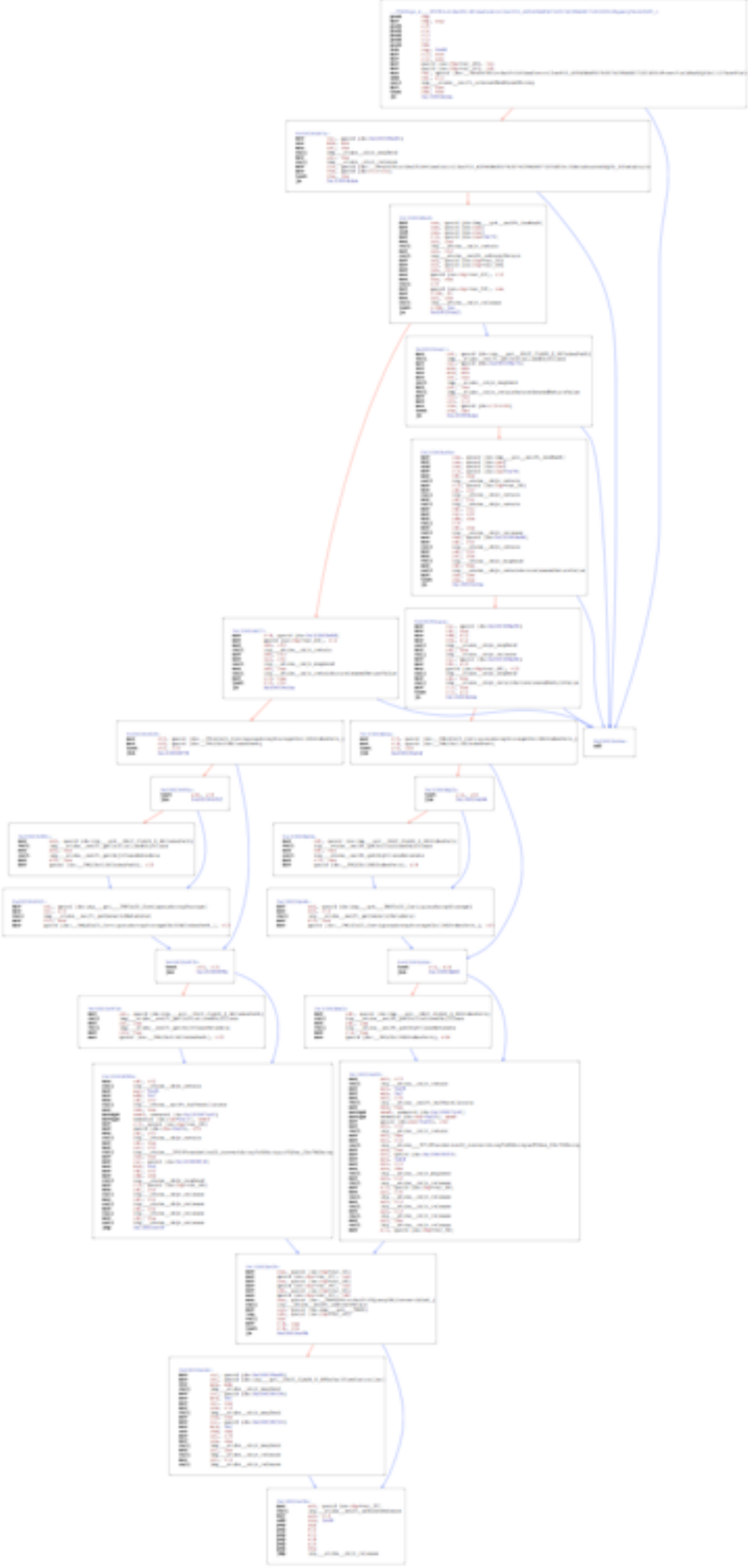
00000000100004c09	mov	rax, qword [ss:rbp+var_50]	; XREF=__TTSf4g
00000000100004c0d	mov	qword [ss:rbp+var_40], rax	
00000000100004c11	mov	rax, qword [ss:rbp+var_48]	
00000000100004c15	mov	qword [ss:rbp+var_38], rax	
00000000100004c19	mov	rdi, qword [ss:rbp+var_60]	; argument "ins
00000000100004c1d	mov	qword [ss:rbp+var_30], rdi	
		; protocol witness table for	
		; Swift.String : DictSwift.QueryURLConvertible	
00000000100004c21	mov	rbx, qword [ds:__TWPSS9DictSwift19QueryURLConvertibleS_]	
00000000100004c28	call	imp___stubs__swift_unknownRetain	
00000000100004c2d	mov	rsi, qword [ds:imp___got___TMSS]	
00000000100004c34	lea	rdi, qword [ss:rbp+var_40]	; argument #1 f
00000000100004c38	call	rbx	; __TTWSS9DictS
00000000100004c3a	mov	r14, rax	
00000000100004c3d	test	r14, r14	
00000000100004c40	je	0x100004c9b	

