RSAConference2015

Singapore | 22-24 July | Marina Bay Sands

CHANGE

Challenge today's security thinking

SESSION ID: MBS-R03

ShadowOS: Modifying the Android OS for Mobile Application Testing

Ray Kelly

Research and Innovation
HP Fortify On Demand
@vbisbest

About Me

- Ray Kelly
- Innovation and Research, HP Fortify on Demand
- SPI guy, lead developer of WebInspect
- FoD Mobile pen test team
- Twitter: @vbisbest

Agenda

- Why is mobile testing important
- Challenges of mobile testing
- Example mobile vulnerabilities
- How do we make this easier, ShadowOS
- The Android build process
- Identify key Android source code files for modification
- Demonstrate a custom Android OS with intercepting code

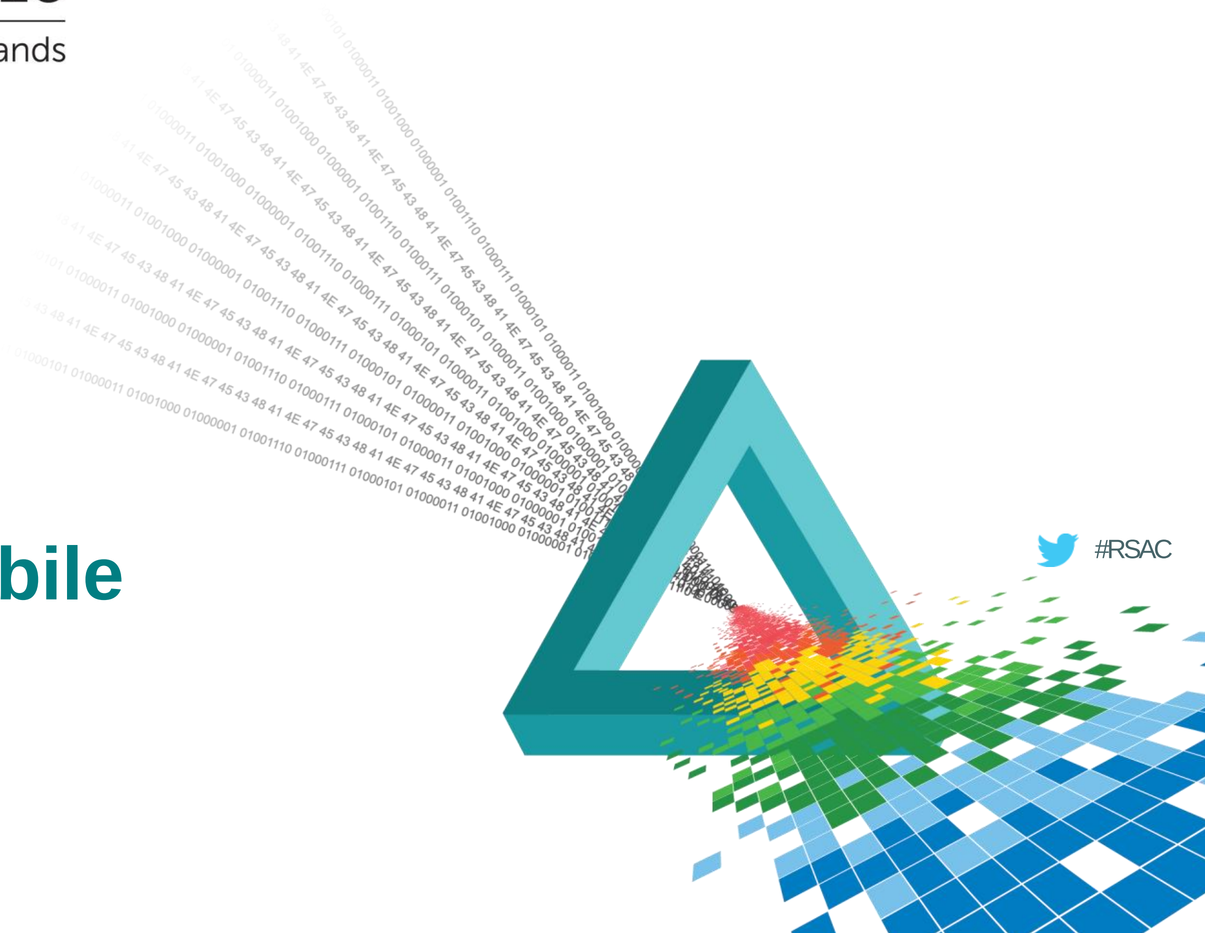
MITF?

Why is Mobile Testing Important

- Mobile development is the hottest type of development right now. New surface area equals dangerous surface area
- If anyone’s going to put features over security to get the product out the door, it’s likely to be a mobile team
- Many enterprise mobile developers haven’t had the security training that other types of developers have had – Anyone can make apps, its easy!
- Many assume that because mobile back ends aren’t visited directly they are more secure (obscurity assumption)

RSAConference2015

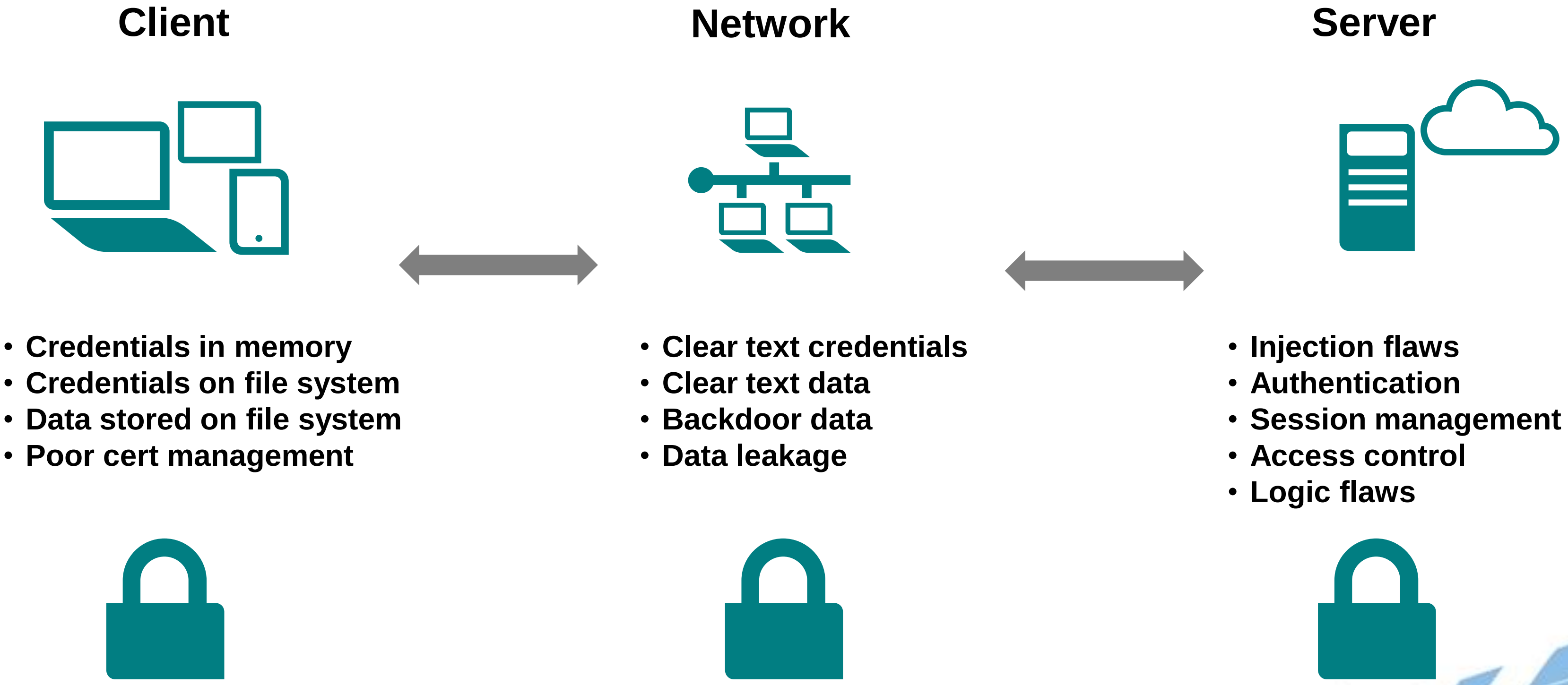
Singapore | 22-24 July | Marina Bay Sands



#RSAC

Challenges of Mobile Testing

Full Mobile App Coverage



Client

- Credentials in memory
- Credentials on file system
- Data stored on file system
- Poor cert management

