# Pentesting Android Mobile Application

## Overview on Mobile applications

**Connect in Superior Way!!**

Mobile market is the worldwide rapidly developing segments since many customers are using mobile phones. Now-a-days mobile phones have become more like portable computers which supports each and every applications like Internet banking, travel portal, e-commerce etc. Mobile applications have started attracting hackers as mobile applications are also involved in money transactions.

These applications run on different Operating System such as Symbian, Windows, IPhone, IPad and Android Mobile. The testing procedure varies based on the supporting OS, however in case of web application the testing procedure remains the same; irrespective of application platform like ASP. Net, PHP, JSP, etc.

## Approach

In this article we'll look how we can proceed for Android Mobile application testing. Testing is focused on three categories:-

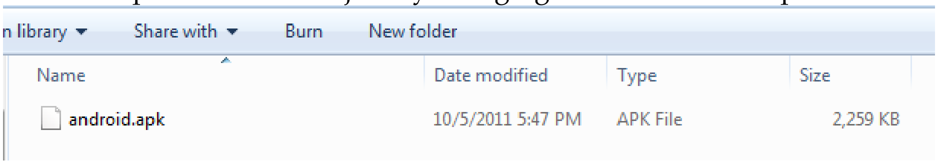
- Reverse Engineering
- Intercepting Mobile Application Traffic
- Memory Analysis

### Reverse Engineering

Reverse Engineering is a process to analyze the source code, find sensitive data, exploit and modify code.

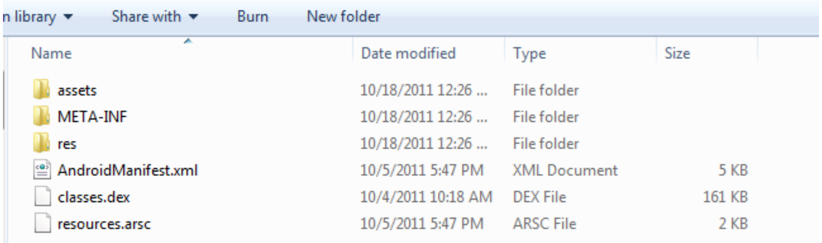
In general, for Symbian application if you extract .Jar files you will get access to compiled code and can begin the analysis but in case of android specific .apk file, follow the below steps to extract complied codes from .apk file.

1. Convert .apk file to .rar file just by changing the extension of .apk to .rar or .zip



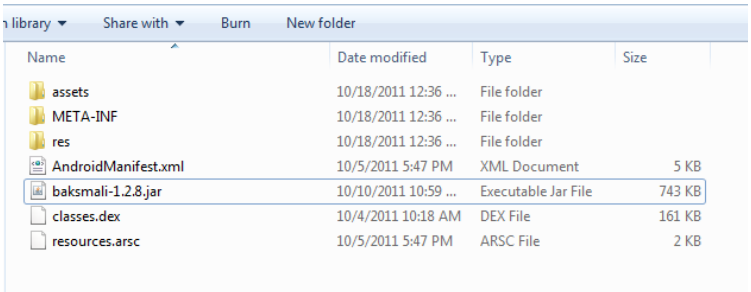
| Name | Date modified | Type | Size |
|---|---|---|---|
| android.apk | 10/5/2011 5:47 PM | APK File | 2,259 KB |

2. Extract .rar file
3. Under the extracted folder you will get classes.dex file, extract compiled code files from classes.dex file



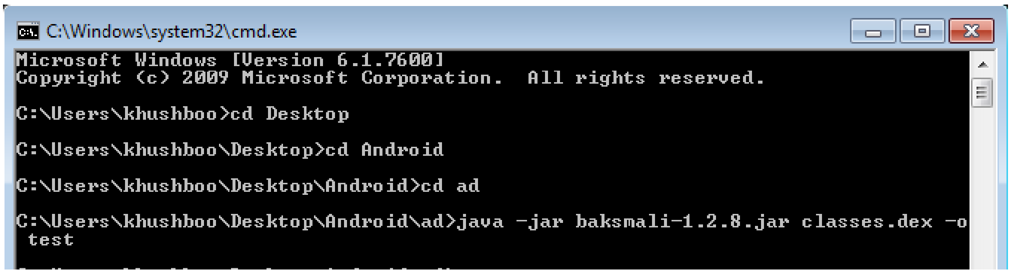
| Name | Date modified | Type | Size |
|---|---|---|---|
| assets | 10/18/2011 12:26 ... | File folder | |
| META-INF | 10/18/2011 12:26 ... | File folder | |
| res | 10/18/2011 12:26 ... | File folder | |
| AndroidManifest.xml | 10/5/2011 5:47 PM | XML Document | 5 KB |
| classes.dex | 10/4/2011 10:18 AM | DEX File | 161 KB |
| resources.arsc | 10/5/2011 5:47 PM | ARSC File | 2 KB |