DEMO

```
12 class ViewController: UITableViewController {
13
14     private var toolbar: UIToolbar!
15     private weak var textField: UITextField!
16
17     private var dataSource: DataSource!
```

```
98     private func queryTerm(term: String) {
99         textField.text = nil
100
101         let (indexPath, isNewTerm) = dataSource.addTerm(term)
102         if (isNewTerm) {
103             tableView.insertRowsAtIndexPaths([indexPath], withRowAnimation: .Bottom)
104         } else {
105             let to = NSIndexPath(forRow: 0, inSection: 0)
106             dataSource.moveTermAtIndexPath(indexPath, toIndexPath: to)
107             tableView.moveRowAtIndexPath(indexPath, toIndexPath: to)
108             tableView.reloadRowsAtIndexPaths([to], withRowAnimation: .Automatic)
109         }
110
111         showTerm(term)
112     }
```

DEMO



DEMO

98
99
100

```
private func queryTerm(term: String) {  
    textField.text = nil  
}
```

```
                                ; function signature specialization  
                                ; <Arg[0] = Owned To Guaranteed and Exploded>  
                                ; DictSwift.ViewController.(queryTerm) (Swift.String) -> ()  
                                ;  
                                ; {rdi, rsi, rdx} = term: String  
                                ; rcx = self  
__TTSf4gs_n____TFC9DictSwift14ViewControllerP33_A554EBA85D7A3574C09A68073201E51  
00000000100004840      push    rbp                                ; XREF=-[_TtC9DictS  
00000000100004841      mov     rbp, rsp  
00000000100004844      push    r15  
00000000100004846      push    r14  
00000000100004848      push    r13  
0000000010000484a      push    r12  
0000000010000484c      push    rbx  
0000000010000484d      sub     rsp, 0x48  
00000000100004851      mov     r13, rcx  
00000000100004854      mov     r12, rdx  
00000000100004857      mov     qword [ss:rbp+var_48], rsi  
0000000010000485b      mov     qword [ss:rbp+var_50], rdi  
                                ; direct field offset for  
                                ; DictSwift.ViewController.(textField) : weak __ObjC.UITextF  
0000000010000485f      mov     rdi, qword [ds:__TWvdvC9DictSwift14ViewControllerP33_A554EBA85D  
00000000100004866      add     rdi, r13  
00000000100004869      call    imp___stubs__swift_unknownWeakLoadStrong  
0000000010000486e      mov     rbx, rax  
00000000100004871      test    rbx, rbx  
00000000100004874      je      0x100004cba
```

00000000100004cba

ud2

DEMO

```
101 let (indexPath, isNewTerm) = dataSource.addTerm(term)
102 if (isNewTerm) {
```

```
                                ; direct field offset for
                                ; DictSwift.ViewController(dataSource) : DictSwift.DataSource!
0000000100004893 mov r14, qword [ds:__TWvdvC9DictSwift14ViewControllerP33_A554EBA85D7A357
000000010000489a mov rbx, qword [ds:r13+r14]
000000010000489f test rbx, rbx
00000001000048a2 je 0x100004cba

-----
00000001000048a8 mov rax, qword [ds:imp__got__swift_isaMask]
00000001000048af mov rax, qword [ds:rax]
00000001000048b2 and rax, qword [ds:rbx]
00000001000048b5 mov r15, qword [ds:rax+0x70]
00000001000048b9 mov rdi, rbx ; argument "instance" for
00000001000048bc call imp__stubs__objc_retain
00000001000048c1 mov rdi, r12 ; argument "instance" for
00000001000048c4 call imp__stubs__swift_unknownRetain
00000001000048c9 mov rdi, qword [ss:rbp+var_50]
                                ; mov qword [ss:rbp+var_48], rsi
                                ; {var_50, var_48, r12} == term
00000001000048cd mov rsi, qword [ss:rbp+var_48]
00000001000048d1 mov rdx, r12
00000001000048d4 mov qword [ss:rbp+var_60], r12
00000001000048d8 mov rcx, rbx
00000001000048db call r15
00000001000048de mov qword [ss:rbp+var_58], rax
00000001000048e2 mov r15b, dl
00000001000048e5 mov rdi, rbx ; argument "instance" for
00000001000048e8 call imp__stubs__objc_release
00000001000048ed test r15b, 0x1
00000001000048f1 je 0x100004a21
```

