Mobile Security Framework

(MobSF) Configuration

https://github.com/MobSF/Mobile-Security-Framework-MobSF/wiki/1.-Documentation

Requirements

Static Analysis

- Python 2.7 - Python 2 Download (Latest Python 2.7 release is recommended)
- Oracle JDK 1.7 or above - Java JDK Download
- Mac OSX Users must install Command-line tools for
- MAC OS X - How to Install Commandline Tools in Mac
- iOS IPA Analysis works only on OSX and requires a MAC
- Windows App Static analysis requires a Windows Host or Windows VM for Mac
- and Linux. For Windows App Static Analysis, Read Windows App Static Analysis

NOTE:

- On Linux and Mac, install Oracle Java 1.7 or above and **make it the default** one.
- On Linux, make sure you have 32 bit execution support enabled.

Dynamic Analysis

- MobSF x86 Android VM requires Oracle VirtualBox - VirtualBox Download
- Android Studio and a configured virtual device is required if your using
- MobSF ARM Emulator. Intel HAXM is recommended.
- Hardware Requirements: Min 4GB RAM, 5GB HDD/SSD and Virtualization
- Support for running MobSF VM

Downloads

- Download MobSF Android x86 4.4.2 VM (v0.3) ova file: https://goo.gl/QxgHZa
- Download MobSF Android arm Emulator 4.1.2 (v1.0) file https://goo.gl/LRrGs3

Installation

- **Windows**: Clone MobSF Repository to C:\
- **Mac**: Clone MobSF Repository to /Users/[username]/
- **Linux**: Clone MobSF Repository to /home/[username]/

Configuring Static Analyzer

```
git clone https://github.com/MobSF/Mobile-Security-Framework-MobSF.git
cd Mobile-Security-Framework-MobSF
```

Install MobSF Python dependencies using **pip**

Windows

```
C:\Python27\python.exe -m pip install -r requirements.txt
```

NOTE: If you face any issues, download and install the latest python 2.7.x

Mac

```
pip install -r requirements.txt --user
```

```
If it throws error like pip command not found then run the following command.
```

```
sudo easy_install pip
```

```
Then run the command,
```

```
pip install -r requirements.txt --user
```

Linux

```
sudo apt install build-essential libssl-dev libffi-dev python-dev
pip install -r requirements.txt --user
```

PDF Report Generation

- You need to install `wkhtmltopdf` binary separately for generating PDF reports.
- Check [wkhtmltopdf downloads]() and [Installing wkhtmltopdf wiki]() for more information.
- In Windows, you need to add the folder that contains `wkhtmltopdf` binary to environment variable PATH.

Running MobSF

```
python manage.py runserver
```

If you need to run on a specific port number try

`python manage.py runserver PORT_NO`.

To expose MobSF to a particular IP, you can try

`python manage.py runserver IP:PORT_NO`.

