Introspy

Security Profiling for Blackbox iOS and Android

Marc Blanchou

Alban Diquet

Introduction – What is it about?

- Tool release: Introspy
  - Security profiler for iOS and Android applications
  - Useful to developers, pen-testers & security researchers
- Security profiling ?
  - Figuring out what an application is doing at runtime
  - Automatically Identifying potentially dangerous behaviors

Introduction – Who are we?

- Three persons worked on this project
  - Tom Daniels – *github/thirstscolr*
  - Marc Blanchou – *github/mblanchou*
  - Alban Diquet – *github/nabla_cod3*

- Security Consultants @ iSEC Partners
  - Security consulting company
  - Based in San Francisco

Agenda

- Mobile threats
- Blackbox iOS & Android
- Introspy
- Demo
- Conclusion

Mobile Security

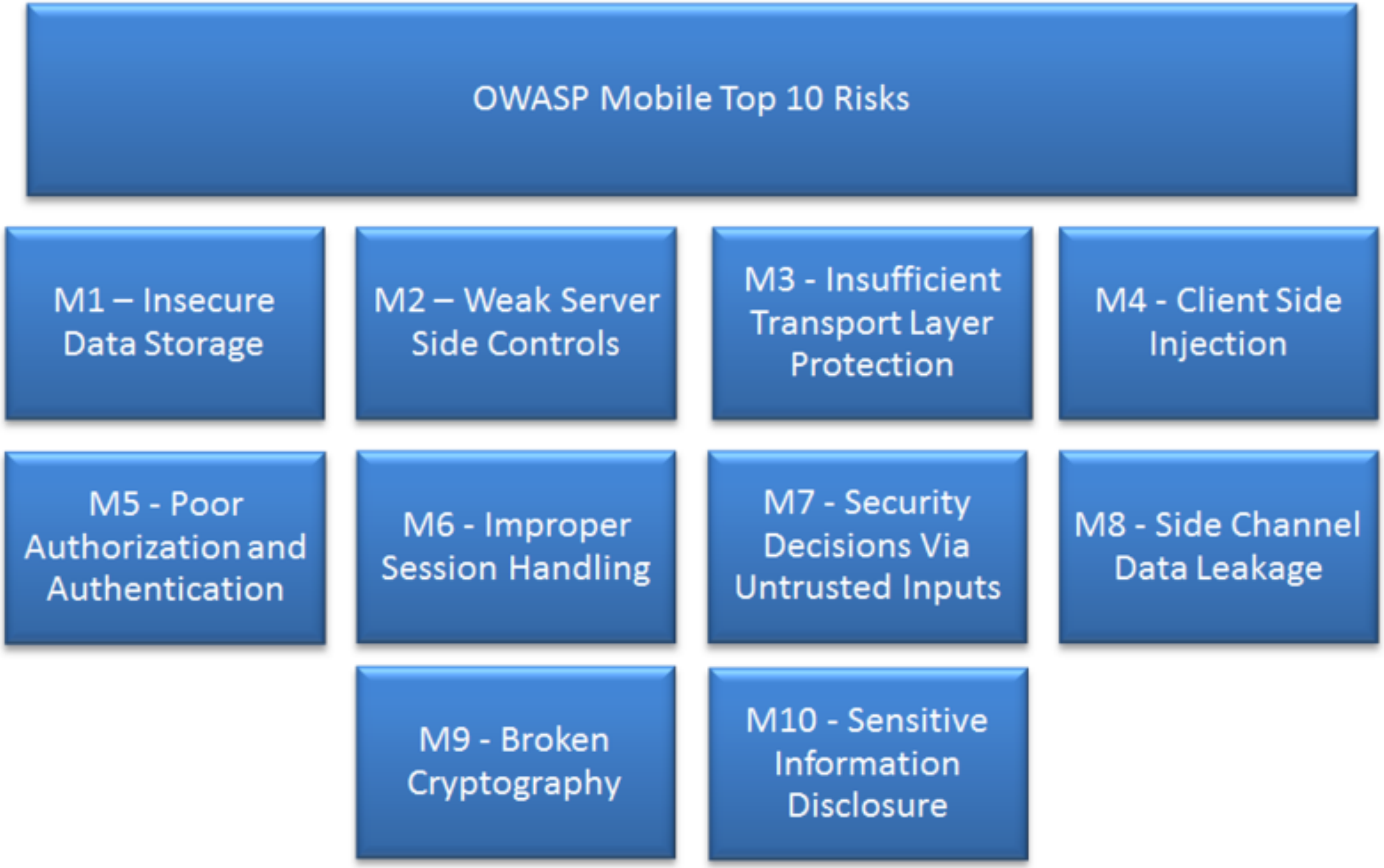
- Sensitive data is stored on the device
  - User data: mobile banking, password managers
  - Corporate data: email, documents, VPN credentials