逆向工程理论基础

- ▶ C / C++ / Objective-C / Swift
- ▶ Assembly (x86, x86_64, arm / thumb, arm64)
- ▶ 平台 ABI / 语言特定 ABI
- ▶ 编译器优化
- ▶ 操作系统

逆向工程方法和工具

▶ 静态分析

- ▶ Hopper Disassembler
- ▶ IDA Pro
- ▶ otool
- ▶ class-dump

▶ 动态调试

- ▶ lldb / gdb
- ▶ F-Script
- ▶ cycrypt

NEW TOY: VOLTRON

► <https://github.com/snare/voltron>

```
2. lldb /Applications/Slack.app (lldb)

[code]
libsystem_kernel.dylib`mach_msg:
-> 0x7fff9a72f3b7 <+50>: je     0x7fff9a72f46f      ; <+243>
0x7fff9a72f3bd <+65>: mov     edx, r15d
0x7fff9a72f3c0 <+68>: mov     ecx, dword ptr [rbp + 0x10]
0x7fff9a72f3c3 <+71>: mov     r15d, ebx
0x7fff9a72f3c6 <+74>: mov     esi, dword ptr [rbp - 0x2c]
0x7fff9a72f3c9 <+77>: test    sil, 0x40
0x7fff9a72f3cd <+81>: jne     0x7fff9a72f415      ; <+153>
0x7fff9a72f3cf <+83>: cmp     eax, 0x10000007
0x7fff9a72f3d4 <+88>: mov     esi, ecx
0x7fff9a72f3d6 <+90>: mov     r8d, edx
0x7fff9a72f3d9 <+93>: mov     ecx, r12d
0x7fff9a72f3dc <+96>: mov     r12d, r13d
0x7fff9a72f3df <+99>: mov     rbx, r14
0x7fff9a72f3e2 <+102>: mov     edx, dword ptr [rbp - 0x30]
0x7fff9a72f3e5 <+105>: jne     0x7fff9a72f420      ; <+164>
0x7fff9a72f3e7 <+107>: mov     dword ptr [rsp], esi
0x7fff9a72f3ea <+110>: mov     rdi, rbx
0x7fff9a72f3ed <+113>: mov     esi, edx
0x7fff9a72f3ef <+115>: mov     edx, r12d
0x7fff9a72f3f2 <+118>: mov     r14d, ecx
0x7fff9a72f3f5 <+121>: mov     r9d, r15d
0x7fff9a72f3f8 <+124>: mov     r13d, r8d
0x7fff9a72f3fb <+127>: call    0x7fff9a72ff60      ; mach_msg_trap

-> 0x7fff9a72f3b5 <+57>: test    eax, eax
0x7fff9a72f3b7 <+59>: je     0x7fff9a72f46f      ; <+243>
0x7fff9a72f3bd <+65>: mov     edx, r15d
0x7fff9a72f3c0 <+68>: mov     ecx, dword ptr [rbp + 0x10]