Network

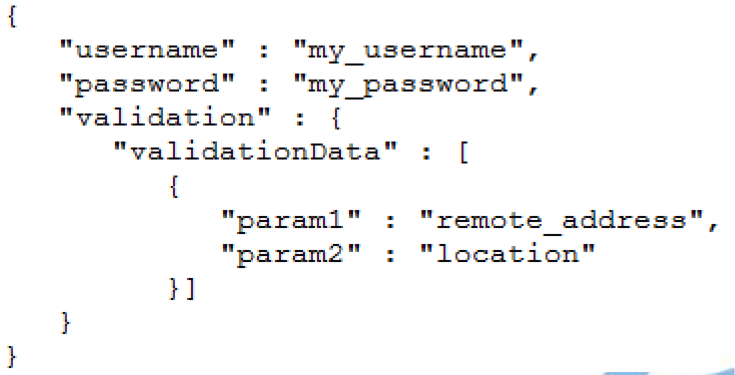
- Clear text credentials
- Clear text data
- Backdoor data
- Data leakage

Server

- Injection flaws
- Authentication
- Session management
- Access control
- Logic flaws

Server

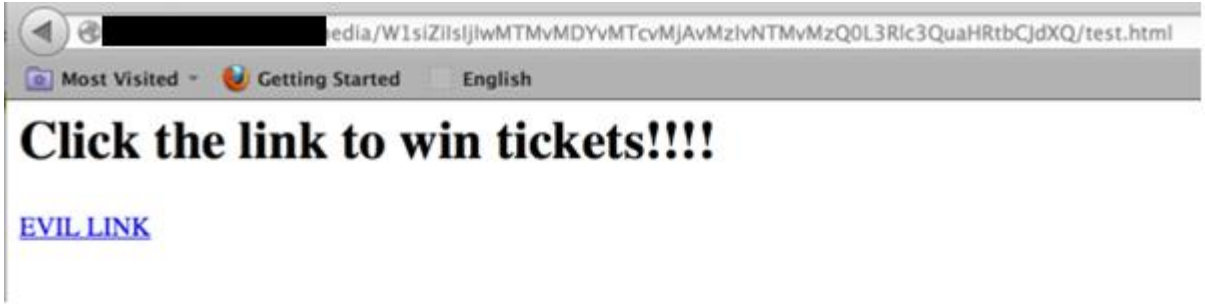
- Mobile API's are vulnerable to most of the same vulnerabilities as standard websites e.g. SQL Injection, XSS, path traversal etc.
- Testing JSON/XML based API's should be tested with valid structures as well as invalid structures.
- Difficult to test when app is using SSL and pinning certificates.



```
{
    "username" : "my_username",
    "password" : "my_password",
    "validation" : {
        "validationData" : [
            {
                "param1" : "remote_address",
                "param2" : "location"
            }]
    }
}
```

Server

- Backend API allowed WebDAV



[redacted]edia/W1siZiIsIjIwMTMvMDYvMTcvMjAvMzIvNTMvMzQ0L3Rlc3QuaHRtbCJdXQ/test.html

Click the link to win tickets!!!!

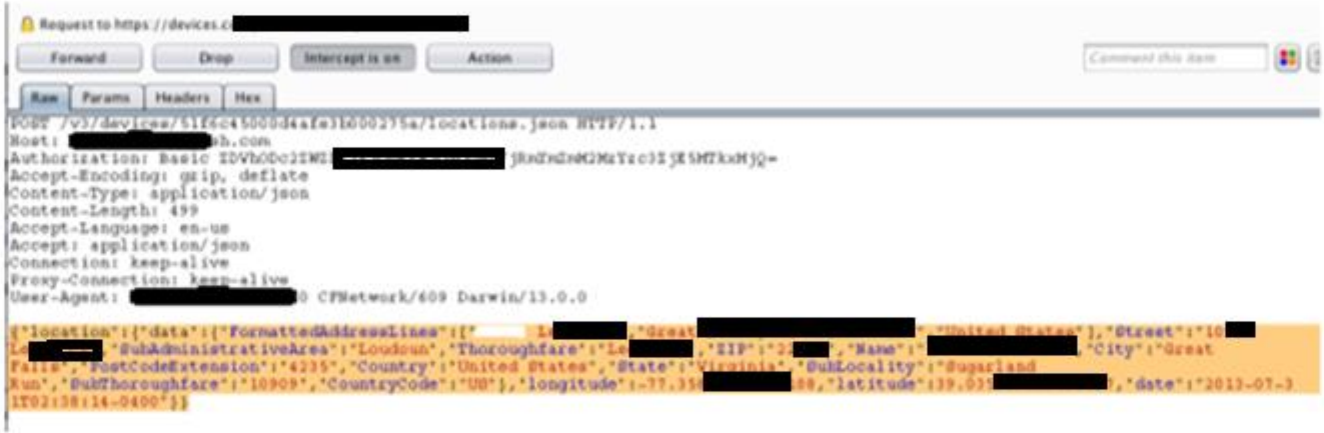
EVIL LINK

Network

- Privacy/data leakage, clear text data
- 3rd party data leakage
- Need to MITM, same challenges as server side
- Difficult to test when app is using SSL and certificate pinning

Network

- Transmission of private information
- Used SSL but did not pin certificate



```
POST /v3/devices/51f6c45000d4afe3100275a/locations.json HTTP/1.1
Host: [redacted]h.com
Authorization: Basic ZDVhODc2IWZ[redacted]jRmZmZmM2MzYzc3IjE5MTkxMjQ=
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 499
Accept-Language: en-us
Accept: application/json
Connection: keep-alive
Proxy-Connection: keep-alive
User-Agent: [redacted]0 CFNetwork/609 Darwin/13.0.0

{"location":{"data":{"FormattedAddressLines":["      Le[redacted],"Great[redacted]","United States"],"Street":"10[redacted]
Le[redacted],"SubAdministrativeArea":"Loudoun","Thoroughfare":"Le[redacted],"ZIP":"2[redacted]","Name":"[redacted],"City":"Great
Falls","PostCodeExtension":"4235","Country":"United States","State":"Virginia","SubLocality":"Sugarland
Run","SubThoroughfare":"10909","CountryCode":"US"},"longitude":-77.35[redacted]88,"latitude":39.03[redacted]?,"date":"2013-07-3
1T02:38:14-0400"}}
```

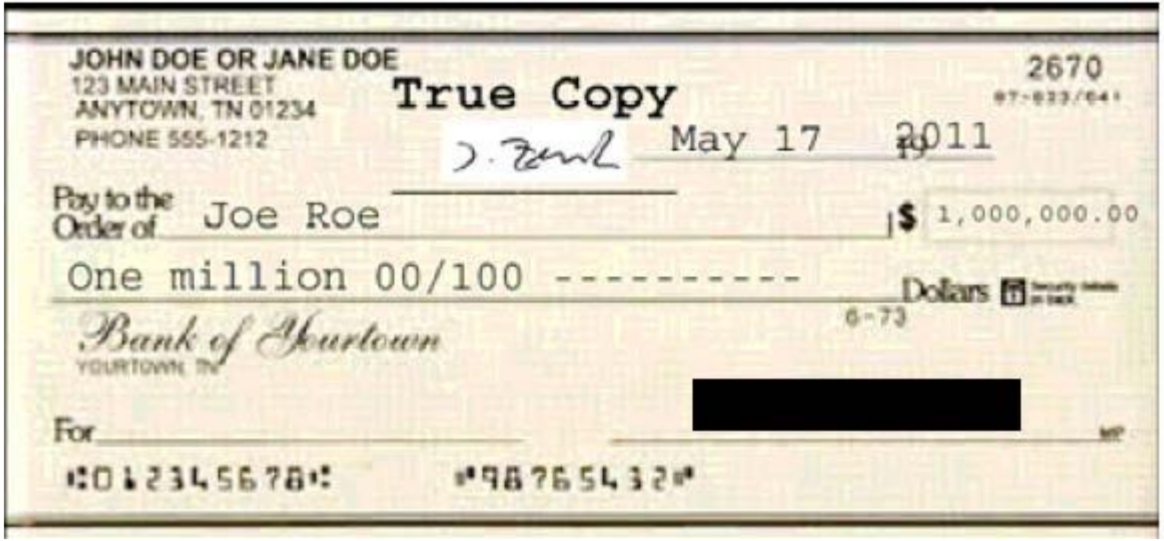
Client

- The big unknown especially without source code. Even with source code its not always easy (what is sensitive input?)
- What is being written to the file system?
  - Credentials
  - Private information
  - Sensitive photos outside of sandbox
- SQL Lite
  - Application storage
  - iOS WebKit cache (includes query string)

Client

- Promiscuous client-side storage
  - Storage of credentials in plist files, SQLite databases
  - Failure to use Key Chain to store credentials
  - Storage of sensitive application data on file system
  - Apps storing their images in the public folder rather than in their sandbox
  - Applications logging to the system log, but sending sensitive app data along with it (e.g. logcat output)