If everything goes right, you will get an output like the one below.

```
Mobile Security Framework v0.9.3.8 Beta

OS: Darwin
Platform: Darwin-16.3.0-x86_64-i386-64bit

[INFO] Finding JDK Location in Linux/MAC....

[INFO] JDK 1.7 or above is available

[INFO] Checking for Update.

[INFO] No updates available.
System check identified no issues (0 silenced).
January 16, 2017 - 19:16:24
Django version 1.10b1, using settings 'MobSF.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

You can navigate to `http://localhost:8000/` to access the MobSF Web interface.

Configuring Dynamic Analyzer

MobSF Dynamic Analysis currently supports Android and can be done in four ways.

1. Dynamic Analysis with MobSF Android 4.4.2 x86 VirtualBox VM - **default**
2. (Fast, not all Apps work)
3. Dynamic Analysis with MobSF Android 4.1.2 arm Emulator - (Slow, Most Apps work)
4. Dynamic Analysis using a Rooted Android 4.03 - 4.4 Device (Very Fast, All Apps work)
5. Dynamic Analysis using a Rooted Android 4.03 - 4.4 VM (not tested)

Configuring Dynamic Analyzer with MobSF Android 4.4.2 x86 VirtualBox VM

Dynamic Anlayzer is available only for Android binaries (APK)

and works only if your computer has at least 4GB of RAM and Full Virtualization support.

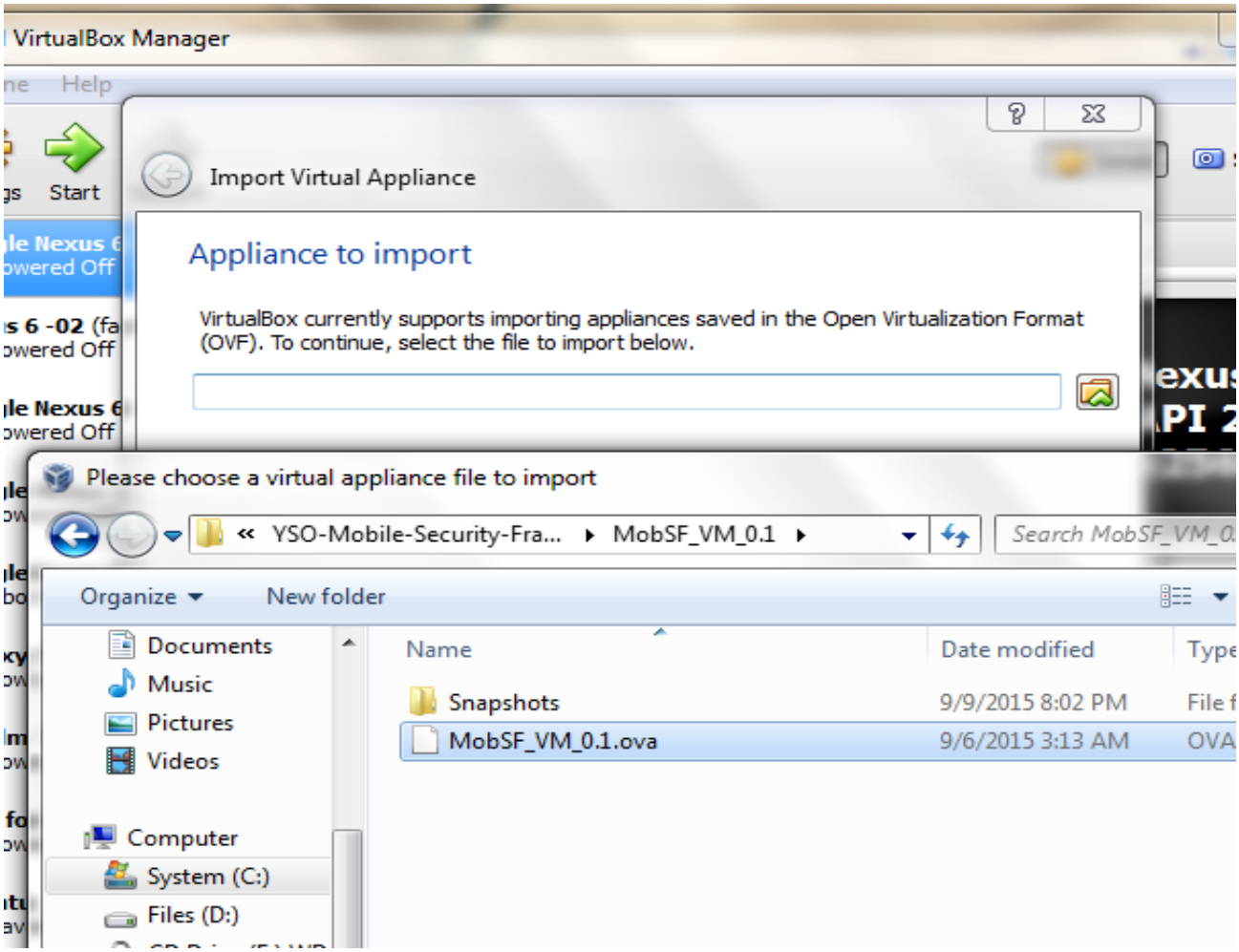
To Configure Dynamic Analyzer we need 4 things.

- VM UUID
- Snapshot UUID
- Host/Proxy IP
- VM/Device IP

Steps to Follow

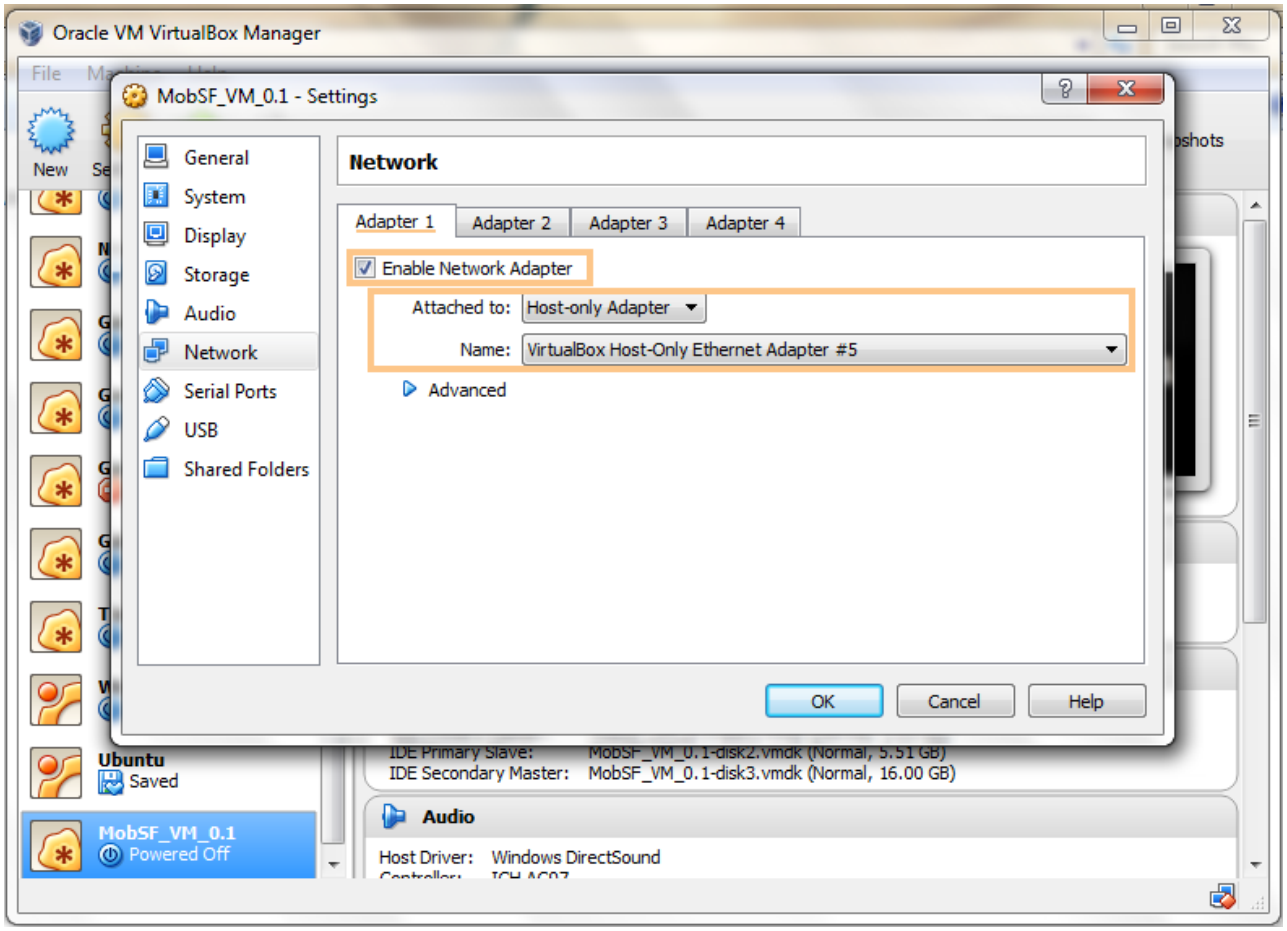