- Mobile security model
  - Always-on
  - Highly exposed
  - Constant access required
    - Low requirements for passcode
    - Small keyboard, weak CPU

Mobile Attack Vectors

- Malicious application running on the device
  - Poorly policed markets
  - Exploits
  - Side-loading
- Active network attacker
  - Wifi or even GSM
- Stolen device

OWASP Mobile Top 10



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

Blackbox Testing

- No access to the source code
- Usually time-constrained
- Tester has to:
  - Understand how the app works
  - Understand how it interacts with other components/apps
  - Identify security issues

Blackbox Testing: Methodology

- Static testing
  - Recovering the application’s binary and analyzing it
- Dynamic testing
  - Proxy-ing the application’s network traffic
  - I/O monitoring (file system etc.)
  - Debugging the application (Gdb, JDB, Cycript)
  - Hooking functions (Cydia Subtrate, Xposed)

Blackbox Testing: iOS

Dynamic analysis

- Run the application and look at inputs / outputs
  - Files in the application's container, keychain, preferences
  - IPCs: pasteboard, URI schemes
  - Network
- Hook methods using MobileSubstrate
- Monitor/debug the application using GDB or Cycript
- Bypass jailbreak detection (xCon)

Blackbox Testing: iOS

Static analysis

- Decrypt and analyze the binary
  - Dump encrypted code section (Appstore DRM)
  - Use mach-o tools
    - Otool
    - Class dump
- IDA + obj_helper

Blackbox Testing: Android

Dynamic analysis

- Run the application and look at inputs / outputs
  - File system and preferences
  - Exported IPCs: Activities, Receivers, Content Providers, Services
  - Network
- Monitor/debug the application using JDB or GDB for native
- Automatize hooking using Cydia Substrate

- Memory analysis (what is available and when)
  - Leverage the GC
  - Pro filesystem (/proc/[PID])

Blackbox Testing: Android

Static analysis

- Convert Dalvik bytecode to Java bytecode
- Decompile to Smali or Java
- Can usually be re-compiled and re-signed with modifications / debug info (from Smali code)
- Use IDA on native parts
- Control flow visualizers

Blackbox Testing: Conclusion

- Lack of automated, security-focused tools on Mobile
  - Debuggers and hooking frameworks are generic
  - Better tools are available on the desktop
- It should be easier than this
  - Most security issues on Mobile are well-known
  - Pen-testing engagements are time-constrained

Introspy

- Security profiler for iOS and Android applications
- Goals
  - Easy to use
  - Help the tester understand what an application is doing at runtime
  - Automatically identify potentially dangerous behaviors

Introspy: How it Works

Introspy is actually comprised of three components:

- Two “tracers”
  - One for iOS, one for Android
  - Runs on the device
  - Collects data about functions called by the applications
- One “analyzer”
  - Runs on the tester’s computer
  - Analyzes data collected by the tracers

Introspy: Android & iOS Tracers

- Has to be installed on a jailbroken/rooted device

- Hooks security-sensitive system APIs
  - Logs API calls made by applications
    - Class, method name, arguments and return value
  - Hooks implemented using Cydia/Mobile Substrate

- Stores logged data in a SQLite DB on the device
  - Optionally displays function calls to the console in real-time

Introspy: iOS Tracer

MobileSubstrate

- "de facto framework that allows 3rd party developers to provide runtime patches to system functions"
- Easy to use and very powerful
- Hooks C functions as well as Objective-C methods
- Requires a jailbroken device
- http://iphonedevwiki.net/index.php/MobileSubstrate