Photo Storage



JOHN DOE OR JANE DOE
123 MAIN STREET
ANYTOWN, TN 01234
PHONE 555-1212

True Copy

2670
87-833/641

May 17 19 2011

Pay to the Order of Joe Roe $ 1,000,000.00

One million 00/100 ----------- Dollars

6-73

Bank of Yourtown
YOURTOWN, TN

For

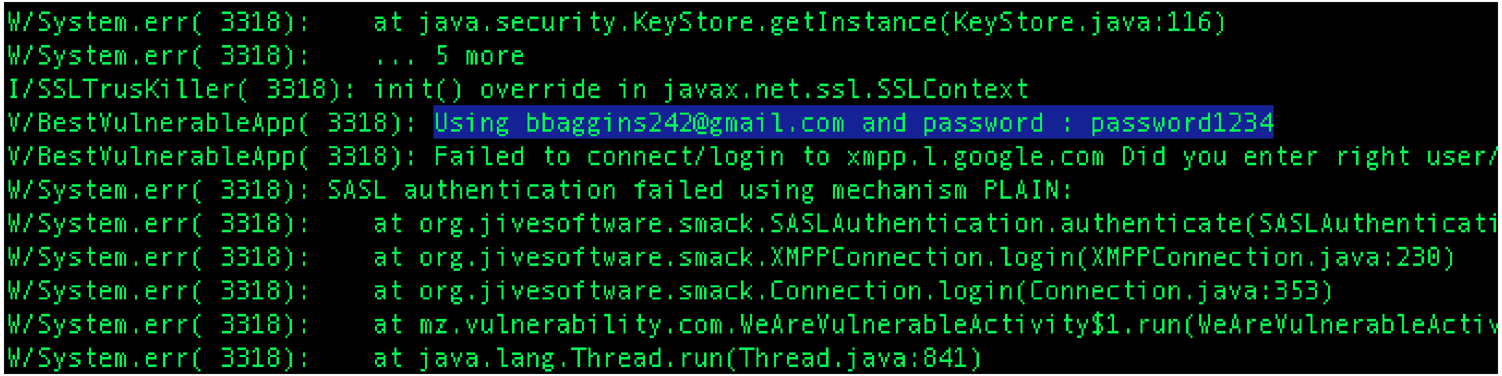
[redacted]

MP

⑆012345678⑆ ⑈98765432⑈

Logging

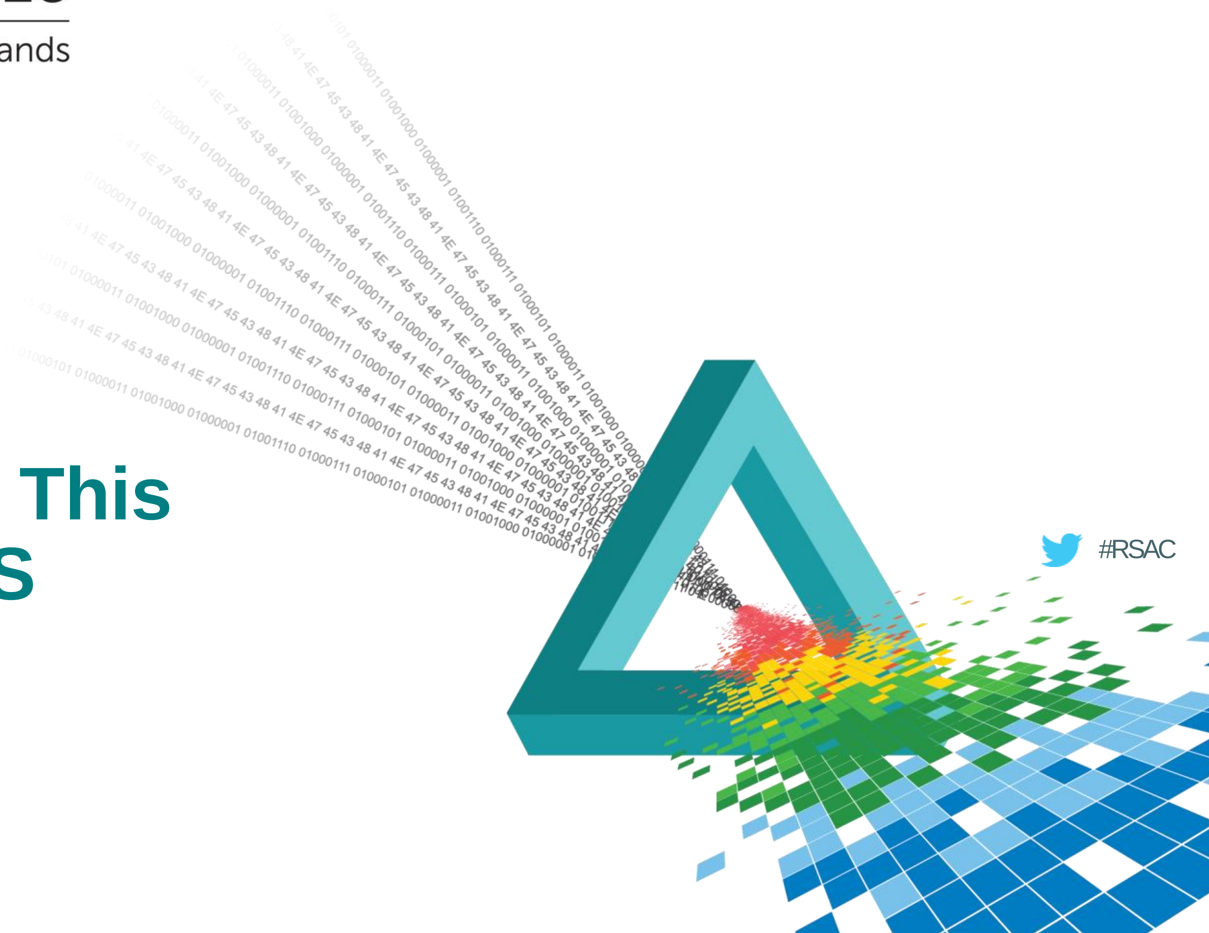
- Using Logcat



```
W/System.err( 3318):    at java.security.KeyStore.getInstance(KeyStore.java:116)
W/System.err( 3318):    ... 5 more
I/SSLTrusKiller( 3318): init() override in javax.net.ssl.SSLContext
V/BestVulnerableApp( 3318): Using bbaggins242@gmail.com and password : password1234
V/BestVulnerableApp( 3318): Failed to connect/login to xmpp.l.google.com Did you enter right user/
W/System.err( 3318): SASL authentication failed using mechanism PLAIN:
W/System.err( 3318):    at org.jivesoftware.smack.SASLAuthentication.authenticate(SASLAuthenticati
W/System.err( 3318):    at org.jivesoftware.smack.XMPPConnection.login(XMPPConnection.java:230)
W/System.err( 3318):    at org.jivesoftware.smack.Connection.login(Connection.java:353)
W/System.err( 3318):    at mz.vulnerability.com.WeAreVulnerableActivity$1.run(WeAreVulnerableActiv
W/System.err( 3318):    at java.lang.Thread.run(Thread.java:841)
```

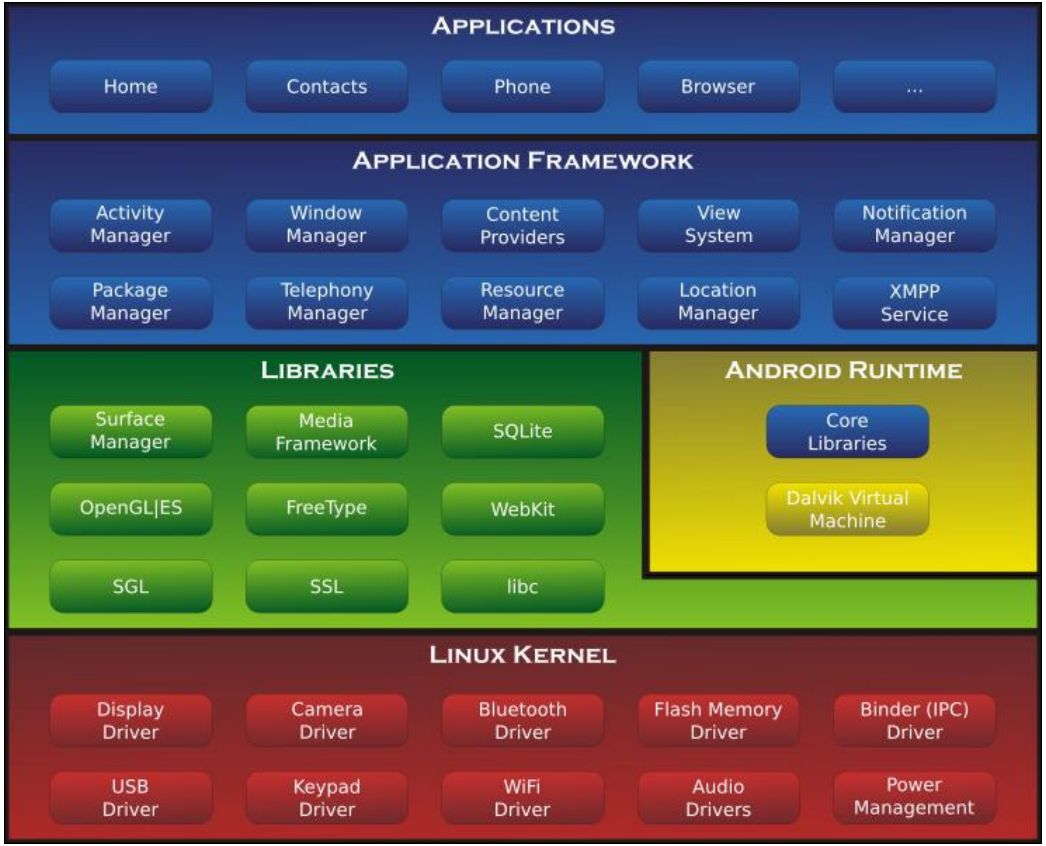
Security Through Obscurity?