4. Download baksmali-1.2.8.jar and save this jar file under extracted folder.



| Name | Date modified | Type | Size |
|---|---|---|---|
| assets | 10/18/2011 12:36 ... | File folder | |
| META-INF | 10/18/2011 12:36 ... | File folder | |
| res | 10/18/2011 12:36 ... | File folder | |
| AndroidManifest.xml | 10/5/2011 5:47 PM | XML Document | 5 KB |
| baksmali-1.2.8.jar | 10/10/2011 10:59 ... | Executable Jar File | 743 KB |
| classes.dex | 10/4/2011 10:18 AM | DEX File | 161 KB |
| resources.arsc | 10/5/2011 5:47 PM | ARSC File | 2 KB |

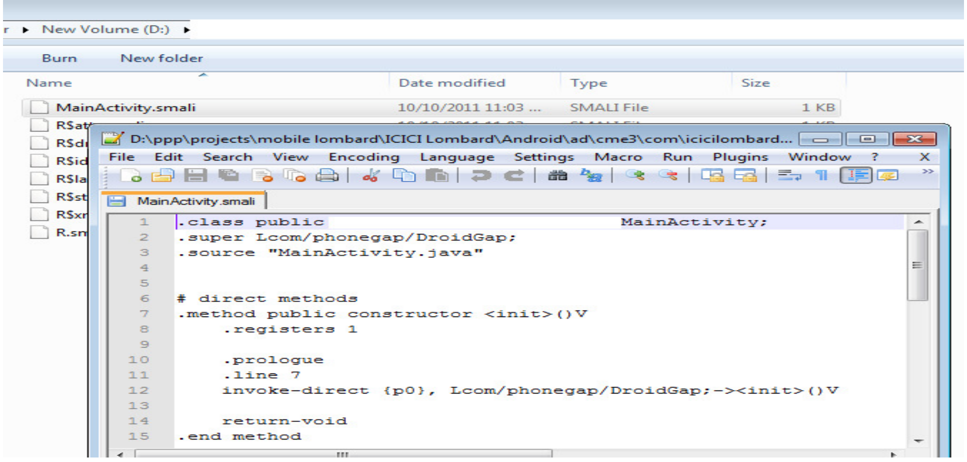
5. Run the baksmali. Jar file using command prompt to extract compiled code files from classes.dex file by giving the below command:-
java -jar baksmali-1.2.8.jar classes.dex -o test



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\khushboo>cd Desktop

C:\Users\khushboo\Desktop>cd Android

C:\Users\khushboo\Desktop\Android>cd ad

C:\Users\khushboo\Desktop\Android\ad>java -jar baksmali-1.2.8.jar classes.dex -o
 test
```

6. Now you can view .smali files under ‘test’ folder.

7. Now .smali files can be opened in any text editor to view the code and do the analysis



| Name | Date modified | Type | Size |
|---|---|---|---|
| MainActivity.smali | 10/10/2011 11:03 ... | SMALI File | 1 KB |
| R$at | | | |
| R$d | | | |
| R$id | | | |
| R$la | | | |
| R$st | | | |
| R$xr | | | |
| R.sm | | | |

```
.class public                               MainActivity;
.super Lcom/phonegap/DroidGap;
.source "MainActivity.java"


