Blackbox analysis of iOS apps

Dmitry 'D1g1' Evdokimov,
Security researcher at Digital Security (ERPScan)
#whoami

- Director of **DSecRG (ERPScan Research Group)**
- Section editor in the **Xakep** magazine
- Co-organizer of **DEFCON Russia & ZeroNights**
- Author of **Python arsenal for RE**

Specialized in finding vulnerabilities in binary applications without source code
Attention please!

It is not rocket science =)
This work is a compilation of public information and my own experience
Goals of this workshop

• How iOS and iOS applications work
• The basics of iOS vulnerabilities
• The skill of using common tools to find vulnerabilities in iOS apps
Blackbox analysis of iOS apps

Agenda

1. iOS platform
   1. How it works, Objective-C, ARM, security mechanisms, jailbreak
2. Introduction to Objective-C
3. iOS apps
   1. Mach-O format, application structure, …
4. iOS vulns
5. Blackbox testing
   1. Static and dynamic analysis
iOS

- iOS is derived from OS X, with which it shares Darwin
  - ARM
  - The kernel sources remain closed
    - __arm__, ARM_ARCH
  - Touch-based
    - SpringBoard
  - Security mechanisms
    - Sandbox as a jail
    - ...
iOS security mechanisms

• Code Signing
  - X.509v3 certificates

• Sandboxing (SeatBelt)
  - Inability to break the app’s directory
    - /var/mobile/Applications/<app-GUID>/
  - Inability to access any other process
  - Inability to use any hardware devices directly
  - Inability to generate code dynamically

• Privilege separation
  - Mobile user + Entitlements
Jailbreak

- Jailbreak depends on SW & HW
  - Tethered
  - Untethered

- Ability to access file system
  - Copy/edit any file in the system
- Bypassing sandbox restrictions
  - Break out of the app’s directory
- Launching unsigned applications
  - Launch applications that do not belong to App Store
Unauthorized modification of iOS can cause security vulnerabilities, instability, shortened battery life, and other issues.

Summary
This article is about adverse issues experienced by customers who have made unauthorized modifications to iOS (this hacking process is often called "jailbreaking").

Products Affected
iPad, iPhone, iPod touch

http://support.apple.com/kb/HT3743
ARM

- Advanced RISC Machine
- Load-store architecture
- Fixed-length instructions
- 3-address instruction formats
- Instructions:
  - Data transfer
  - Data processing
  - Control flow
ARM modes

1. ARM
   • Length(Instr) = 4 bytes
2. Thumb
   • Length(Instr) = 2 bytes
3. Thumb2
   • Length(Instr) = 2/4 bytes
4. Jazzle
   • Java bytecode + ARM/Thumb
**ARM32**

- **Registers:**
  - General Purpose: r0-r12
  - Stack Pointer: r13 (SP)
  - Link Register: r14 (LR)
  - Program Counter: r15 (PC)
  - Current Program Status Register
- **Calling Convention:**
  - Argument Values: r0-r3
  - Local Values: r4-r12
  - Return Value: r0
ARM 64-bit Architecture

1. iPhone 5S
2. AArch64 (ARM), ARM64 (Apple)

<table>
<thead>
<tr>
<th>Register</th>
<th>Special</th>
<th>Role in the procedure call standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP</td>
<td></td>
<td>The Stack Pointer.</td>
</tr>
<tr>
<td>r30</td>
<td>LR</td>
<td>The Link Register.</td>
</tr>
<tr>
<td>r29</td>
<td>FP</td>
<td>The Frame Pointer</td>
</tr>
<tr>
<td>r19...r28</td>
<td></td>
<td>Callee-saved registers</td>
</tr>
<tr>
<td>r18</td>
<td></td>
<td>The Platform Register, if needed; otherwise a temporary register. See notes.</td>
</tr>
<tr>
<td>r17</td>
<td>IP1</td>
<td>The second intra-procedure-call temporary register (can be used by call veneers and PLT code); at other times may be used as a temporary register.</td>
</tr>
<tr>
<td>r16</td>
<td>IP0</td>
<td>The first intra-procedure-call scratch register (can be used by call veneers and PLT code); at other times may be used as a temporary register.</td>
</tr>
<tr>
<td>r9...r15</td>
<td></td>
<td>Temporary registers</td>
</tr>
<tr>
<td>r8</td>
<td></td>
<td>Indirect result location register</td>
</tr>
<tr>
<td>r0...r7</td>
<td></td>
<td>Parameter/result registers</td>
</tr>
</tbody>
</table>
ARM64 Function Calling Conventions

In general, iOS adheres to the generic ABI specified by ARM for the ARM64 architecture. However, there are some choices to be made within that framework, and some divergences from it. This document describes these issues.