How Do We Make This Easier, ShadowOS



There Must Be A Better Way

- There must be a better way to test mobile apps
- Needs to get around certificate pinning
- Watch files being created or modified real time
- Watch SQL queries being executed real time
- Android is open source, so how about we get inside the OS



APPLICATIONS

Home | Contacts | Phone | Browser | ...

APPLICATION FRAMEWORK

Activity Manager | Window Manager | Content Providers | View System | Notification Manager

Package Manager | Telephony Manager | Resource Manager | Location Manager | XMPP Service

LIBRARIES

Surface Manager | Media Framework | SQLite

OpenGL|ES | FreeType | WebKit

SGL | SSL | libc

ANDROID RUNTIME

Core Libraries

Dalvik Virtual Machine

LINUX KERNEL

Display Driver | Camera Driver | Bluetooth Driver | Flash Memory Driver | Binder (IPC) Driver

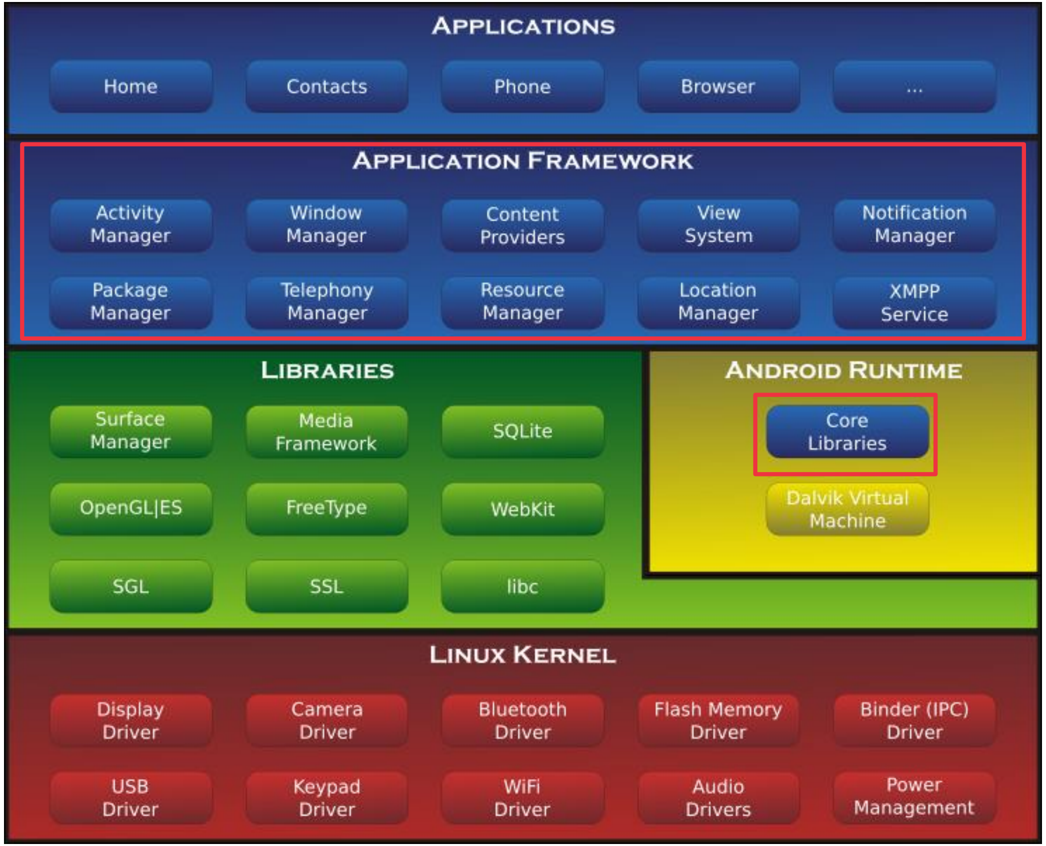
USB Driver | Keypad Driver | WiFi Driver | Audio Drivers | Power Management

Source: Wikipedia

Android OS

WebKit

SQLite



HTTPClient

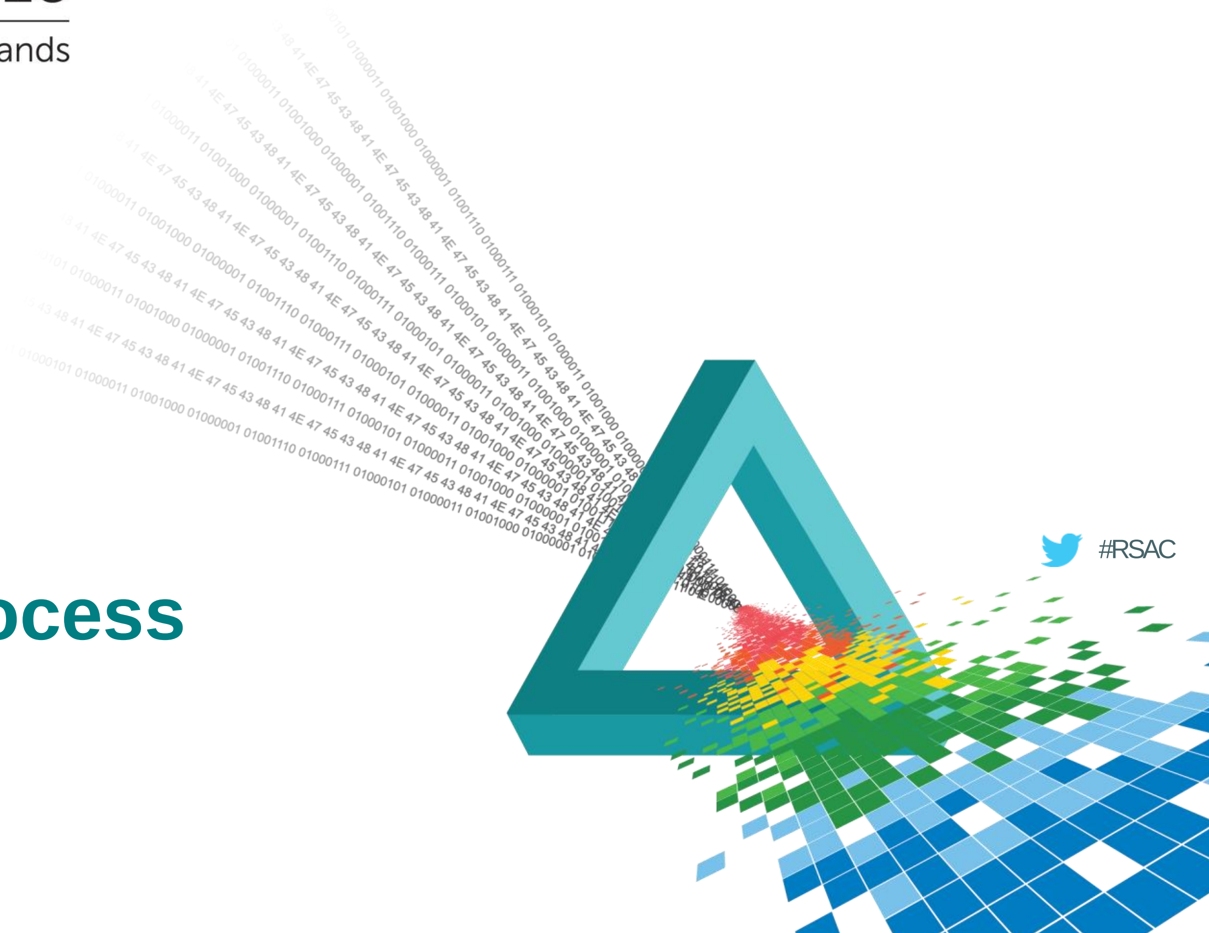
File Access

Source: Wikipedia

Android OS

RSAConference2015

Singapore | 22-24 July | Marina Bay Sands

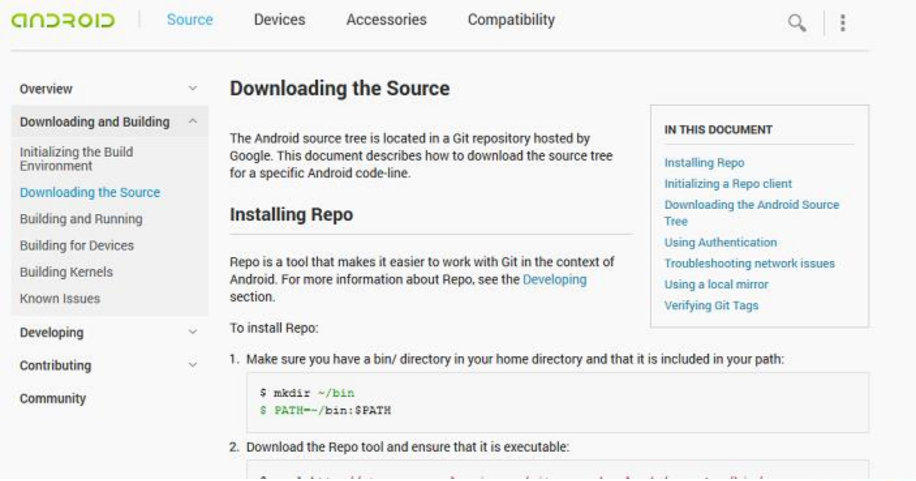


#RSAC

Android Build Process

The Host And Environment

- Ubuntu 12.04 64bit
- Sounds crazy, but follow the instructions!
- http://source.android.com/source/downloading.html



Downloading the Source

The Android source tree is located in a Git repository hosted by Google. This document describes how to download the source tree for a specific Android code-line.