0x7fff9a72f3c3 <+71>: mov     r15d, ebx
(lldb) b CoreFoundation`CFRunLoopRunSpecific
Breakpoint 3: where = CoreFoundation`CFRunLoopRunSpecific, address = 0x00007fff8d25edb0
(lldb) br del 2
1 breakpoint deleted; 0 breakpoint locations disabled.
(lldb) ni
Process 4237 stopped
* thread #1: tid = 0x1247ebf, 0x00007fff9a72f3b7 libsystem_kernel.dylib`mach_msg + 59, queue = 'com.apple.main-thread', stop reason = instruction step over
    frame #0: 0x00007fff9a72f3b7 libsystem_kernel.dylib`mach_msg + 59
libsystem_kernel.dylib`mach_msg:
-> 0x7fff9a72f3b7 <+59>: je     0x7fff9a72f46f      ; <+243>
0x7fff9a72f3bd <+65>: mov     edx, r15d
0x7fff9a72f3c0 <+68>: mov     ecx, dword ptr [rbp + 0x10]
0x7fff9a72f3c3 <+71>: mov     r15d, ebx
(lldb)

[breakpoints]
#3  0x00007fff8d25edb0 h:0  CoreFoundation`CFRunLoopRunSpecific

[regs:general]
[ 0 d i t s z a p c ]
[  !Jump (!z) ]
RIP: 00007fff9a72f3b7
RAX: 0000000010004005
RBX: 00000000FFFFFFFF
RBP: 00007fff5fbfe000
RSP: 00007fff5fbfe040
RDI: 00007fff5fbfe1a0
RSI: 0000000070000006
RDX: 0000000000000000
RCX: 00007fff5fbfe030
R8 : 0000000000001D00
R9 : 00000000FFFFFFFF
R10: 0000000000000C00
R11: 0000000000000206
R12: 0000000000000C00
R13: 0000000000000000
R14: 00007fff5fbfe1a0
R15: 0000000000001D00
CS: 002B DS: n/a
ES: n/a FS: 0000
GS: 0000 SS: n/a

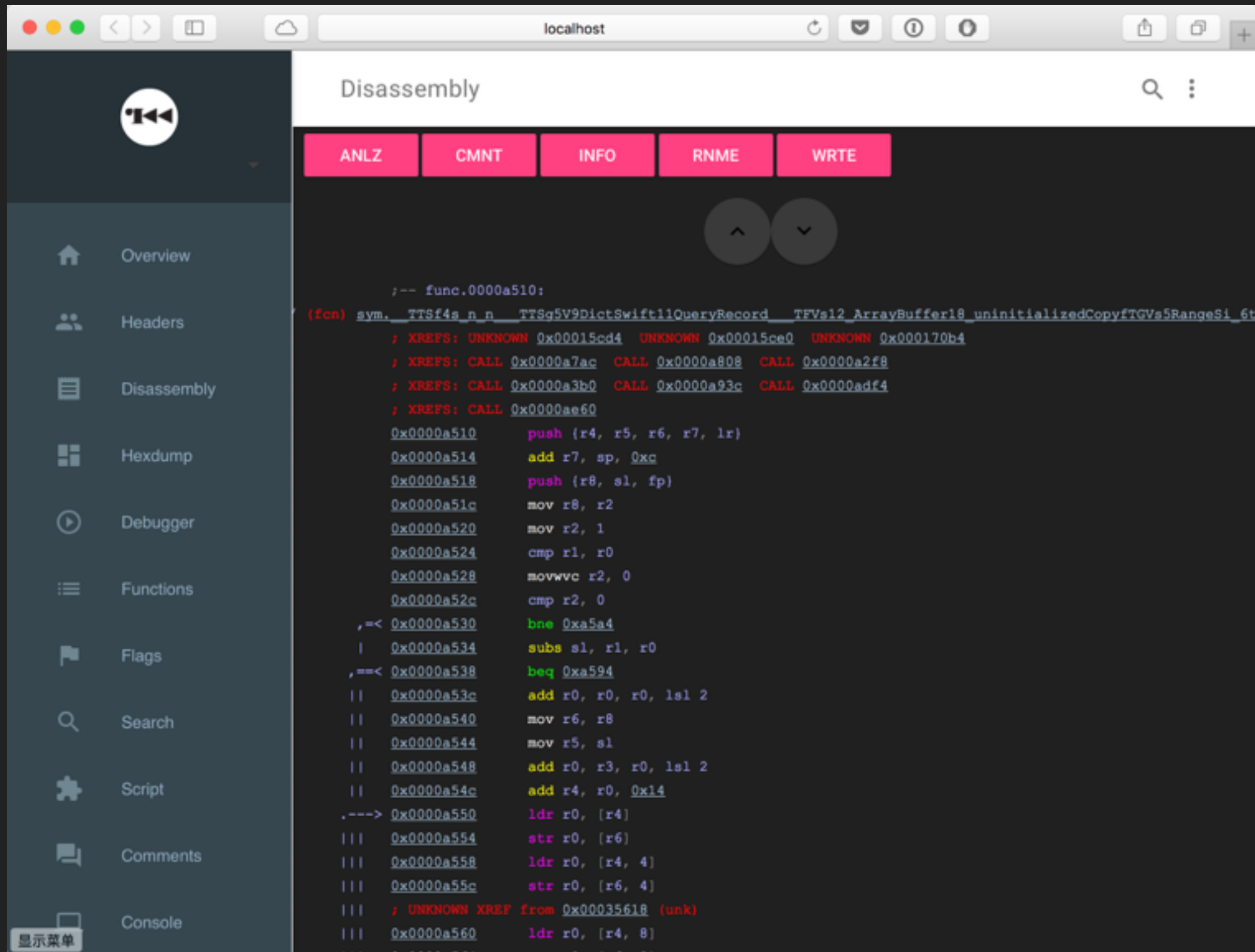
[backtrace]
* thread #1: tid = 0x1247ebf, 0x00007fff9a72f3b7 libsystem_kernel.dylib`mach_msg + 59, queue = 'com.apple.main-thread', stop reason = instruction step over
    frame #0: 0x00007fff9a72f3b7 libsystem_kernel.dylib`mach_msg + 59
    frame #1: 0x00007fff8d2601c4 CoreFoundation`__CFRunLoopServiceMachPort + 1356