- Open VirtualBox,

  Go to **File -> Import Appliance** and select the MobSF_VM_X.X.ova file.

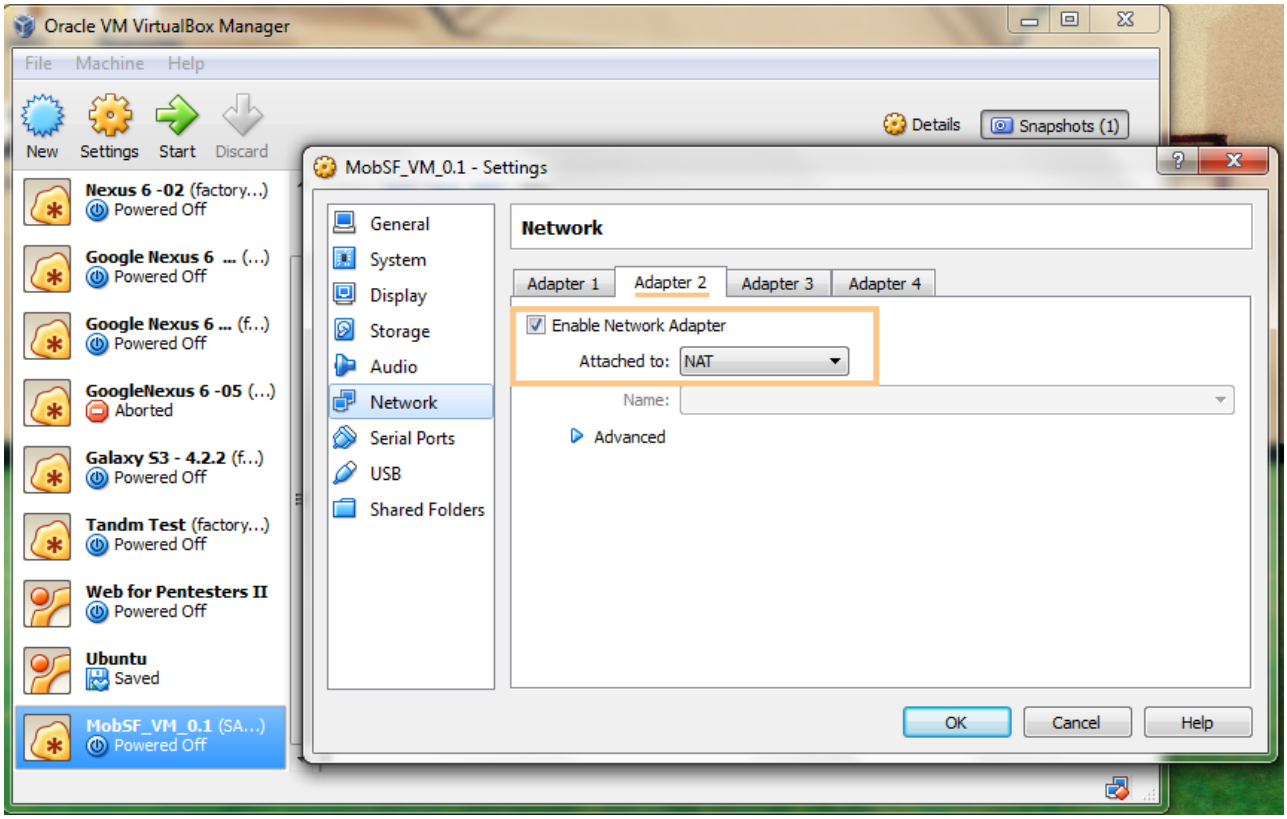


- Proceed with the import process. Do not alter anything.
- Once the OVA is Imported Successfully, you will see a new entry in VirtualBox named MobSF_VM_X.X
- Right Click MobSF VM and Choose Settings, Go to Network tab. Here we need to configure two Network Adapters.

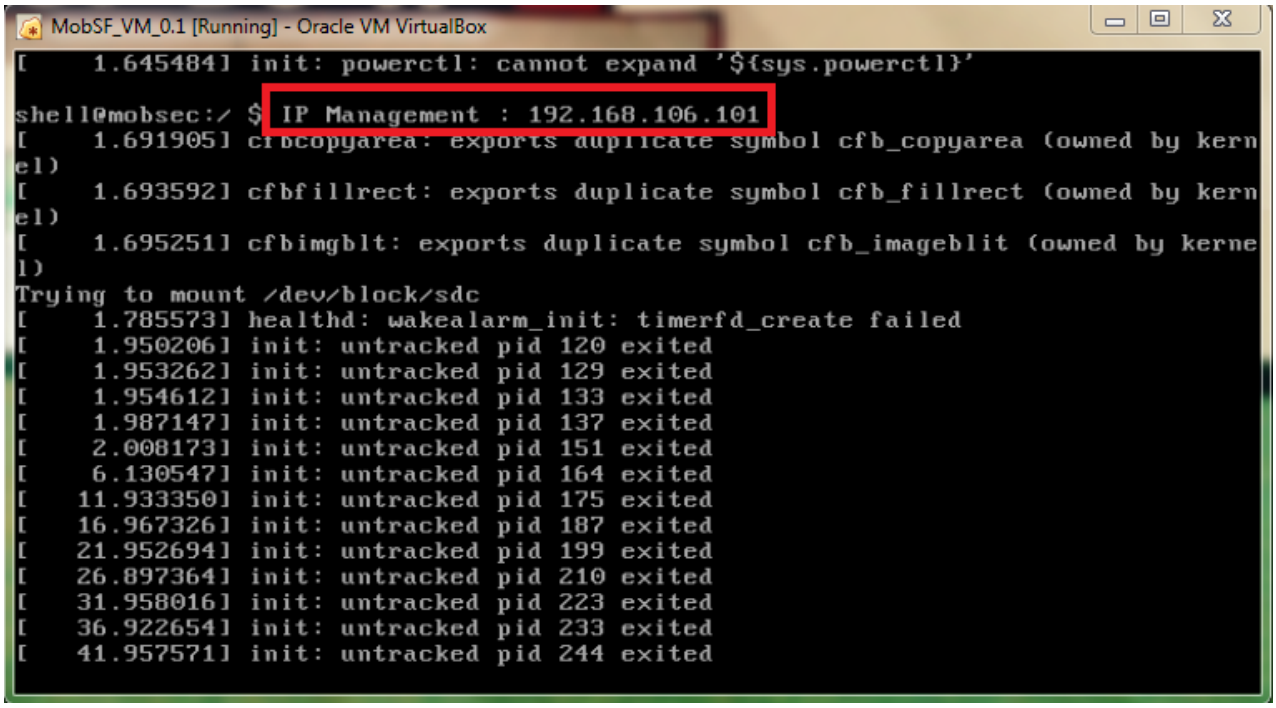
- **Adapter** 1 should be enabled and attached to **Host-only Adapter**. Remember the name of the adapter. We need the name to Identify the Host/Proxy IP.



- **Adapter 2** should be enabled and attached to **NAT**

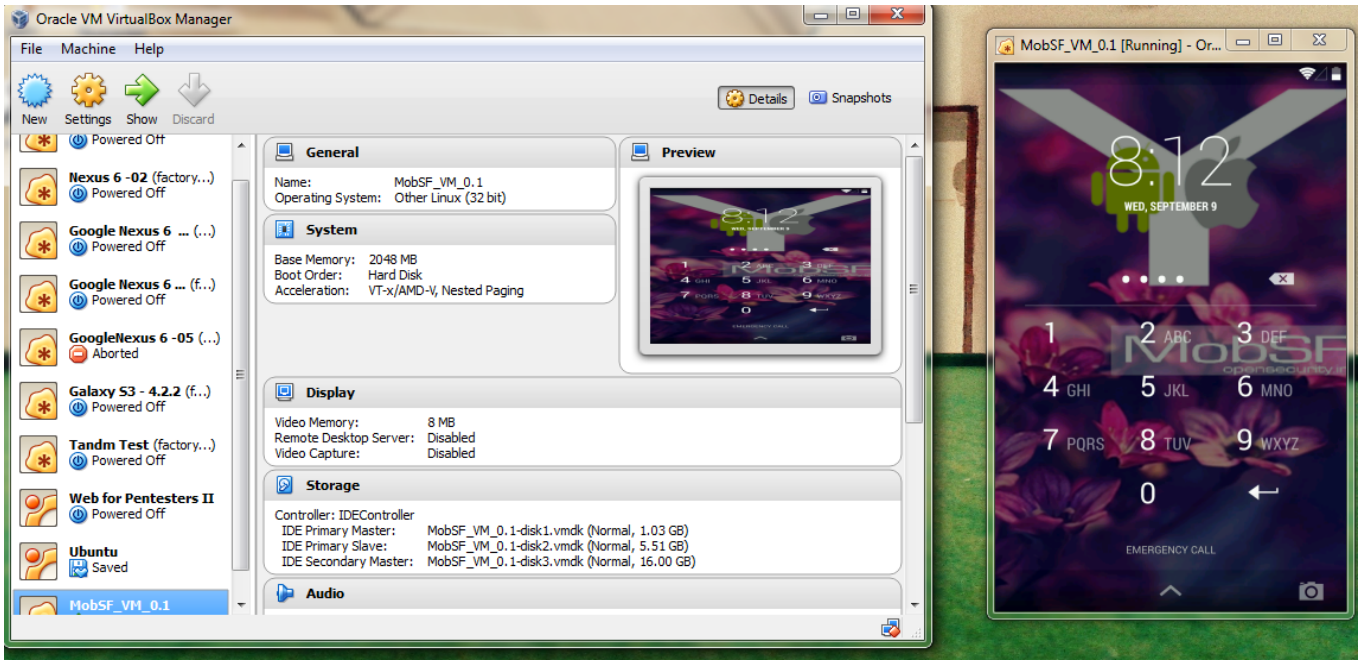


- Save the settings and Start MobSF VM. While the VM is Booting up. Note down the **VM IP**.



- Once the VM Boots up, It will present a Lock Screen.

  The password for the Lock Screen is `1234`



NOTE: If the VM does not boot up properly

then you cannot perform Dynamic Analysis with MobSF VM.

- **Getting the Host/Proxy IP**
  - **Windows** : Issue the command `ipconfig` in command prompt and note down the IP corresponding to the name of the Host-only Adapter.

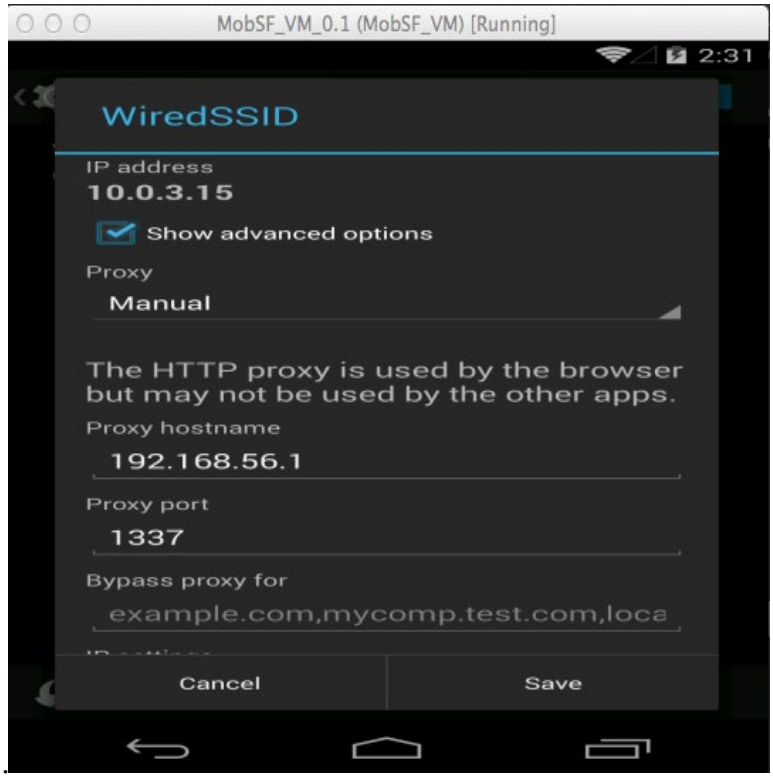
```
Ethernet adapter VirtualBox Host-Only Network #5:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::ad53:d578:d01c:ac1e%46
   IPv4 Address. . . . . . . . . . . : 192.168.106.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