# direct methods
.method public constructor <init>()V
    .registers 1

    .prologue
    .line 7
    invoke-direct {p0}, Lcom/phonegap/DroidGap;-><init>()V

    return-void
.end method
```

8. Agnitio is a good tool by which code analysis can be done

9. After analysis of code, try to write or modify the validation present in the source code and again compile it.

10. After doing the changes run that modified compiled code and check whether changes made by you in code get implemented.

## Pen testing using Handset and Proxy Tool

### Testing Via Connecting With Wi-Fi:-

Let's now test android application using Wi-Fi method:

In this method we have to divert the traffic from handset to server via proxy of your system. To divert traffic we also require setting up proxy and other tools on android handset and for this root privilege is required on the handset. Rooting is the process of getting root access on handset. Rooting is like jailbreaking in Iphone.

Here in this article we are taking example of rooting Samsung S2 handset

Download following tools before starting the rooting process:-

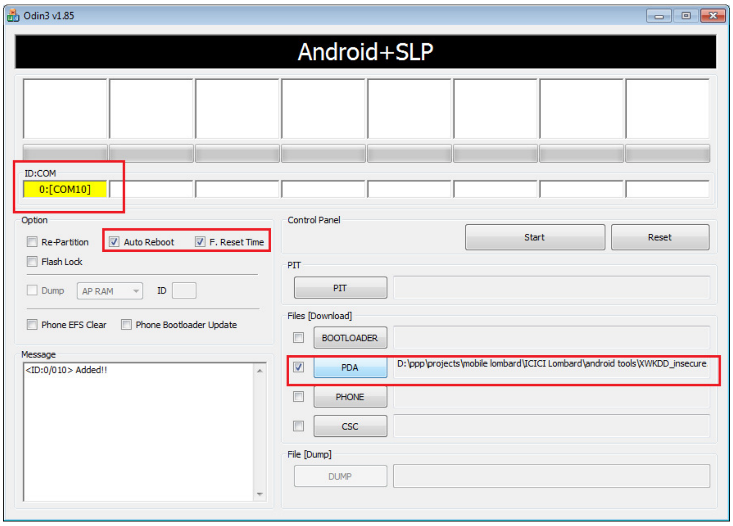
- Odin Downloader
- Download the XWKDD, but do NOT extract the .tar file
- Download and extract SuperOneClick
- Keep the handset in USB debugging mode Settings -> Applications -> Development -> USB debugging

### Steps for Rooting Samsung S2 Handset

1. Reboot the handset into download mode, to do this turn off the device and power on it again by pressing Volume Down + Home + Power button simultaneously.

2. To start Run 'ODIN Downloader' run 'Odin3 v1.85.exe'

3. Now connect handset to the PC using USB cable. Now tick the checkboxes of 'Auto reboot', 'F. Reset Time' and 'PDA' in ODIN and press the 'PDA' button to browse 'XWKDD_insecure.tar' file. Wait for few seconds and on the top left corner you will notice a symbol stating that device is now connected, and then click on start button.



ID:COM 0:[COM10]

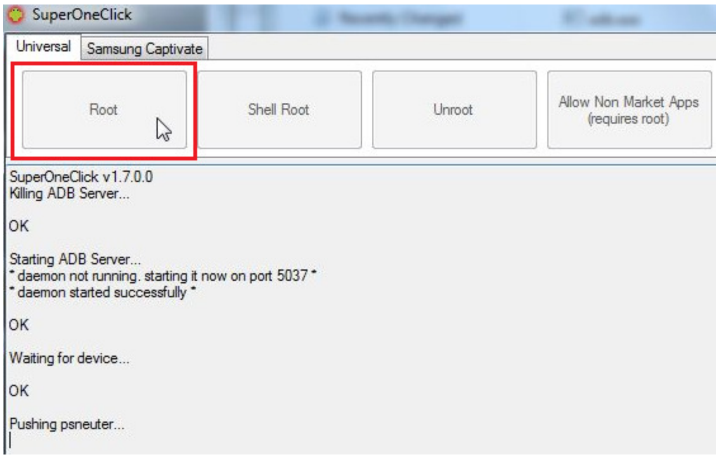
□ Re-Partition ☑ Auto Reboot ☑ F. Reset Time
□ Flash Lock
□ Dump AP RAM ID
□ Phone EFS Clear □ Phone Bootloader Update

Message
<ID:0/010> Added!!

□ BOOTLOADER
☑ PDA D:\ppp\projects\mobile lombard\ICICI Lombard\android tools\XWKDD_insecure
□ PHONE
□ CSC

4. After this device will reboot. Do not remove the USB cable

5. Once the handset is rebooted start 'SuperOneClick' just by running 'SuperOneClick.exe' , and press "ROOT " button and reboot the handset



```
SuperOneClick v1.7.0.0
Killing ADB Server...