Choices Made Within the Generic Procedure Call Standard

Procedure Call Standard for the ARM 64-bit Architecture delegates certain decisions to platform designers. Decisions made for iOS are described below.

- The register x18 is reserved for the platform. Conforming software should not make use of it.
- _wchar_t is 32-bit and Long is a 64-bit type.
- Where applicable, the __fp16 type is IEEE754-2008 format.
- The frame pointer register (x29) must always address a valid frame record, although some functions—such as leaf functions or tail calls—may elect not to create an entry in this list. As a result, stack traces will always be meaningful, even without debug information.
- Empty struct types are ignored for parameter-passing purposes. This behavior applies to the GNU extension in C and, where permitted by the language, in C++.

Divergences from the Generic Procedure Call Standard

iOS diverges from Procedure Call Standard for the ARM 64-bit Architecture in several ways, as described here.
Development for iOS

- Mac
- Xcode
  - gcc/LLVM/LLVM-gcc compilers
  - iPhone Simulator (i386)
- Cocoa Touch
- Objective-C
  - Other: HTML, JavaScript, C# & .NET (Xamarin)
Objective-C

• Object-oriented language
• Based on:
  • Strict superset C
  • Smalltalk

@interface Appirater : NSObject <UIAlertViewDelegate> {
    UIAlertView* ratingAlert;
}
@property(retain, nonatomic) UIAlertView* ratingAlert;
+(void)rateApp;
+(void)userDidSignificantEvent:(BOOL)user;
+(void)appEnteredForeground:(BOOL)foreground;
+(void)appWillResignActive;
+(void)appLaunched:(BOOL)launched;
+(void)appLaunched;
+(id)sharedInstance;
-(void)alertView:(id)view clickedButtonAtIndex:(int)index;
-(void)hideRatingAlert;
-(void)incrementSignificantEventAndRate:(id)rate;
-(void)incrementAndRate:(id)rate;
-(void)incrementSignificantEventCount;
-(void)incrementUseCount;
-(BOOL)ratingConditionsHaveBeenMet;
-(void)showRatingAlert;
-(BOOL)connectedToNetwork;
@end
### Calling methods

<table>
<thead>
<tr>
<th>C++</th>
<th>ObjectPointer-&gt;MethodName(param1, param2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obj-C</td>
<td>[ObjectPointer MethodName:param1 param2Name:param2]</td>
</tr>
</tbody>
</table>

```
objc_msgSend(ObjectPointer, "MethodName", "param1", "param2")
```

```
objc_msgSend()
objc_msgSendSuper()
objc_msgSend_fpret()
objc_msgSend_stret()
objc_msgSendSuper_stret()
objc_msgSendSuper2()
```
Go to device

- Jailbreak
- Cydia
- SSH/putty
- itunnel_mux
- WinSCP/scp
Prepare env in device

- otool
- class-dump-z
- APT 0.6 Transitional
  - apt-get
- Command line tools
  - curl, dpkg, file, grep, netcat, python, sed, …

```bash
dsec:~ root# apt-get install adv-cmds curl cycript odcctools developer-cmds dpkg com.ericasadun.utilities file file-cmds findutils gawk git grep inetutils com.autopar.installicp ldid less lsof mobilesubstrate com.saurik.substrate.safemode mobileterminal-applesdk nano netcat network-cmds python sed shell-cmds sqlite3 syslogd system-cmds tcpdump top uikittools unrar unzip vim wget whois zip
```
Install apps from console

- **Debian package**
  
  ```bash
dpkg -i <package.deb>
kxall -HUP SpringBoard
  ```

- **App without developer license or patched**
  
  ```bash
cp -r HelloWorld.app/ root@yourIP:/Applications/uicache
  ```

- **IPA:**
  - IPA Installer Console
  - iPhone Configuration Utility
Useful commands

- `cd /private/var/mobile/Applications`
- `find . -name '*Appname*'`
- `cd /private/var/mobile/Applications`
- `ls -l | grep 'Time'`
Applications