- **Unix** : Issue the command `ifconfig` in terminal

  and note down the IP corresponding to the name of the Host-only Adapter.

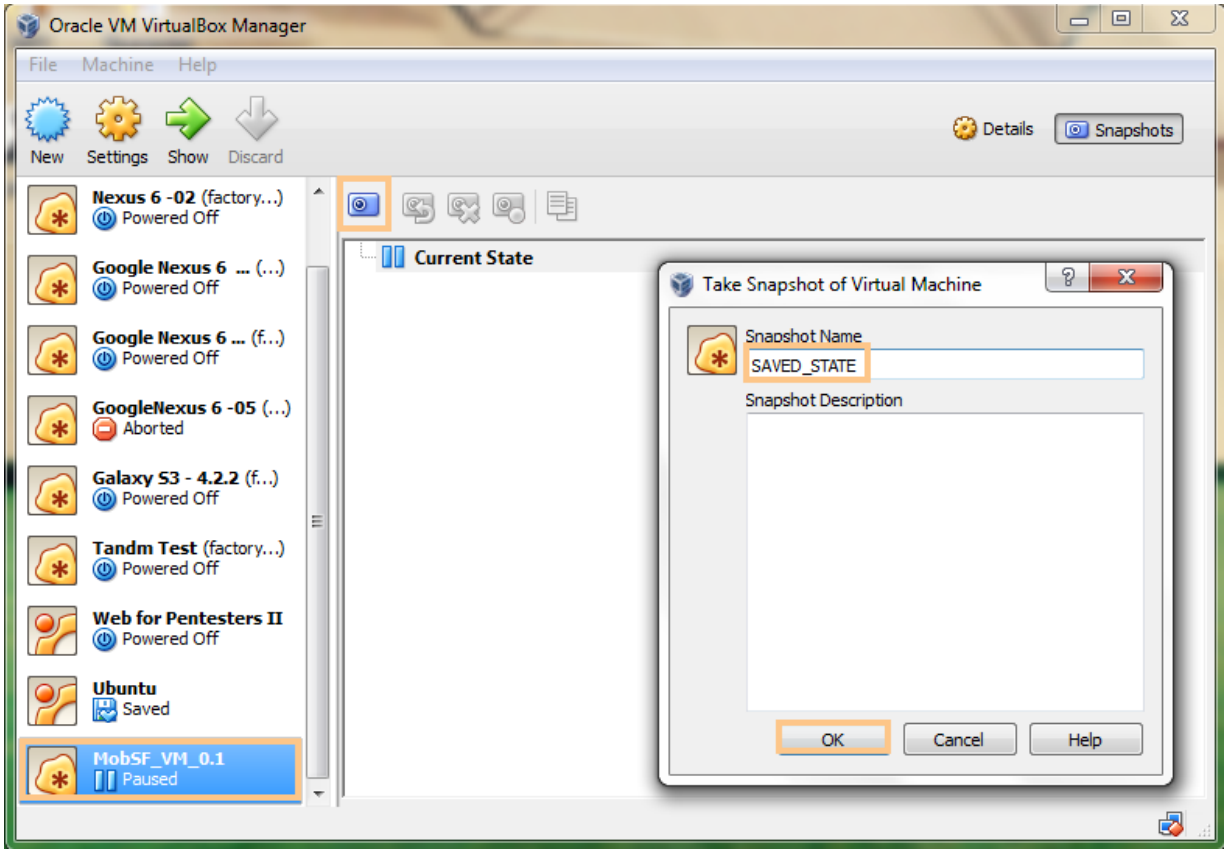
```
vboxnet0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 0a:00:27:00:00:00
        inet 192.168.56.1 netmask 0xffffff00 broadcast 192.168.56.255
```

- **NOTE**: The VirtualBox Host-Only Adapter IP
- and MobSF VM IP should be in the same network range.
- If your MobSF VM IP and Adapter IP are in different network range,
- modify the Adapter IP to be in the same network range as that of MobSF VM IP.
- See:
- [What to do when MobSF VM and VirtualBox Host Only Adapter are not in
- the same network range]
- Go to Wi-Fi Settings in **MobSF VM** and set the Proxy IP as the Host/Proxy IP which you have obtained from the previous step and port no as `1337`



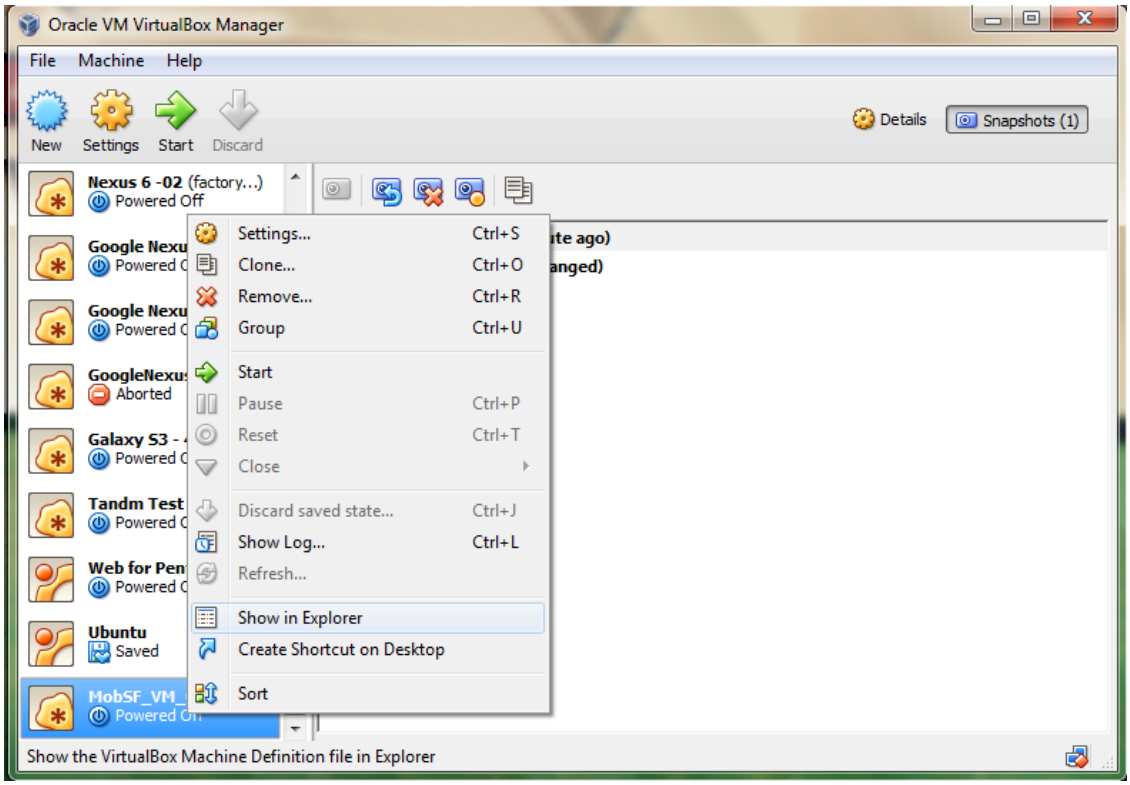
- Save the settings and Navigate to the Home Screen of **MobSF VM**.