OK

Starting ADB Server...
* daemon not running. starting it now on port 5037 *
* daemon started successfully *

OK

Waiting for device...

OK

Pushing psneuter...
```

6. Now your Handset is successfully rooted

Once the handset is rooted you are now able to download testing tools and start your testing

## Steps to divert the traffic via PC

1. To do this you will need a Wireless Router.
2. Connect the wireless router to the LAN network.
3. Connect your PC to the wireless router
4. Enable the Wireless router on your handset
5. Download Proxy tool from android market like Droid proxy in the handset
6. Now to set up the proxy. Enter the IP address which is assigned to your PC after connecting it with the wireless e.g. 192.168.1.6 and also set the port to 8080 or to 5555 in it.
7. Now open burp or Paros proxy in your PC and set the same port number as you set in your handset (8080 or 5555)
8. After completing the above settings when you browse through mobile all the traffic will divert through the burp/Paros proxy of your system.

Now you can start the testing as you do for web application. You can try various attacks like Parameter Manipulation, SQL Injection, etc.

## Handset Memory Analysis

Download terminal emulator from android market. A terminal emulator is a program that makes your Android phone act like an old fashioned computer terminal which is useful for accessing the Linux command line shell that is built into every Android phone. Using terminal emulator look for sensitive information present in different folders like giving following commands:

Ls <enter>

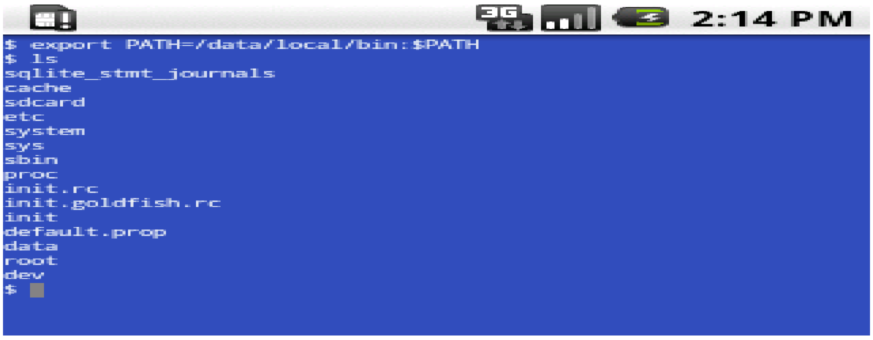
Cd data <enter>

Ls <enter>

Cd data <enter>

Ls <enter>

Cd apps <enter