- **AppStore**
  - IPA packages = ZIP files
- **On devices**
  - `/private/var/mobile/Applications/<UUID>/<AppName>.app/`

```
dsec:/private/var/mobile/Applications root# ls -l
total 0
drwxr-xr-x  6 mobile mobile 272 Oct  5 18:08 024F3845-FC4F-4526-80FD-15D7FD502E63/
drwxr-xr-x  6 mobile mobile 272 Mar 15 2013 03AE93B2-C8BE-450F-ACC6-E4FE8488E7D/
drwxr-xr-x  6 mobile mobile 272 Oct  5 18:16 061C30B1-9D39-4806-82F8-6CFE65D55600/
```

- **Apple apps**
  - `/Applications/`
Mach-O file format basic structure
Mach-O header

1. 32bit (ARMv6, ARMv7)
   - 0xFEEDFACE
2. 64bit
   - 0xFEEDFACF
3. Universal binaries (FAT)
   - 0xCAFEBABE
Blackbox analysis of iOS apps

## Application structure

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppName.app/</td>
<td>App</td>
</tr>
<tr>
<td>Documents/</td>
<td>Data files saved by the app</td>
</tr>
<tr>
<td>Library/</td>
<td>Miscellaneous app files</td>
</tr>
<tr>
<td>iTunesArtwork</td>
<td>App icon</td>
</tr>
<tr>
<td>iTunesMetadata.plist</td>
<td>The property list of the app</td>
</tr>
<tr>
<td>tmp/</td>
<td>Directory for temporary files</td>
</tr>
</tbody>
</table>

```
dsec:/private/var/mobile/Applications/3B3EA1C2-BCF7-47FC-940C-6F71106DDCFB root# ls -l
```

```
total 72
drwxr-xr-x  7 mobile mobile  510 Jul 23 15:35 AngryBirdsRio.app/
drwxr-xr-x  2 mobile mobile  272 Nov  1 14:51 Documents/
drwxr-xr-x  6 mobile mobile  204 Nov  1 14:47 Library/
-rw-r--r--  1 mobile mobile 58614 Nov  1 14:36 iTunesArtwork
-rw-r--r--  1 mobile mobile 2010 Nov  1 14:36 iTunesMetadata.plist
drwxr-xr-x  2 mobile mobile   68 Nov  1 14:48 tmp/
```
Decrypt app from AppStore

1. gdb
   - Choosing the right architecture (if FAT)
   - Breakpoint at `start`
2. Clutch
3. `dumpdecrypted.dylib`

```
root# otool -l AngryBirdsRio | grep LC_ENCRYPTION_INFO -B1 -A4
Load command 11
    cmd LC_ENCRYPTION_INFO
    cmdsize 20
    cryptoff 8192
    cryptsize 5804032
    cryptid 1
```
Decrypted

- **Clutch**
  - `/var/root/Documents/Cracked/`

```
dsec:~ root# ./Clutch-1.3 AngryBirdsRio
Clutch-1.3
Cracking AngryBirdsRio...
```

- `dumpdecrypted.dylib`
Blackbox analysis of iOS apps

OWASP Mobile Top 10 Risks

M1 - Insecure Data Storage
M2 - Weak Server Side Controls
M3 - Insufficient Transport Layer Protection
M4 - Client Side Injection
M5 - Poor Authorization and Authentication
M6 - Improper Session Handling
M7 - Security Decisions Via Untrusted Inputs
M8 - Side Channel Data Leakage
M9 - Broken Cryptography
M10 - Sensitive Information Disclosure
Traffic analysis