  Wait for 30 seconds and save a snapshot of the **MobSF VM** in VirtualBox

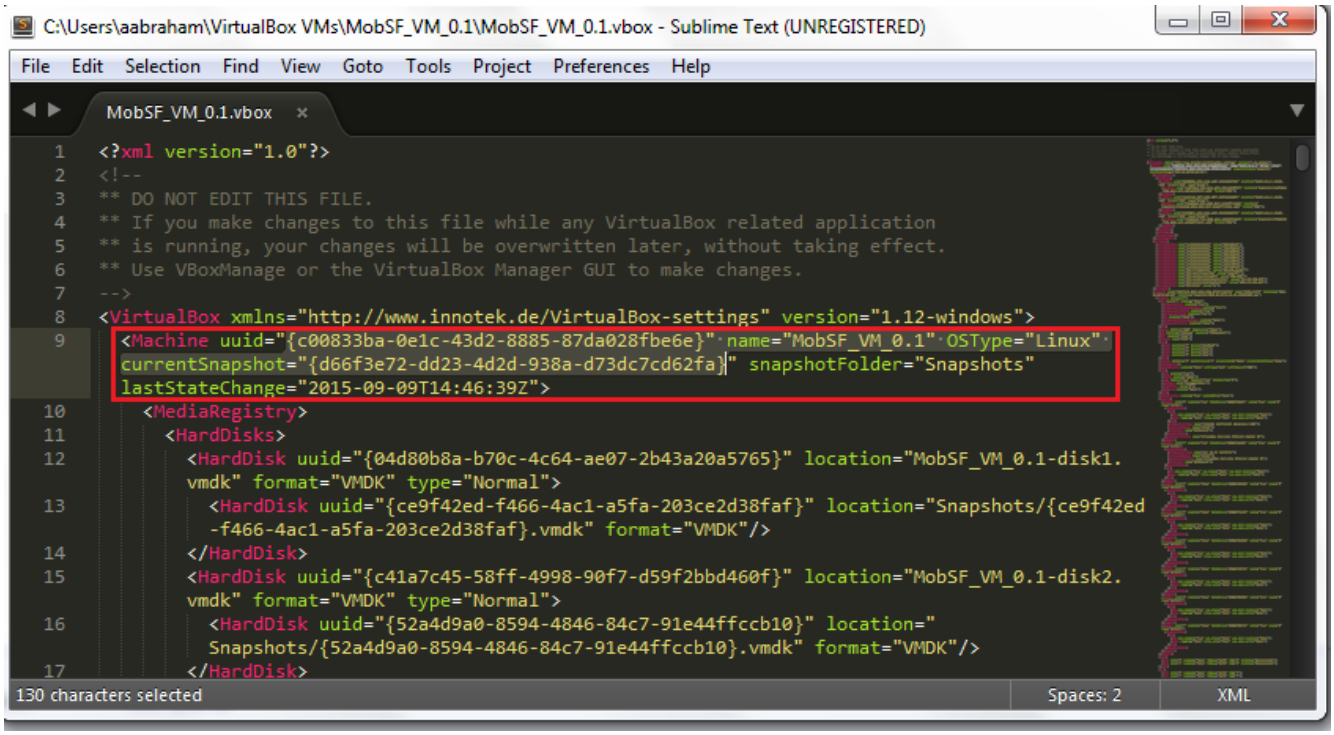


- Once the Snapshot is saved, right click **MobSF VM**

  and select `Show in Explorer` or `Show in Finder`.



- Open the File **MobSF_VM_X.X.vbox** in any Text Editor
- and note down the VM UUID and Snapshot UUID



Here the value of `uuid` is the **VM UUID** and `currentSnapshot` is the **Snapshot UUID**.

- Now we have all the things needed to configure

  the Dynamic Analyzer (Host/Proxy IP, VM IP, VM UUID and Snapshot UUID)

- Go to `MobSF/settings.py` and set the appropriate values as
  - UUID = VM UUID
  - SUUID = Snapshot UUID
  - VM_IP = VM IP
  - PROXY_IP = Host/Proxy IP
- In `MobSF/settings.py`, set `ANDROID_DYNAMIC_ANALYZER = "MobSF_VM"` (default)
- This will configure MobSF to use Android VirtualBox VM for Dynamic Analysis.

Configuring Dynamic Analyzer with with

MobSF Android 4.1.2 arm Emulator

- Make sure Android Studio is installed and an AVD is created.
- (Nexus 5 with Lollipop image is recommended)
- Extract MobSF_ARM_Emulator.zip
- Run `mobsfy_AVD.py` script and specify the directory that contains
- the files extracted from `MobSF_ARM_Emulator.zip`.
- In `MobSF/settings.py`, set `ANDROID_DYNAMIC_ANALYZER = "MobSF_AVD"`
- This will configure MobSF to use Android arm Emulator for Dynamic Analysis.

Manual Configuration (not recommended)

- If `mobsfy_AVD.py` script is not running successfully, you need to set the values for `AVD_EMULATOR` and `AVD_PATH` in `MobSF/settings.py` manually.
- Follow the README inside the emulator zip and change all
- the path fields according to your system
- edit `MobSF/settings.py` and modify

```
AVD_EMULATOR = r'/Users/[USERNAME]/Library/Android/sdk/tools/emulator'
# This can be
/Users/[USERNAME]/Library/Android/Sdk/emulator/emulator for
 newer versions of android SDK
AVD_PATH = r'/Users/[USERNAME]/.android/avd' # Path to the and folder
 where you extracted the emulator
```

- In `MobSF/settings.py`, set `ANDROID_DYNAMIC_ANALYZER = "MobSF_AVD"`

Configuring Dynamic Analyzer with

Rooted Android 4.03 - 4.4 Device

- MobSFy the Rooted Android Device, Follow the instructions here: [Configure MobSF Dynamic Analysis Environment in Android Device]
- In `MobSF/settings.py`, set `ANDROID_DYNAMIC_ANALYZER = "MobSF_REAL_DEVICE"`
- Set `DEVICE_IP` and `DEVICE_ADB_PORT` with the IP and PORT that you got from WiFi ADB

Configuring Dynamic Analyzer with

Rooted Android 4.03 - 4.4 VM

- MobSFy the Custom VM, Follow the instructions here:
- [Configure MobSF Dynamic Analysis Environment in Custom VM]
- **VM on Virtual Box**: If the VM is hosted on VirtualBox,
- follow the same steps that you have followed for configuring
- [MobSF x86 VirtualBox VM] and set appropriate `VM UUID`, `Snapshot UUID`, `Host/Proxy IP`, `VM IP` and set `ANDROID_DYNAMIC_ANALYZER = "MobSF_VM"`
- **Any Other VM**: Configure it as a Real device. Set `ANDROID_DYNAMIC_ANALYZER = "MobSF_REAL_DEVICE"` and specify `DEVICE_IP`and `DEVICE_ADB_PORT`.
- Snapshot feature is only available with VM(s) hosted in VirtualBox.

Updating MobSF

If you are updating MobSF, In most cases you might have

to perform database migrations or you will see errors such as

```
[ERROR] Saving to DB
 (E:\Mobile-Security-Framework-MobSF\StaticAnalyzer\views\android
\db_interaction.py, LINE 236 "static_db.save()"):
table StaticAnalyzer_staticanalyzerandroid has no column named
```

Run the below command to migrate your db

```
python manage.py makemigrations
python manage.py migrate
```

If the above changes didn't work, you might need to delete the file `db.sqlite3`,

or run `clean.sh` in Mac/Linux. After that run the above commands.

NOTE: This will remove the previously saved MobSF scan results.