```
$ export PATH=/data/local/bin:$PATH
$ ls
sqlite_stmt_journals
cache
sdcard
etc
system
sys
sbin
proc
init.rc
init.goldfish.rc
init
default.prop
data
root
dev
$
```

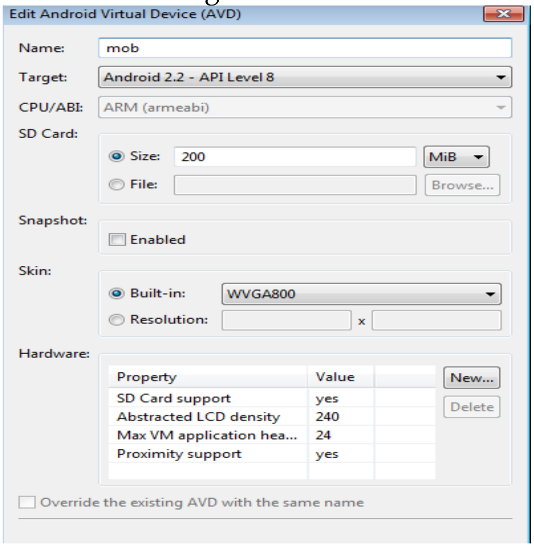
## Pen Testing using Phone Emulator and Proxy Tool

### Testing using emulator

An emulator is a software application which allows a computer to run programs written for Mobile devices. Device specific emulator can be freely downloaded from internet.  There is no need of handset if you are testing using emulator.

### Steps to setup Emulator

1. Download Android SDK 2.2
2. Run 'SDK Manager.exe'. Now create a new 'Virtual device'.



Edit Android Virtual Device (AVD)

Name: mob

Target: Android 2.2 - API Level 8

CPU/ABI: ARM (armeabi)

SD Card:

● Size: 200 MiB

○ File: Browse...

Snapshot:

□ Enabled

Skin:

● Built-in: WVGA800

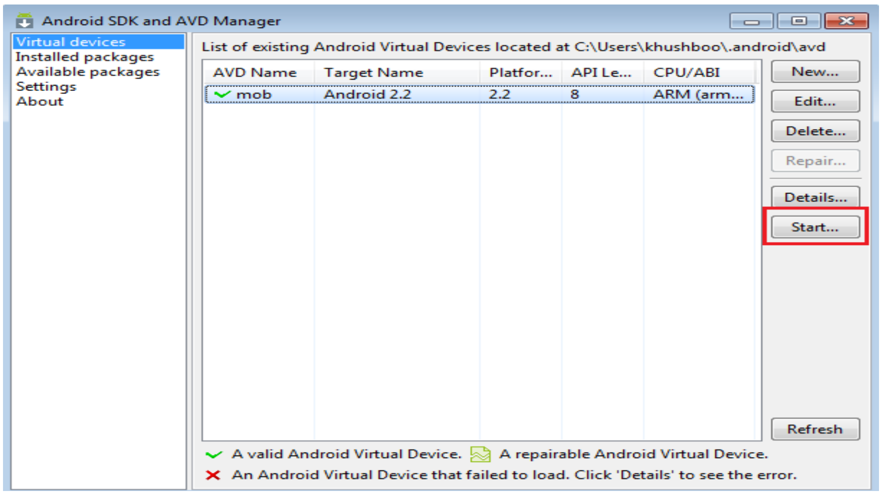
○ Resolution: x

Hardware:

| Property | Value |
|---|---|
| SD Card support | yes |
| Abstracted LCD density | 240 |
| Max VM application hea... | 24 |
| Proximity support | yes |

□ Override the existing AVD with the same name

3. Select the created virtual device and click on 'start' button and then on 'Launch' button.



Android SDK and AVD Manager

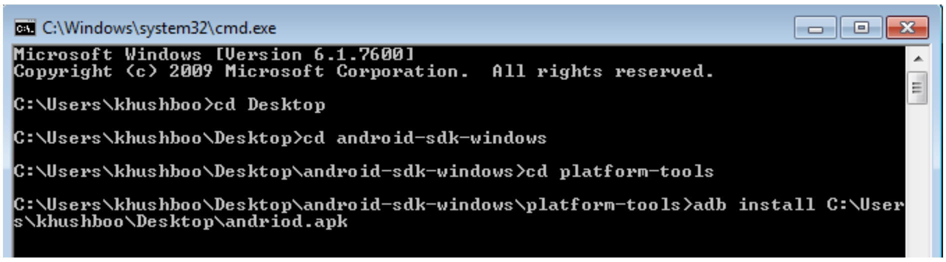
List of existing Android Virtual Devices located at C:\Users\khushboo\.android\avd

| AVD Name | Target Name | Platfor... | API Le... | CPU/ABI |
|---|---|---|---|---|
| ✓ mob | Android 2.2 | 2.2 | 8 | ARM (arm... |

✓ A valid Android Virtual Device. A repairable Android Virtual Device.

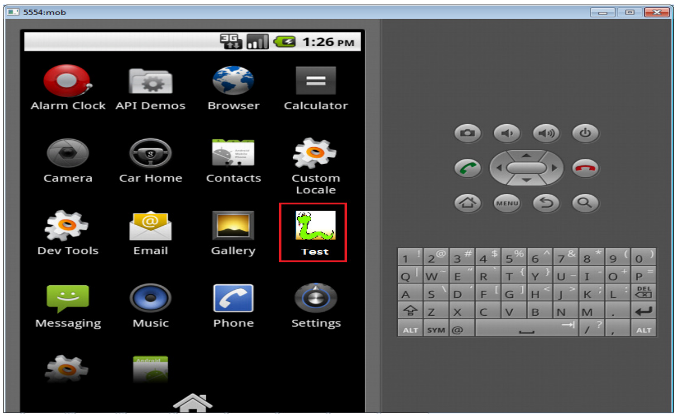
✗ An Android Virtual Device that failed to load. Click 'Details' to see the error.

4. Now you have to install android application on emulator for this open command prompt on your PC and enter following commands
   C:\Users\khushboo\Desktop\android-sdk-windows\platform-tools>adb install C:\User s\khushboo\Desktop\andriod.apk



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\khushboo>cd Desktop

C:\Users\khushboo\Desktop>cd android-sdk-windows

C:\Users\khushboo\Desktop\android-sdk-windows>cd platform-tools

C:\Users\khushboo\Desktop\android-sdk-windows\platform-tools>adb install C:\User
s\khushboo\Desktop\andriod.apk
```