Introspy: iOS Tracer

```
/* Example: hooking rand() */
extern SQLiteStorage *traceStorage; // Introspy's SQLite storage functions
static int (*original_rand)(); // Points to the "original" rand()

// Introspy code to replace rand()
static int replaced_rand() {

    int origResult = original_rand(); // Call the original rand() and store the result
    // Log this function call to the Introspy DB
    CallTracer *tracer = [[CallTracer alloc] initWithClass:@"C" andMethod:@"rand"];
    [tracer addReturnValueFromPlistObject: [NSNumber numberWithUnsignedInt:origResult]];
    [traceStorage saveTracedCall: tracer];
    [tracer release];

    return origResult;
}

MSHookFunction(rand, replaced_rand, (void **) &original_rand); // Hook rand()
```

Introspy: iOS Tracer

Security-Sensitive APIs on iOS ?

- **Crypto:** CCCryptor, CCHmac, CCDigest, rand(), etc.
- **IPCs:** UIPasteboard, URI Handlers
- **File System:** NSData, NSFileHandle, NSFileManager, NSInputStream, etc.
- **User Preferences:** NSUserDefaults
- **Keychain**: SecItemAdd(), SecItemDelete(), etc.
- And more…

Introspy: Android Tracer

Cydia Subtrate

- Supported from Android 2.3 to 4.3
- Same person behind Mobile Substrate on iOS
- Inject code into the Zygote process
- Hook “all” traditional and system apps
- Can also hook native code with a native API (as opposed to Xposed)

Introspy: Android Tracer

Security-Sensitive APIs on Android ?

- Crypto
  - javax.crypto.Cipher (init, update, dofinal etc.)
  - java.crypto.spec (KeySpec, PBEKeySpec)
  - Etc.
- IPCs
  - startService, startActivity, registerReceiver, sendBroadcast, grantUriPermission etc.
  - Programmatic permissions
- File permissions
  - java.io.File, java.io.FileOutputStream etc.
- Shared (hidden?) preferences, URI handlers, Logs, etc.
- SSL
  - Used everywhere? Cert validation?

Introspy: Analyzer

- Python script running on the tester's computer
- Enumerates and retrieves tracer DBs available on the device (using SSH)
- Analyzes and processes tracer DBs
    - Turns a tracer DB into an HTML report
    - Can also list all files or URLs accessed by the application

Demo

Introspy: Limitations

- It doesn't trace what happens outside of the system APIs
  - Including libraries packaged with the app (such as OpenSSL)
  - We may add hooks to support popular libraries
- It requires a good understanding of the iOS & Android frameworks/APIs
  - Not an autopwn tool

Try it !

- Only the iOS version is available for now
  - https://github.com/iSECPartners/introspy
  - Feedback/suggestions appreciated

- Android version to be released soon™

- Lots of other pen-testing tools on iSEC Partners' Github
  - Mobile, Web, Network, etc.

There's More… Bonus Tools

- Cydia Substrate extension for Android to bypass SSL certificate pinning checks
  - Hook 6 different methods that applications can use to implement certificate pinning for SSL connections
  - Modify return values or what data is passed to these methods to accept invalid SSL certificates
  - https://github.com/iSECPartners/Android-SSL-TrustKiller

- SSL cert pinning bypass on iOS: https://github.com/iSECPartners/ios-ssl-kill-switch

- Cydia Substrate extension for Android to make any application debuggable: https://github.com/iSECPartners/Android-OpenDebug

Acknowledgements

- Saurik
- Dhowett
- Rpetrich
- Irc.saurik.com
  - #theos

Thank You

- Marc Blanchou
  - Principal Security Consultant at iSEC Partners
  - marc@isecpartners.com

- Alban Diquet
  - Principal Security Consultant at iSEC Partners
  - http://nabla-cod3.github.io
  - alban@isecpartners.com

Questions ?

UK Offices

Manchester - Head Office
Cheltenham
Edinburgh
Leatherhead
London
Thame

European Offices

Amsterdam - Netherlands
Munich – Germany
Zurich - Switzerland

North American Offices

San Francisco
Atlanta
New York
Seattle

Australian Offices

Sydney