Running Tests

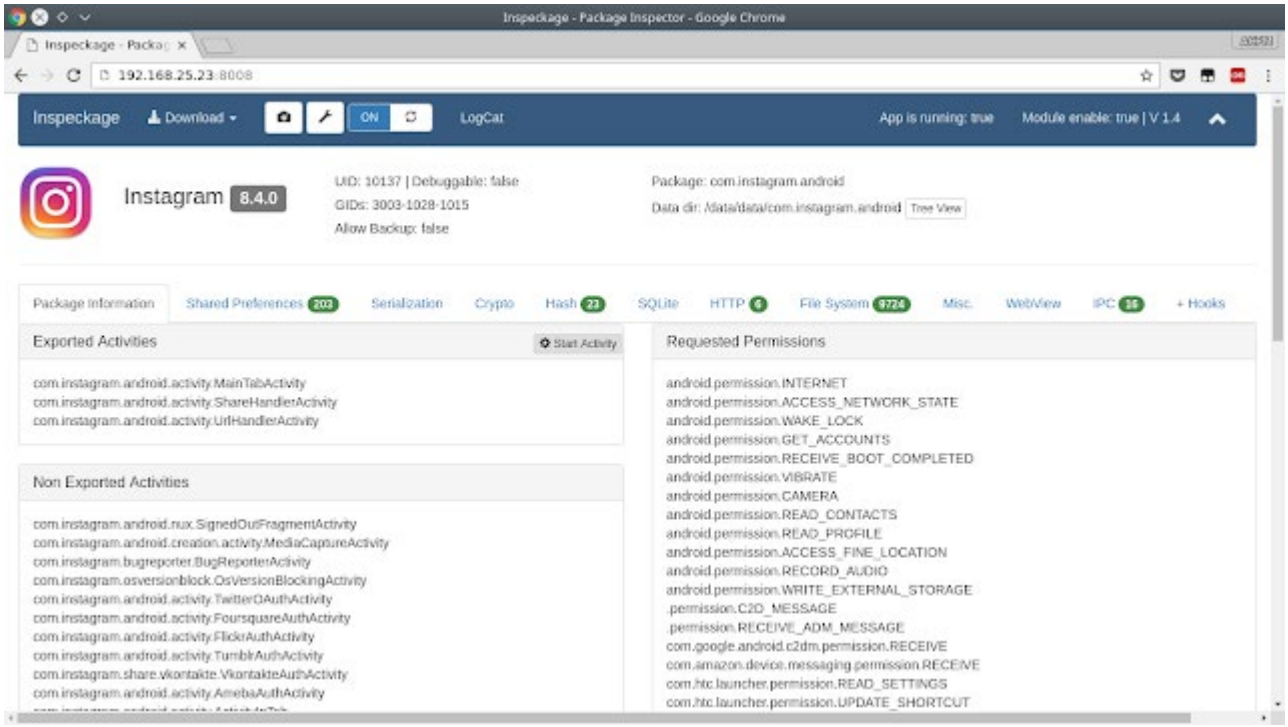
- Basic Static Analyzer unit tests - run MobSF and navigate to `http://127.0.0.1:8000/runtest/`
- MobSF REST API unit tests - run MobSF and navigate to `http://127.0.0.1:8000/runapitest/`

Dynamic analysis with Inspackage
https://www.slideshare.net/VodqaBLR/dynamic-security-analysis-static-security-analysis-for-android-apps/

nspeckage - (Android Package nspector) Dynamic Analysis With Api Hooks, Start Unexported Activities And More

http:

//www.kitploit.com/2017/04/inspeckage-android-package-inspector.html



Inspeckage is a tool developed to offer dynamic analysis of Android applications.

By applying hooks to functions of the Android API, Inspeckage will help you understand what an Android application is doing at runtime.

- http://ac-pm.github.io/Inspeckage
- https://twitter.com/inspeckage
- https://play.google.com/store/apps/details?id=mobi.acpm.inspeckage
- http://repo.xposed.info/module/mobi.acpm.inspeckage

Features

With Inspeckage, we can get a good amount of information about the application's behavior:

Information gathering

- Requested Permissions;
- App Permissions;
- Shared Libraries;
- Exported and Non-exported Activities, Content Providers,Broadcast Receivers and Services;
- Check if the app is debuggable or not;
- Version, UID and GIDs;
- etc.

Hooks (so far)

With the hooks, we can see what the application is doing in real time:

- Shared Preferences (log and file);
- Serialization;

- Crypto;
- Hashes;
- SQLite;
- HTTP (an HTTP proxy tool is still the best alternative);
- File System;
- Miscellaneous (Clipboard, URL.Parse());
- WebView;
- IPC;
- + Hooks (add new hooks dynamically)

Actions
With Xposed it's possible to perform actions such as start a unexported activity and much else:

- Start any activity (exported and unexported);
- Call any provider (exported and unexported);
- Disable FLAG_SECURE;
- SSL uncheck (bypass certificate pinning - JSSE, Apache and okhttp3);
- Start, stop and restart the application;
- Replace params and return value (+Hooks tab).

Extras

- APK Download;
- View the app's directory tree;
- Download the app's files;
- Download the output generated by hooks in text file format;
- Take a screen capture;
- Send text to android clipboard.

Configuration
Even though our tool has some hooks to the HTTP libraries, using an external proxy tool is still the

best option to analyze the app's traffic. With Inspeckage, you can:

- Add a proxy to the target app;
- Enable and disable proxy;
- Add entries in the arp table.

Logcat
Logcat.html page. A experimental page with websocket to show some information from the logcat.

Installation

Requirements: Xposed Framework

Xposed Installer

1. Go to Xposed Installer, select "Download"
2. Refresh and search for "Inspeckage"
3. Download the latest version and install
4. Enable it in Xposed
5. Reboot and enjoy!

Xposed Repository
Get it from Xposed repo: http://repo.xposed.info/module/mobi.acpm.inspeckage

```
adb install mobi.acpm.inspeckage.apk
```

1. Enable it in Xposed
2. Reboot and enjoy!

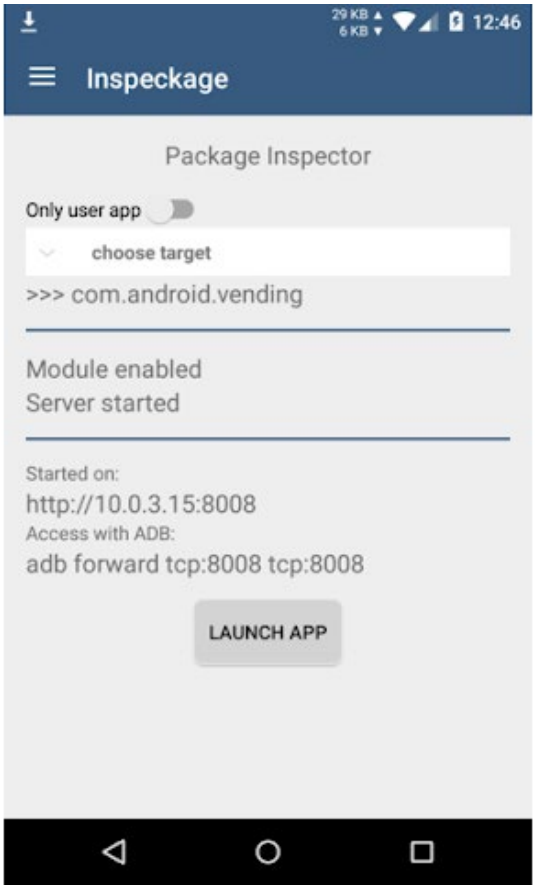
From Source
Feel free to download the source!