   New application gets installed on emulator

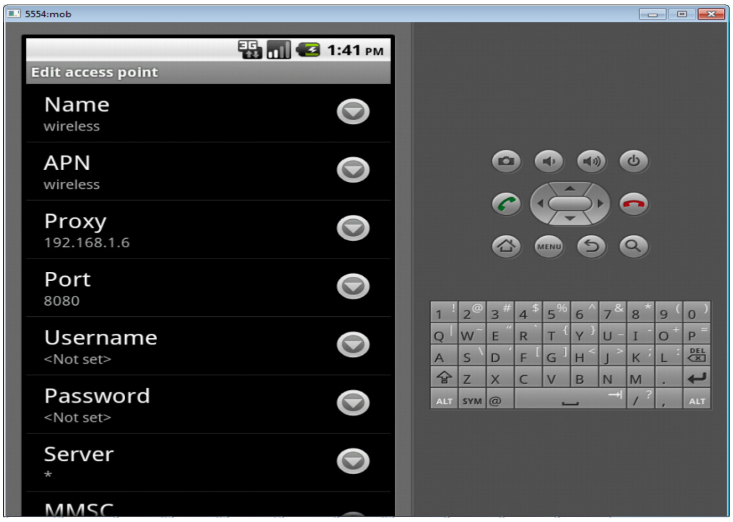


5. If in case you get error message while installation enter following commands
   adb kill-server
   adb start-server

Now you can access the application as you do using your Samsung S2 handset

## Steps to setup Proxy

1. To set up proxy in the emulator follow the process Home -> Menu -> Settings -> Wireless & Networks -> Mobile Networks -> Access Points Names and update the settings
   - Name: wireless
   - APN: Wireless
   - Proxy: IP address of your PC
   - Username: <Not set>
   - Password: <Not set>



Name
wireless

APN
wireless

Proxy
192.168.1.6

Port
8080

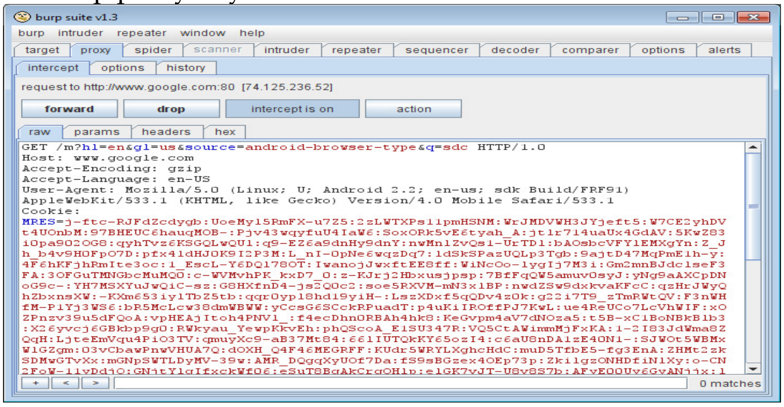
Username
<Not set>

Password
<Not set>

Server
*

MMSC

2. Open burp proxy in your PC and set the port as 8080 or 5555
3. After doing the above settings when you browse through emulator all the traffic will go through the burp proxy of your machine


burp suite v1.3
burp intruder repeater window help
target proxy spider scanner intruder repeater sequencer decoder comparer options alerts
intercept options history
forward
drop
intercept is on
action
raw params headers hex
+ < >
0 matches