• Passive network traffic monitoring with tcpdump

```
dsec:~ root# tcpdump -i en0 -w /var/root/capture.pcap
tcpdump: listening on en0, link-type EN10MB (Ethernet), capture size 68 bytes
^C1782 packets captured
1785 packets received by filter
0 packets dropped by kernel
dsec:~ root# ```

Then load the *.pcap file into wireshark for analysis

• Gateway method
• BurpSuite
  • HTTPS: Import PortSwigger CA to the iDevice
• dnsRedir
• Mallory (by Intrepidus Group)
Certificate pinning?!

- Pinning is the process of associating a host with their *expected* X509 certificate or public key.
- OWASP
  - [https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning#iOS](https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning#iOS)
- Attack
  - `trustme`
    - `SecTrustEvaluate`
  - `ios-ssl-killswitch`
    - `SSLCreateContext, SSLSetSessionOption, SSLHandshake`
Working with SSL certificates

- **NSURLConnection class**
- **Accepting a self-signed certificate or incorrect error processing**
  - `allowsAnyHTTPSSCertificateForHost`
  - `setAllowsAnyHTTPSSCertificate`
  - `continueWithoutCredentialForAuthenticationChallenge`
CFStreams sockets

- `kCFStreamPropertySSLSettings`
  - `kCFStreamSSLLevel`
  - `kCFStreamSSLAllowsExpiredCertificates`
  - `kCFStreamSSLAllowsExpiredRoots`
  - `kCFStreamSSLAllowsAnyRoot`
  - `kCFStreamSSLValidatesCertificateChain`
  - `kCFStreamSSLPeerName`
Cross-site scripting

- **UIWebView** class
  - `stringByEvaluatingJavaScriptFromString`
  - `shouldStartLoadWithRequest`
List of interesting strings

- Don’t use and don’t leak
  - UDID
  - IMEI
  - ICCID
  - PII
  - OSN-ID
  - LID
XML injections

- XML External Entity (XXE) flaws
- NSXMLParser class
  - `setShouldResolveExternalEntities`
  - `foundExternalEntityDeclarationWithName`
- `libxml2` library
  - `_xmlParseMemory`
- 3rd party libraries and classes
Directory traversal

- **NSFileManager class**
  ```objective-c
  NSString *file = [[NSString alloc] initWithFormat: @"%@/%@
NSTemporaryDirectory(), attackerControlledString];

  NSFileManager *m = [NSFileManager defaultManager];

  [m createFileAtPath:text contents:nsd attributes:nil];
  ```
  - contentsAtPath
  - fileHandleForReadingAtPath

- **C functions**
  - fopen
  - ...

Blackbox analysis of iOS apps
File storage

- **NSFileManager class**
- **NSFileProtectionKey attribute**
  - NSFileProtectionNone
  - NSFileProtectionComplete
  - NSFileProtectionCompleteUnlessOpen
  - NSFileProtectionCompleteUntilFirstUserAuthentication

- **Tools:**
  - filemon.iOS
  - FileDP
Blackbox analysis of iOS apps

filemon.iOS

GOT PID: 10280 and rc: 0 - iGoat
iGoat (PID:10280) Created /private/var/mobile/Library/Preferences/com.krvw.iGoat.plist.NPoVZLR
DEV: 1,3 INODE: 591109 MODE: 8180 UID: 501 GID: 501 Arg64: 1398076394374

GOT PID: 10280 and rc: 0 - iGoat
iGoat (PID:10280) Modified /private/var/mobile/Library/Preferences/com.krvw.iGoat.plist.NPoVZLR
DEV: 1,3 INODE: 591109 MODE: 8180 UID: 501 GID: 501 Arg64: 1398076480517

GOT PID: 10280 and rc: 0 - iGoat
iGoat (PID:10280) Chowned /private/var/mobile/Library/Preferences/com.krvw.iGoat.plist.NPoVZLR
DEV: 1,3 INODE: 591109 MODE: 8180 UID: 501 GID: 501 Arg64: 1398076485006

GOT PID: 10280 and rc: 0 - iGoat
iGoat (PID:10280) Renamed /private/var/mobile/Library/Preferences/com.krvw.iGoat.plist.NPoVZLR
DEV: 1,3 INODE: 591109 MODE: 8180 UID: 501 GID: 501 /private/var/mobile/Library/Preferences/com.krvw.iGoat.plist
8180 UID: 501 GID: 501 Arg64: 1398076500136
Plist files

- **plist** – property lists
- Serialized objects
- XML
- **NSUserDefaults** class

**Tools:**
- Python library: `plistlib, bpplist`
- plist Editor
- `plutil`
  - `plutil - convert xml1`
SQLite and SQL injections

- **SQLite database**
  - `/usr/lib/libsqlite3.dylib`
  - `/<GUID>/Documents/`
    - `*.sqlite`, `*.db`, `*.sqlite3`
    - `sqlite3_open`
    - `sqlite3_prepare_v2`
    - `sqlite3_step`
    - Use parameterized queries
      - `sqlite3_bind_*`
Keychain

- **Secure storage**
  - File `/private/var/Keychains/keychain-2.db`
  - `SecItemAdd()`
  - `SecItemUpdate()`
  - `SecItemCopyMatching()`
  - `SecItemDelete()`

- **Tools:**
  - `keychain_dumper`
  - `keychain_dump`
Cookies

- **Persistent cookies**: `Cookies.binarycookies`
  - `/private/var/mobile/Library/`
  - `/private/var/mobile/<App GUID>/Library/Cookies`

- **Tool**: `BinaryCookieReader.py`
NSLog() 

Tools: 
• iPhone Configuration Utility 
• syslogd
Cache

- **UIPasteboard class**
  - `generalPasteboard`

- **Backgrounding**
  - `<Application
    GUID>/Library/Caches/Snapshots/*/*.png`
  - `applicationDidEnterBackground`

- **Keyboard cache**
  - `/var/mobile/Library/Keyboard/en_GB-dynamic-text.dat`
  - `secureTextEntry = Yes`
  - `autocorrectionType = UITextAutocorrectionTypeNo`
• URL schemes
  • `handleOpenURL`
  • `openURL`
Memory corruptions

- **Obj-C + C/C++ function =**
  - **Format string**
    - NSLog()
    - [NSString stringWithFormat:]
    - [NSString initWithFormat:]
    - [NSMutableString appendFormat:]
    - [NSAlert informativeTextWithFormat:]
    - [NSPredicate predicateWithFormat:]
    - [NSEXception format:]
    - NSRunAlertPanel

- **Buffer overflow**
- **Use-after-free**
Check for exploit mitigations

- **Stack cookie**
  - \_stack\_chk\_fail
  - \_stack\_chk\_guard

- **PIE**

- **ARC**
  - \_objc\_release
  - \_objc\_retainAutoreleaseReturnValue
  - \_objc\_autoreleaseReturnValue
  - \_objc\_storeStrong
  - \_objc\_retain
  - \_objc\_retainAutoreleasedReturnValue

Blackbox analysis of iOS apps
Blackbox analysis of iOS apps
IDAPro

Blackbox analysis of iOS apps

IDA Pro

© 2002—2013, Digital Security
radare2 ARM64 Mach-O
Blackbox analysis of iOS apps

Hopper