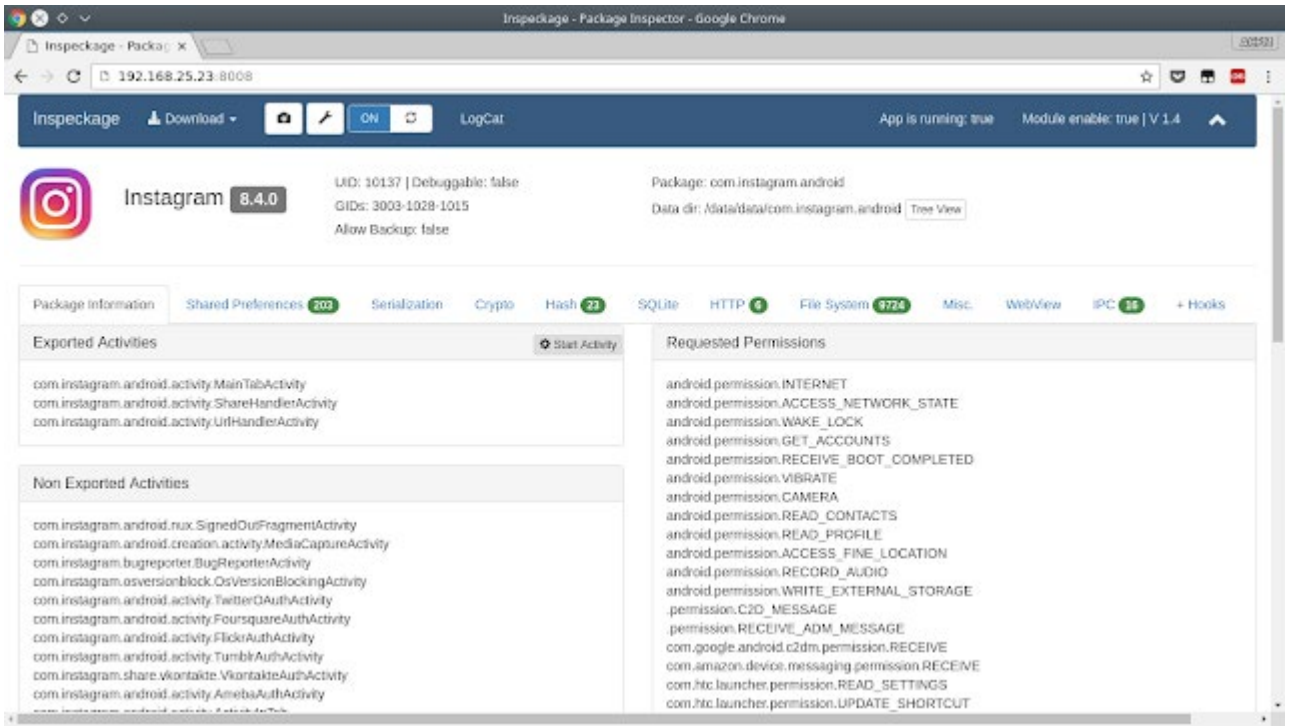
How to uninstall

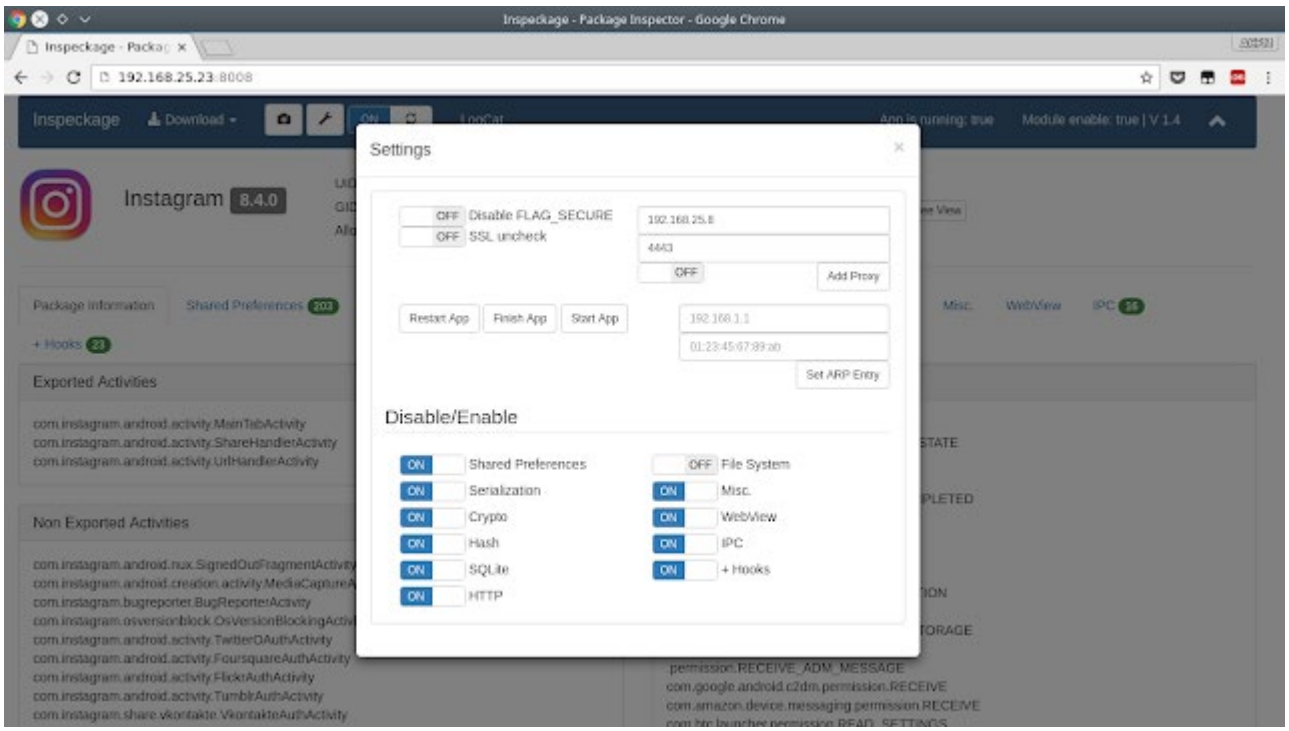
```
adb uninstall mobi.acpm.inspeckage
```

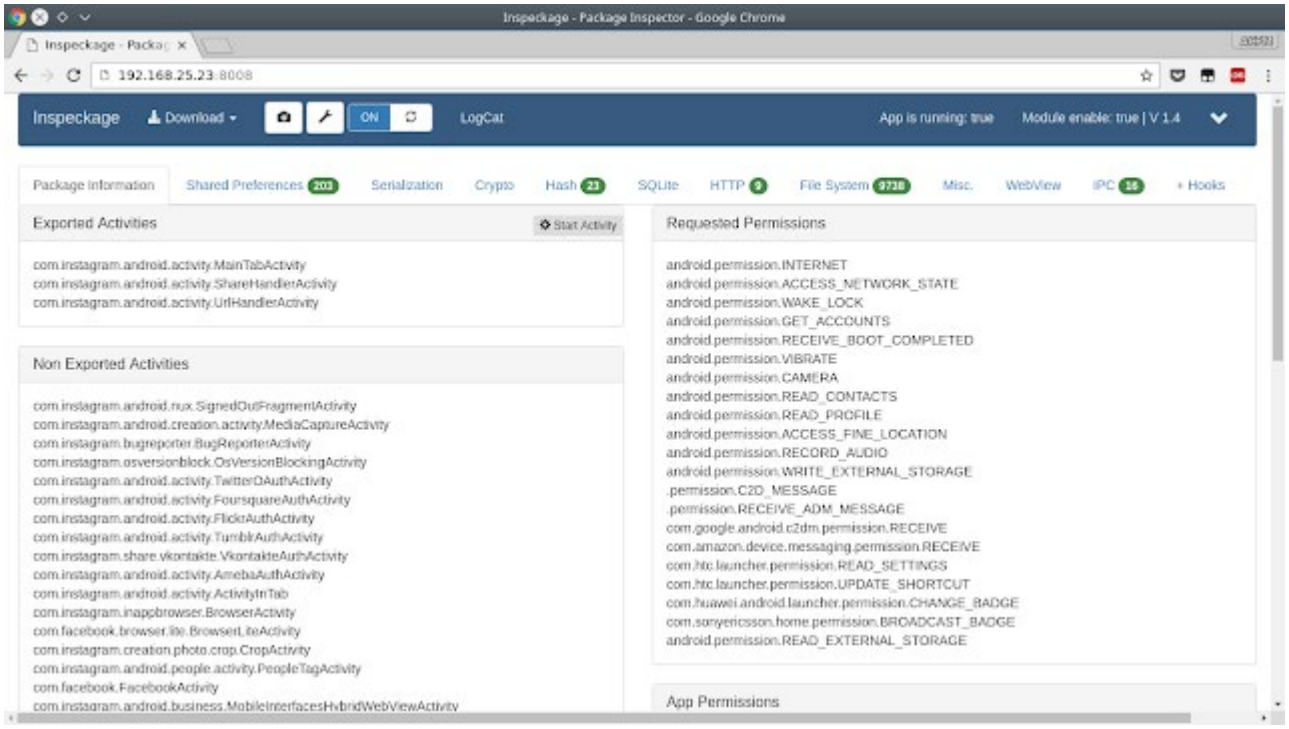
And reboot!

