Reverse Engineering 101

What is "reverse engineering"?

- "Is this legal?"
- Dark arts known only to a select few

Reverse engineering actually is...

- Alternative problem solving skill
- A learning process

What can these skills be applied to?

- Anti-piracy measures in software
- Reproducing behavior
- Enhancing existing products and services

How do I start?

- Start with a project of your own, get familiar with the tools
- Build off the work of others or join an existing project
- Don't be afraid to get in over your head

The Approach

- Determine the target
- Document the starting state
- Work in small steps between all tools, and verify before moving forward

Getting Started on OS X and iOS

Step 1: Determine the target

- Frameworks, libraries, loadable code
- Applications, running/active processes

Step 2: Where to start

- Type of target defines the starting approach
- Create a foundation based on documentation
- What is your goal, what do you hope to learn

Step 3: Verification

- Documentation is critical during initial stages
- Switch between tools - get a clear picture
- Differences in low level and high level analysis

Static Analysis

Disassemblers

- Generate assembly code from binaries
- Requires understanding of code logic, not programmer logic
- Tools:
  - otool/otx
  - Hopper/IDA

Decompilers

- Generate pseudo or compile-able code from assembly
- Useful overviews into how code works
- Tools:
  - Hopper/Hex-Rays
  - classdump/codedump

Hex Editors

- Not everything you need to know is code
- Resource and supplemental files
- Tools:
  - Hexfiend
  - 0xED
  - Synalyze it!

Dynamic and Runtime Analysis

Debuggers

- Step through code
- Manipulate and track application state
- Test understanding of static code analysis
- Tools:
  - GDB
  - LLDB

Memory Editors

- Finding data in memory
- Modifying application state
- Understanding how data is used
- Tools
  - Bit Slicer

samdmarshall.com/mactech.html

Daodan

- (present): code injection and binary analysis
- (future): all purpose toolkit for OS X and iOS reverse engineering and experimentation

Toolkit

- Disassembler, symbol remapping, code injection, and Objective-C decompiling all built-in
- Stand-alone controller app that can manage and monitor applications through this library
- Sandbox friendly and System ABI safe

Instructions

Open a Terminal window

$ bash

$ export DYLD_FORCE_FLAT_NAMESPACE=1;

$ export DYLD_INSERT_LIBRARIES=/path/to/libDaodan.dylib;

$ /Applications/MyApp.app/Contents/MacOS/MyApp

Reverse Engineering Resources

Sam Marshall

me@samdmarshall.com

@Dirk_Gently