IN THIS DOCUMENT

- Installing Repo
- Initializing a Repo client
- Downloading the Android Source Tree
- Using Authentication
- Troubleshooting network issues
- Using a local mirror
- Verifying Git Tags

Installing Repo

Repo is a tool that makes it easier to work with Git in the context of Android. For more information about Repo, see the Developing section.

To install Repo:

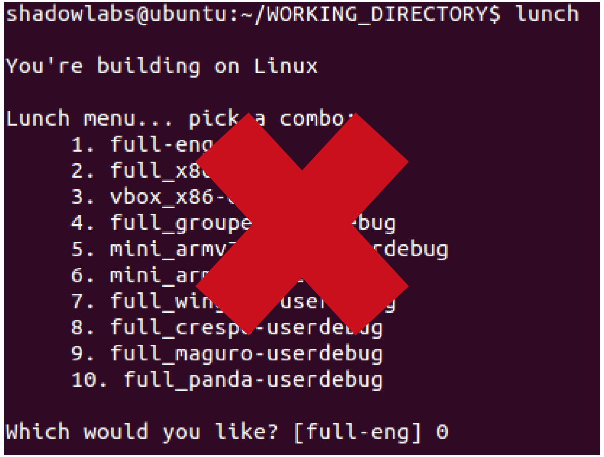
1. Make sure you have a bin/ directory in your home directory and that it is included in your path:

```
$ mkdir ~/bin
$ PATH=~/bin:$PATH
```

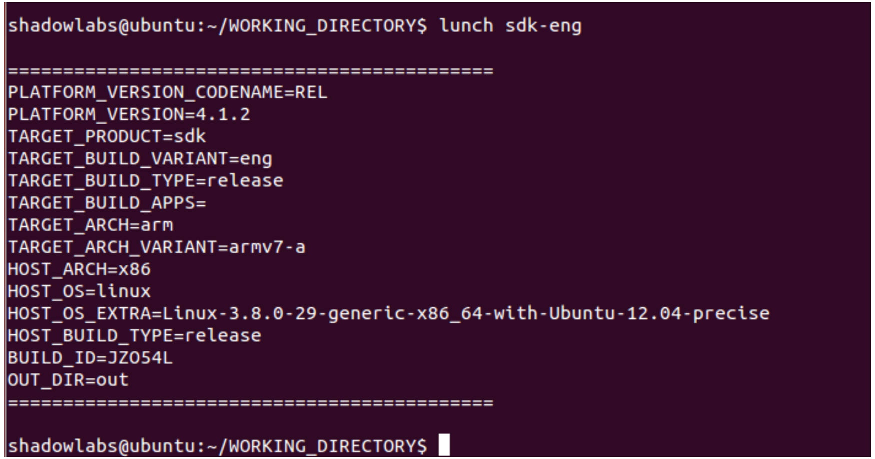
2. Download the Repo tool and ensure that it is executable:

Hidden Targets

- Run “lunch sdk-eng” to select the sdk target and images
- Don’t bother with the lunch menu



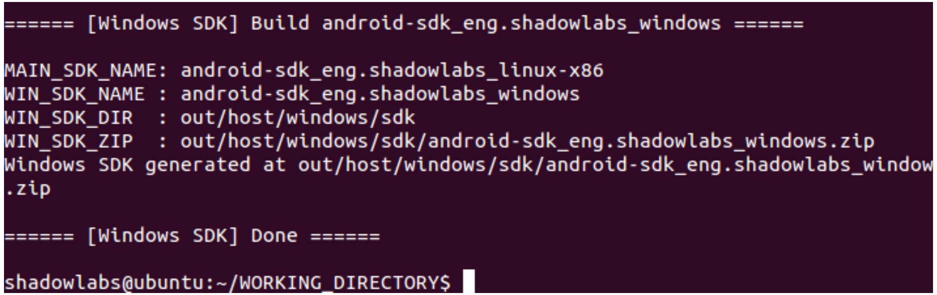
```
shadowlabs@ubuntu:~/WORKING_DIRECTORY$ lunch

You're building on Linux

Lunch menu... pick a combo:
     1. full-eng
     2. full_x8
     3. vbox_x86-
     4. full_groupe        -bug
     5. mini_armv          rdebug
     6. mini_ar
     7. full_win      user    g
     8. full_cresp  -userde  ug
     9. full_maguro-userdebug
     10. full_panda-userdebug

Which would you like? [full-eng] 0
```



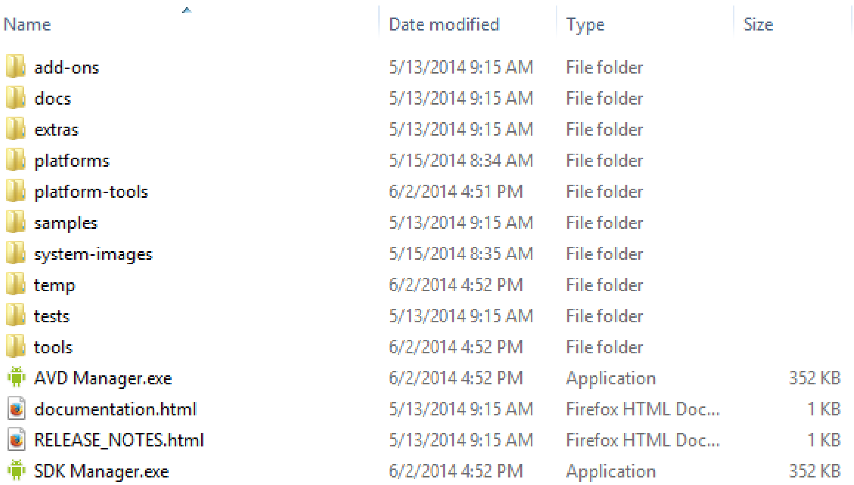
```
shadowlabs@ubuntu:~/WORKING_DIRECTORY$ lunch sdk-eng

============================================
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=4.1.2
TARGET_PRODUCT=sdk
TARGET_BUILD_VARIANT=eng
TARGET_BUILD_TYPE=release
TARGET_BUILD_APPS=
TARGET_ARCH=arm
TARGET_ARCH_VARIANT=armv7-a
HOST_ARCH=x86
HOST_OS=linux
HOST_OS_EXTRA=Linux-3.8.0-29-generic-x86_64-with-Ubuntu-12.04-precise
HOST_BUILD_TYPE=release
BUILD_ID=JZO54L
OUT_DIR=out
============================================

shadowlabs@ubuntu:~/WORKING_DIRECTORY$ █
```