    frame #2: 0x00007fff8d25f68c CoreFoundation`__CFRunLoopRun + 1356
    frame #3: 0x00007fff8d25eed8 CoreFoundation`CFRunLoopRunSpecific + 296
    frame #4: 0x00007fff914c0935 HIToolbox`RunCurrentEventLoopInMode + 235
    frame #5: 0x00007fff914c076f HIToolbox`ReceiveNextEventCommon + 432
    frame #6: 0x00007fff914c05af HIToolbox`_BlockUntilNextEventMatchingListInMode + 187
    frame #7: 0x00007fff9c63cefa AppKit`_DPSNextEvent + 1067

0x7fff5fbfe0a0: 04 E1 BF 5F FF 7F 00 00 | ...
0x7fff5fbfe090: 02 00 54 A0 FF FF FF FF | ..T....
0x7fff5fbfe080: 00 00 00 00 FF 7F 00 00 | .....
0x7fff5fbfe070: C4 01 26 8D FF 7F 00 00 | ..&.... 0x7fff8d2601c4 => '__CFRunLoopServiceMachPort + 0xd4'
0x7fff5fbfe060: F0 E0 BF 5F FF 7F 00 00 | .....
0x7fff5fbfe050: A0 E1 BF 5F FF 7F 00 00 | .....
0x7fff5fbfe040: 00 0C 00 00 00 00 00 00 | .....
0x7fff5fbfe030: 00 E1 BF 5F FF 7F 00 00 | .....
0x7fff5fbfe020: 00 00 00 07 00 00 00 00 | .....
0x7fff5fbfe010: FF FF FF FF 00 00 00 00 | .....
0x7fff5fbfe000: 06 00 00 07 06 00 00 07 | .....
0x7fff5fbfe0f0: 10 A0 4A 00 01 00 00 00 | ..J.... 0x10044A010 => 0x7fff7b4fefa0 => 0x7fff7b4fepc8 => 0x7fff702af118 => 0x7fff702af118 => 0x7fff702af118 => 0x7fff702af118 => 0x7fff702af118
0x7fff5fbfe0e0: 00 00 00 00 00 00 00 00 | .....

[stack]
[0x0070:00007fff5fbfe040]
```


NEW TOY: RADARE2

► <https://github.com/radare/radare2>



- ▶ Wikipedia
- ▶ System V Application Binary Interface (AMD64)
- ▶ Procedure Call Standard for the ARM 64-bit Architecture
- ▶ iOS ABI Function Call Guide
- ▶ The Swift ABI
- ▶ The Swift Calling Convention
- ▶ Friday Q&A

相关资源

- ▶ <https://github.com/apple/swift>
- ▶ <https://github.com/hankbao/DictSwift>

THANKS

Q & A