request to http://www.google.com:80 [74.125.236.52]

```
GET /m?hl=en&gl=us&source=android-browser-type&q=sdc HTTP/1.0
Host: www.google.com
Accept-Encoding: gzip
Accept-Language: en-US
User-Agent: Mozilla/5.0 (Linux; U; Android 2.2; en-us; sdk Build/FRF91)
AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1
Cookie:
MRES=j-ftc-RJFdZcdygb:UoeMy15RmFX-u7Z5:2zLWTXPs11pmHSNM:WrJMDVWH3JYjeft5:W7CE2yhDV
t4UOnbM:97BHEUC6hauqMOB-:Pjv43wqyfuU4IaW6:SoxORk5vE6tyah_A:jt1r714uaUx4GdAV:5KwZ83
i0pa902OG8:qyhTvz6KSGQLwQU1:q9-EZ6a9dnHy9dnY:nwMn1ZvQs1-UrTD1:bAOsbcVFY1EMXgYn:Z_J
h_b4v9HOFpO7D:pfx41dHJOK9I2P3M:L_nI-OpNe6wqzDq7:1dSkSPazUQLp3Tgb:9ajtD47MqPmE1h-y:
4F6hKFjhRmIte3oc:1_EscL-Y6DQ178OT:IwanojJwxftEE8ff:WiNcOo-1ygIj7M3i:Gm2mnBJdc1seF3
FA:3OFGuTMNGbcMuMQ0:c-WVMvhPK_kxD7_0:z-KJrj2Hbxusjpsp:7BfFqQW5amuv0syJ:yNg9aAXCpDN
oG9c-:YH7MSXYuJwQiC-sz:G8HXfnD4-js2Q0c2:soe5RXVM-mN3x1BP:nwdZSw9dxkvaKFcC:qzHrJWyQ
hZbxnsXW:-KXm653iy1TbZ5tb:qqr0yp18hd19yiH-:LszXDxf5qQDv4z0k:g22i7T9_zTmRWtQV:F3nWH
fM-P1Yj3WS6:bR5McLcw38dmWBWW:yCcsG6SCckRPuadT:p4uKiIROffPJ7KwL:ue4ReUCo7LcVhWIF:xO
ZPnzv39u5dFQoA:vpHEAjItoh4PNV1_:f4ecDhnORBAh4hk8:KeGvpm4aV7dNOza5:t5B-xC1BoNBkB1b3
:X26yvcj6GBkbp9g0:RWkyau_YewpKkvEh:phQScoA_E1SU347R:VQ5CtAWimmMjFxKA:1-2I83JdWma8Z
QqH:LjteEmVqu4PiO3TV:qmuyXc9-aB37Mt84:661IUTQkKY65ozI4:c6aU8nDA1zE4ON1-:SJWOt5WBMx
W1GZgm:03vCbawPnwVHUA7Q:dOXH_Q4F46MEGRFF:KUdr5WRYLXghcHdC:muD5TfbE5-fg3EnA:ZHMt2zk
SDMwGTvXx:mGNpSWTLDyMV-39w:AMR_DQgqXyUOf7Da:fS9sBGzex4OEp73p:ZkilgzONHDfiN1Xy:o-CN
2FoW-11vDdjO:GNjtYlgIfxckWf06:eSuT8BgAkCrgOHlp:e1GK7vJT-U8v8S7b:AFvE0OUv6GvANjjx:1
```

Now you can start the testing as you do for normal web application. You can try various attacks like Parameter Manipulation, SQL Injection, etc.

## Memory Analysis

If you install file on memory card it is possible that all the supporting files are also get installed on memory card and can get sensitive information like passwords, PIN, etc from locally stored files. In emulator this can be achieved by using MKSDCARD command. This command creates virtual SD card on the emulator

1. create virtual SD card on emulator  
   mksdcard [-l -label] <size>[k|M] <file>
2. Now start the emulator by specifying the location of SD card file.  
   emulator -sdcard <file>

Tools like SQLITE3 can also be used to grab information of the database.

**References**

http://samsunggalaxys2review.org/root-your-samsung-galaxy-s2-heres-how/

http://www.mcafee.com/us/resources/white-papers/foundstone/wp-pen-testing-android-apps.pdf

http://securityxploded.com/android_reversing.php