Successful Build

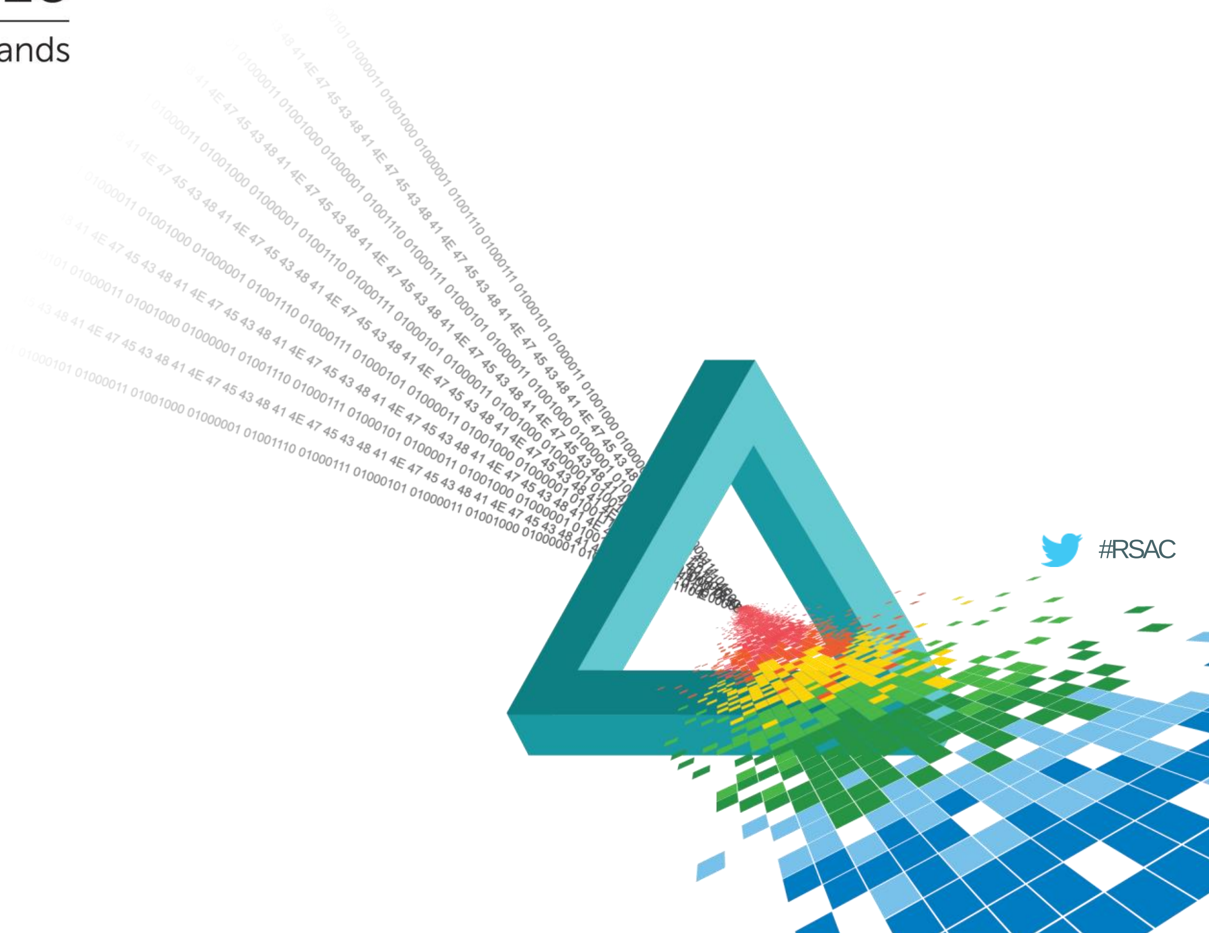
- Success!



```
====== [Windows SDK] Build android-sdk_eng.shadowlabs_windows ======

MAIN_SDK_NAME: android-sdk_eng.shadowlabs_linux-x86
WIN_SDK_NAME : android-sdk_eng.shadowlabs_windows
WIN_SDK_DIR  : out/host/windows/sdk
WIN_SDK_ZIP  : out/host/windows/sdk/android-sdk_eng.shadowlabs_windows.zip
Windows SDK generated at out/host/windows/sdk/android-sdk_eng.shadowlabs_window
.zip

====== [Windows SDK] Done ======

shadowlabs@ubuntu:~/WORKING_DIRECTORY$ █
```



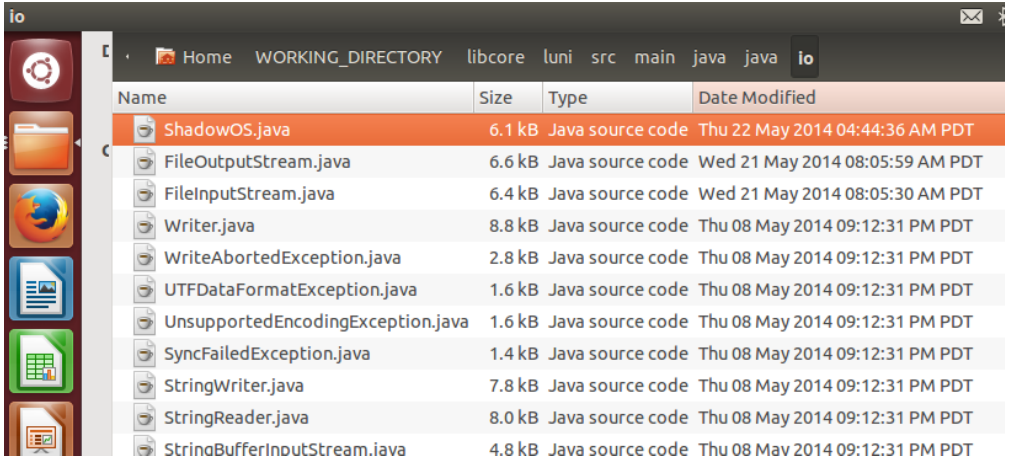
Name	Date modified	Type	Size
add-ons	5/13/2014 9:15 AM	File folder	
docs	5/13/2014 9:15 AM	File folder	
extras	5/13/2014 9:15 AM	File folder	
platforms	5/15/2014 8:34 AM	File folder	
platform-tools	6/2/2014 4:51 PM	File folder	
samples	5/13/2014 9:15 AM	File folder	
system-images	5/15/2014 8:35 AM	File folder	
temp	6/2/2014 4:52 PM	File folder	
tests	5/13/2014 9:15 AM	File folder	
tools	6/2/2014 4:52 PM	File folder	
AVD Manager.exe	6/2/2014 4:52 PM	Application	352 KB
documentation.html	5/13/2014 9:15 AM	Firefox HTML Doc...	1 KB
RELEASE_NOTES.html	5/13/2014 9:15 AM	Firefox HTML Doc...	1 KB
SDK Manager.exe	6/2/2014 4:52 PM	Application	352 KB