```c
// Some code here
```

© 2002—2013, Digital Security
Blackbox analysis of iOS apps

iNalyzer

**iNalyzer Packager**

How to use:

1. Install **GraphViz.Dot** on PC/Laptop
2. Install **DoxyGen** on PC/Laptop
3. Choose Application from the list and click Package

Choose application to Pack: **AngryBirdsRio**

Package

Be patient as package creation can take a while

4. Save .zip to disk and extract
5. Run Doxygen wizard on the doxygen.template(Win) or doxMe.sh(Other)
6. Then open main dashboard page at Doxygen/html/index.html with FireFox
7. To toggle Cycrypt console on and off double press the left-arrow keyboard key, make sure to set the iDevice IP to your target and to swich to your application when trying to communicate with it
cycrypt

dsec:~ root# ps -ax | grep iGo
  9194 ?? 0:01.70 /Applications/iGoat.app/iGoat
  9196 ttys001 0:00.00 grep iGo

dsec:~ root# cycrypt -p 9194

cy$ function printMethods(className) {
  cy> var count = new new Type("I");
  cy> var methods = class_copyMethodList(objc_getClass(className), count);
  cy> var methodsArray = [];
  cy> for(var i = 0; i < *count; i++) {
  cy>   var method = methods[i];
  cy>   methodsArray.push({selector:method_getName(method), implementation:method_getImplementation(method)});
  cy> }
  cy$ free(methods);
  cy$ free(count);
  cy$ return methodsArray;
  cy$ }

cy$ printMethods(SQLInjectionExerciseController)
[[{selector:@selector(setSearchField:), implementation:0x46605},{selector:@selector(searchField),implementation:0x465dd},{selector:@selector(search:),implementation:0x45fc1},{selector:@selector(.cxx_destruct),implementation:0x46635}]

cy$ []
Blackbox analysis of iOS apps

Introspy

User Applications Profiling

- Bitzer
- Evernote
- Good
- LogMeIn
- McAfee EMM
- Password Keeper & Data Safe
- Receiver
- SafeKey
- Twitter
- Xfinity Connect

Views:
- Videos
- Photos & Camera
- Twitter
- Facebook
- Activator
- Introspy - Apps
- Introspy - Settings
- SSL Kill Switch
- Developer
- Bitzer

© 2002—2013, Digital Security
Snoop-it

Blackbox analysis of iOS apps
Q&A

d.evokimov@dsec.ru
@evdokimovds