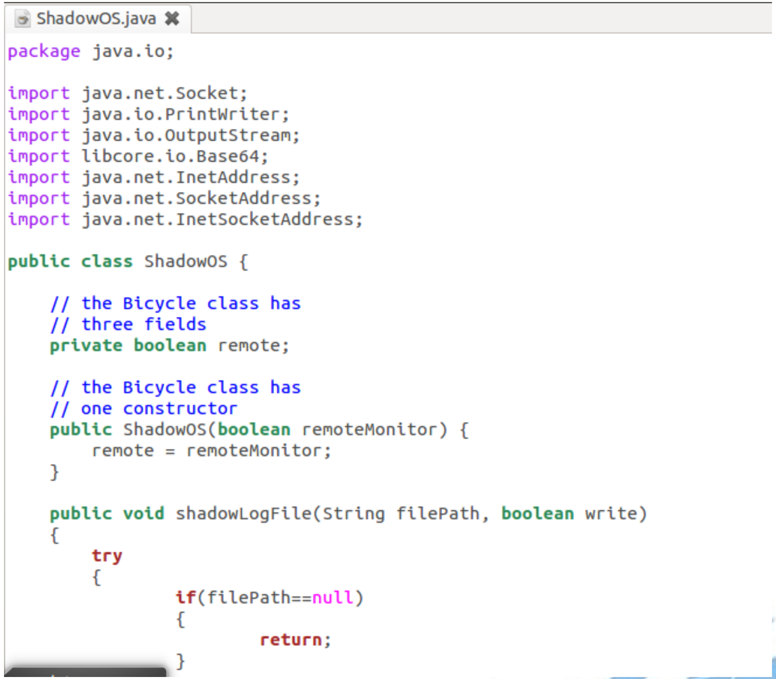
Modifications

Helper Class

- Common class for logging and monitoring
- Place class in java.io



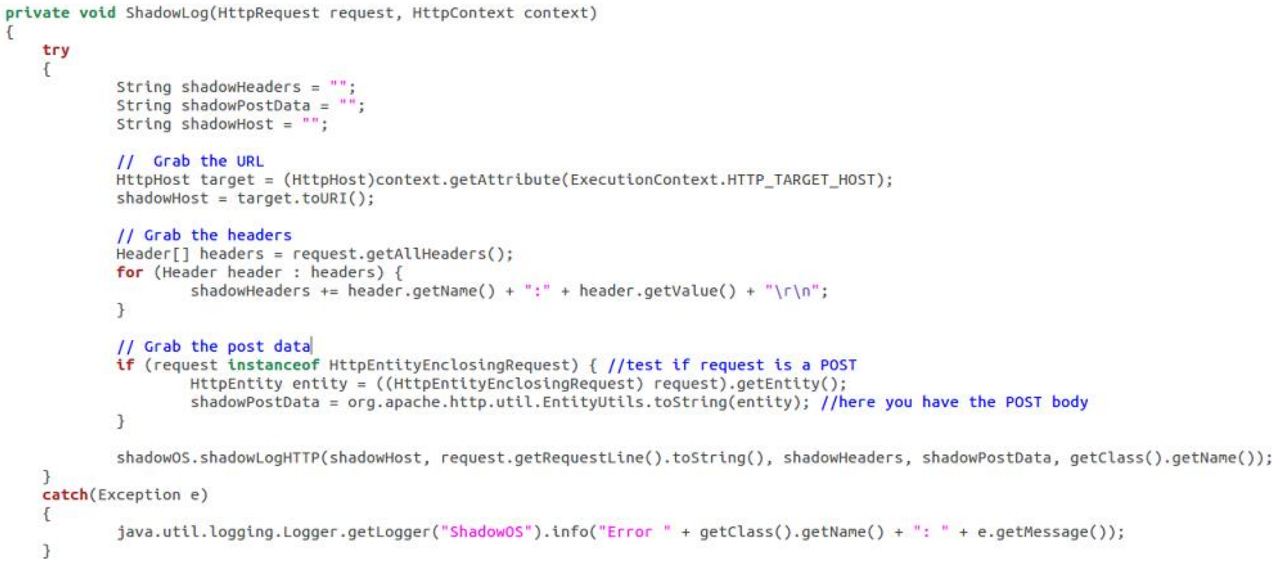
Name	Size	Type	Date Modified
ShadowOS.java	6.1 kB	Java source code	Thu 22 May 2014 04:44:36 AM PDT
FileOutputStream.java	6.6 kB	Java source code	Wed 21 May 2014 08:05:59 AM PDT
FileInputStream.java	6.4 kB	Java source code	Wed 21 May 2014 08:05:30 AM PDT
Writer.java	8.8 kB	Java source code	Thu 08 May 2014 09:12:31 PM PDT
WriteAbortedException.java	2.8 kB	Java source code	Thu 08 May 2014 09:12:31 PM PDT
UTFDataFormatException.java	1.6 kB	Java source code	Thu 08 May 2014 09:12:31 PM PDT
UnsupportedEncodingException.java	1.6 kB	Java source code	Thu 08 May 2014 09:12:31 PM PDT
SyncFailedException.java	1.4 kB	Java source code	Thu 08 May 2014 09:12:31 PM PDT
StringWriter.java	7.8 kB	Java source code	Thu 08 May 2014 09:12:31 PM PDT
StringReader.java	8.0 kB	Java source code	Thu 08 May 2014 09:12:31 PM PDT
StringBufferInputStream.java	4.8 kB	Java source code	Thu 08 May 2014 09:12:31 PM PDT



```java
package java.io;

import java.net.Socket;
import java.io.PrintWriter;
import java.io.OutputStream;
import libcore.io.Base64;
import java.net.InetAddress;
import java.net.SocketAddress;
import java.net.InetSocketAddress;

public class ShadowOS {

    // the Bicycle class has
    // three fields
    private boolean remote;

    // the Bicycle class has
    // one constructor
    public ShadowOS(boolean remoteMonitor) {
        remote = remoteMonitor;
    }

    public void shadowLogFile(String filePath, boolean write)
    {
        try
        {
                if(filePath==null)
                {
                        return;
                }
```

HTTP/HTTPS

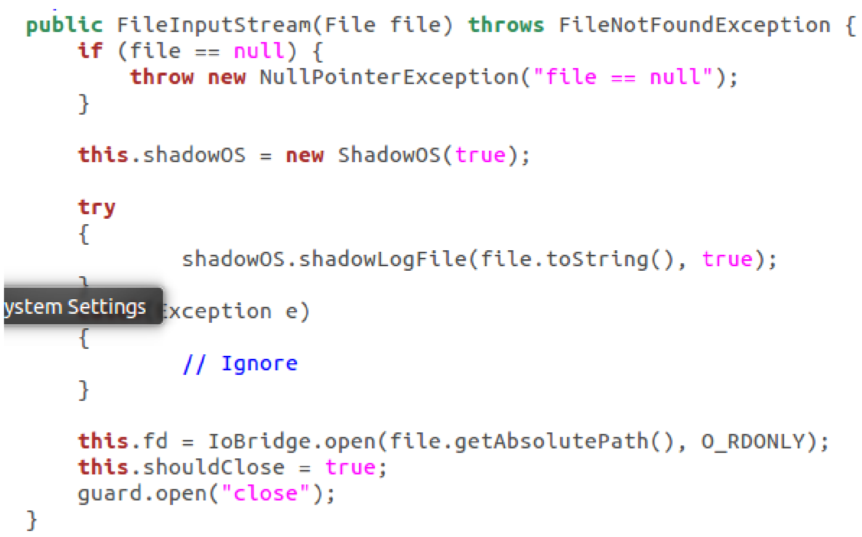
- There are a few places to capture HTTP traffic
- Most apps utilize Java.Net and Apache.HTTP



```java
private void ShadowLog(HttpRequest request, HttpContext context)
{
    try
    {
            String shadowHeaders = "";
            String shadowPostData = "";
            String shadowHost = "";

            //  Grab the URL
            HttpHost target = (HttpHost)context.getAttribute(ExecutionContext.HTTP_TARGET_HOST);
            shadowHost = target.toURI();

            // Grab the headers
            Header[] headers = request.getAllHeaders();
            for (Header header : headers) {
                    shadowHeaders += header.getName() + ":" + header.getValue() + "\r\n";
            }

            // Grab the post data
            if (request instanceof HttpEntityEnclosingRequest) { //test if request is a POST
                    HttpEntity entity = ((HttpEntityEnclosingRequest) request).getEntity();
                    shadowPostData = org.apache.http.util.EntityUtils.toString(entity); //here you have the POST body
            }

            shadowOS.shadowLogHTTP(shadowHost, request.getRequestLine().toString(), shadowHeaders, shadowPostData, getClass().getName());
    }
    catch(Exception e)
    {
            java.util.logging.Logger.getLogger("ShadowOS").info("Error " + getClass().getName() + ": " + e.getMessage());
    }
```

File System

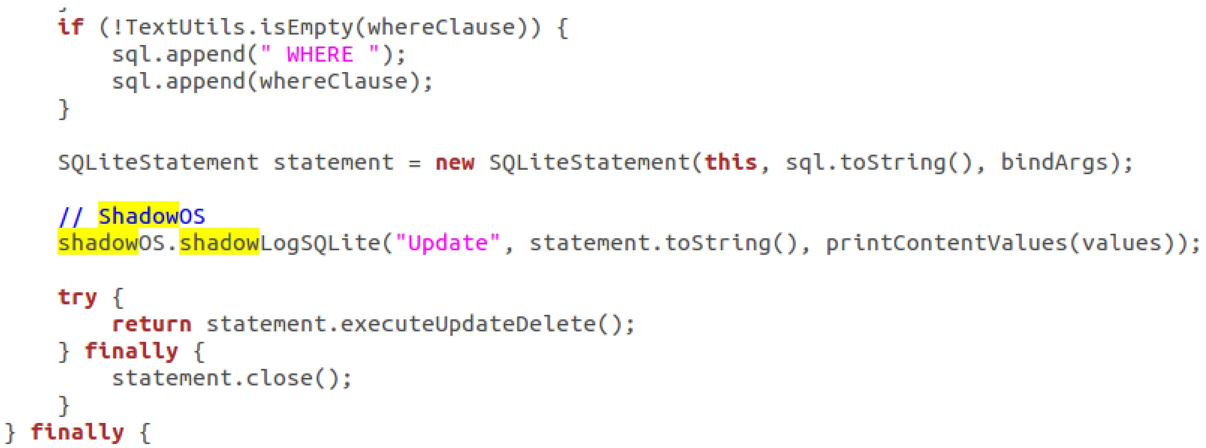
- Common read/write functions
- FileInputStream/FileOutputStream



```
public FileInputStream(File file) throws FileNotFoundException {
    if (file == null) {
        throw new NullPointerException("file == null");
    }

    this.shadowOS = new ShadowOS(true);

    try
    {
            shadowOS.shadowLogFile(file.toString(), true);
    }
              xception e)
    {
            // Ignore
    }

    this.fd = IoBridge.open(file.getAbsolutePath(), O_RDONLY);
    this.shouldClose = true;
    guard.open("close");
}
```

SQLite

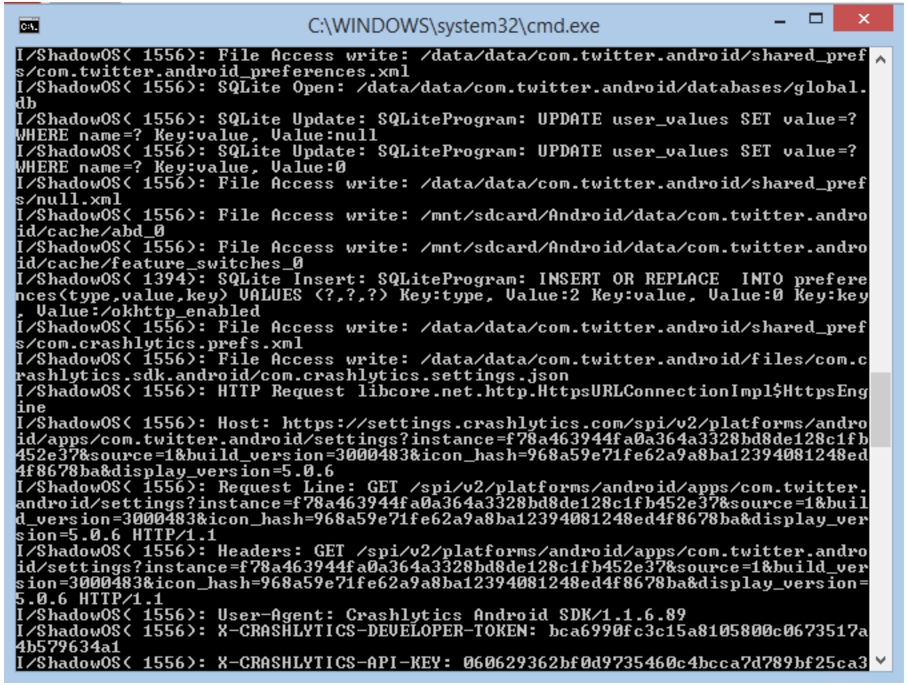
- One main class, SQLiteDatabase.java
- Intercept Open, Insert and Update



```
    if (!TextUtils.isEmpty(whereClause)) {
        sql.append(" WHERE ");
        sql.append(whereClause);
    }

    SQLiteStatement statement = new SQLiteStatement(this, sql.toString(), bindArgs);

    // ShadowOS
    shadowOS.shadowLogSQLite("Update", statement.toString(), printContentValues(values));

    try {
        return statement.executeUpdateDelete();
    } finally {
        statement.close();
    }
} finally {
```

Using Logcat

- adb.exe logcat -s "ShadowOS"



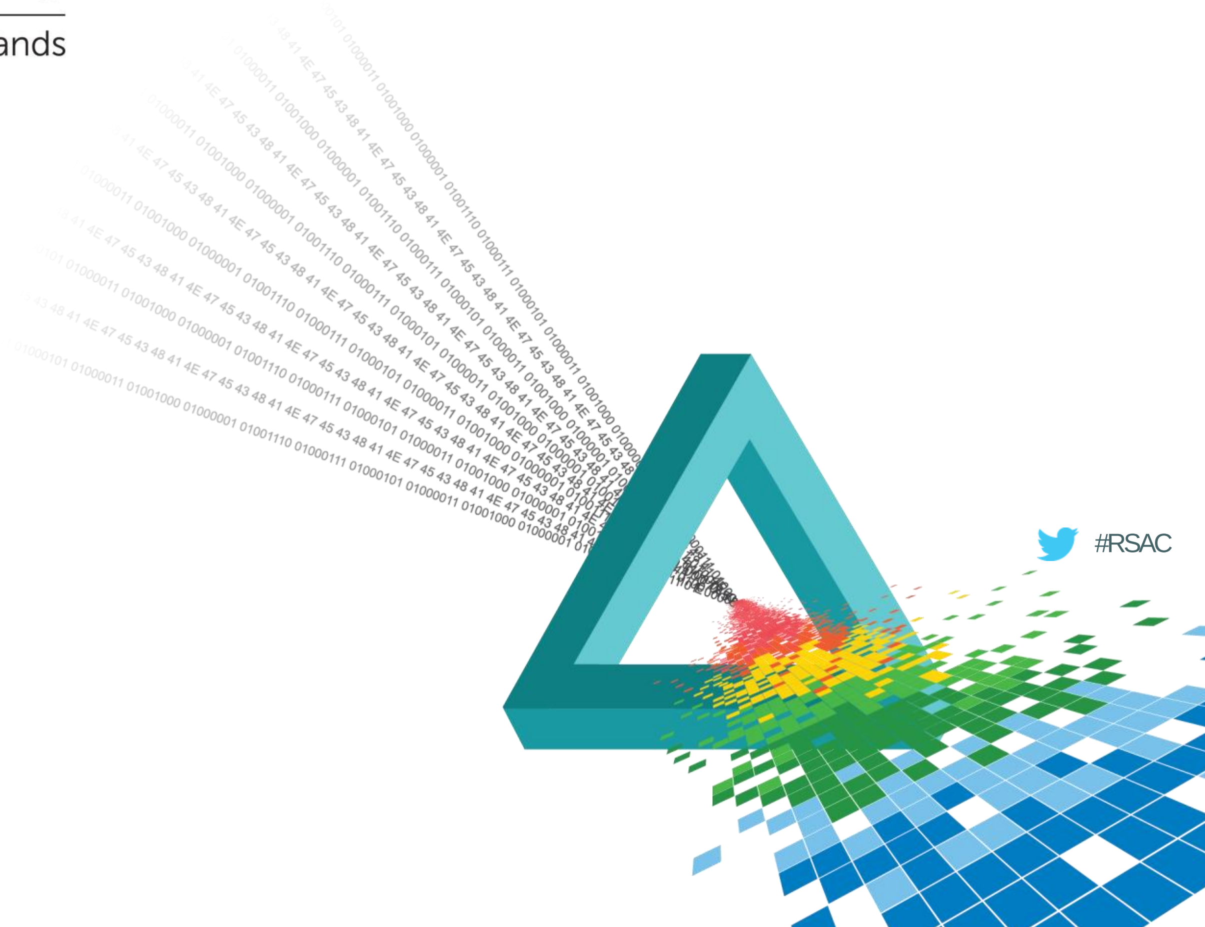
```
I/ShadowOS( 1556): File Access write: /data/data/com.twitter.android/shared_pref
s/com.twitter.android_preferences.xml
I/ShadowOS( 1556): SQLite Open: /data/data/com.twitter.android/databases/global.
db
I/ShadowOS( 1556): SQLite Update: SQLiteProgram: UPDATE user_values SET value=?
WHERE name=? Key:value, Value:null
I/ShadowOS( 1556): SQLite Update: SQLiteProgram: UPDATE user_values SET value=?
WHERE name=? Key:value, Value:0
I/ShadowOS( 1556): File Access write: /data/data/com.twitter.android/shared_pref
s/null.xml
I/ShadowOS( 1556): File Access write: /mnt/sdcard/Android/data/com.twitter.andro
id/cache/abd_0
I/ShadowOS( 1556): File Access write: /mnt/sdcard/Android/data/com.twitter.andro
id/cache/feature_switches_0
I/ShadowOS( 1394): SQLite Insert: SQLiteProgram: INSERT OR REPLACE  INTO prefere
nces(type,value,key) VALUES (?,?,?) Key:type, Value:2 Key:value, Value:0 Key:key
, Value:/okhttp_enabled
I/ShadowOS( 1556): File Access write: /data/data/com.twitter.android/shared_pref
s/com.crashlytics.prefs.xml
I/ShadowOS( 1556): File Access write: /data/data/com.twitter.android/files/com.c
rashlytics.sdk.android/com.crashlytics.settings.json
I/ShadowOS( 1556): HTTP Request libcore.net.http.HttpsURLConnectionImpl$HttpsEng
ine
I/ShadowOS( 1556): Host: https://settings.crashlytics.com/spi/v2/platforms/andro
id/apps/com.twitter.android/settings?instance=f78a463944fa0a364a3328bd8de128c1fb
452e37&source=1&build_version=3000483&icon_hash=968a59e71fe62a9a8ba12394081248ed
4f8678ba&display_version=5.0.6
I/ShadowOS( 1556): Request Line: GET /spi/v2/platforms/android/apps/com.twitter.
android/settings?instance=f78a463944fa0a364a3328bd8de128c1fb452e37&source=1&buil
d_version=3000483&icon_hash=968a59e71fe62a9a8ba12394081248ed4f8678ba&display_ver
sion=5.0.6 HTTP/1.1
I/ShadowOS( 1556): Headers: GET /spi/v2/platforms/android/apps/com.twitter.andro
id/settings?instance=f78a463944fa0a364a3328bd8de128c1fb452e37&source=1&build_ver
sion=3000483&icon_hash=968a59e71fe62a9a8ba12394081248ed4f8678ba&display_version=
5.0.6 HTTP/1.1
I/ShadowOS( 1556): User-Agent: Crashlytics Android SDK/1.1.6.89
I/ShadowOS( 1556): X-CRASHLYTICS-DEVELOPER-TOKEN: bca6990fc3c15a8105800c0673517a
4b579634a1
I/ShadowOS( 1556): X-CRASHLYTICS-API-KEY: 060629362bf0d9735460c4bcca7d789bf25ca3
```

Remote Monitoring

- Using socket connection to specific port
- Data formatted in XML
- Host loopback (127.0.0.1) is 10.0.2.2

RSA®Conference2015

Singapore | 22-24 July | Marina Bay Sands



#RSAC

Demonstration

Apply What You Have Learned Today

- Download and try ShadowOS
- Think of new ideas for areas of interception
- Think of new visualization of captured data
- Submit ideas to ShadowOS@hp.com

RSAConference2015

Singapore | 22-24 July | Marina Bay Sands

CHANGE

Challenge today's security thinking

SESSION ID: MBS-R03

ShadowOS: Modifying the Android OS for Mobile Application Testing

Ray Kelly

Research and Innovation
HP Fortify On Demand
@vbisbest