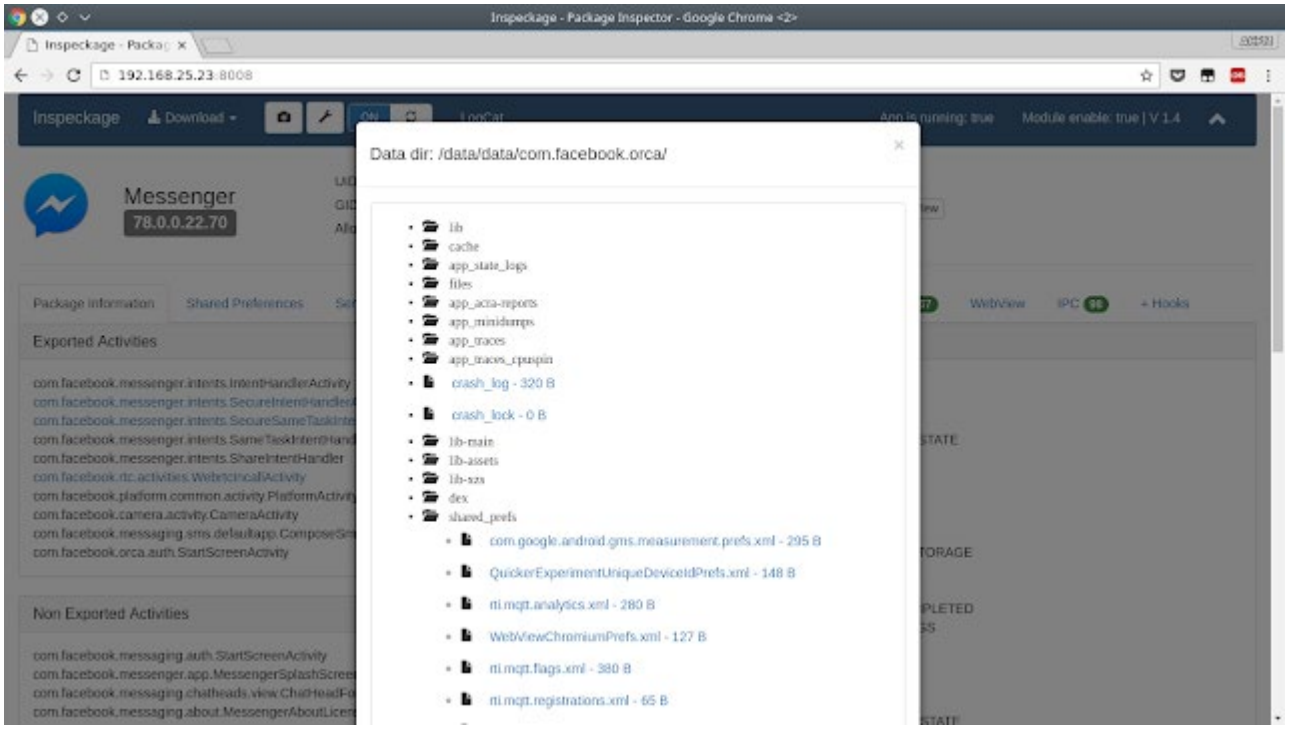
Genymotion Screenshots

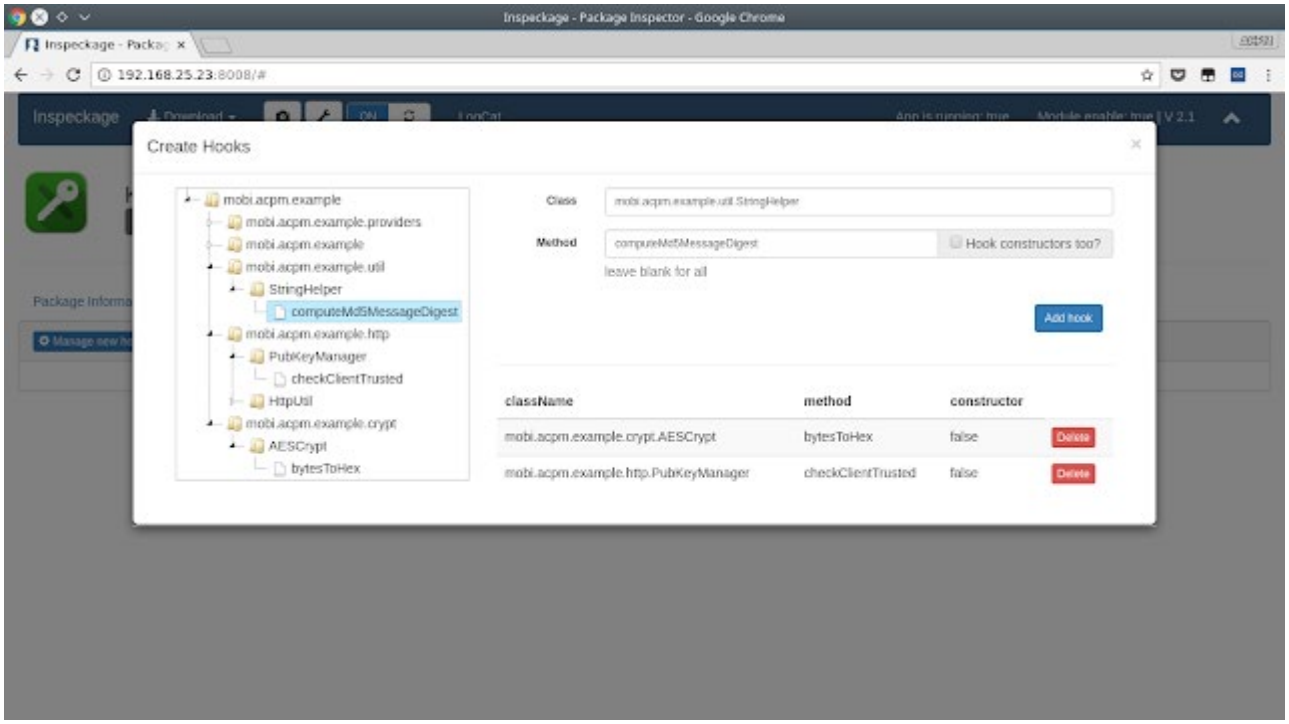



Inspeckage - Package Inspector - Google Chrome
192.168.25.23:8008
Inspeckage
Download
ON
LogCat
App is running: true
Module enable: true | V 1.4
Instagram 8.4.0
UID: 10137 | Debuggable: false
GIDs: 3003-1028-1015
Allow Backup: false
Package: com.instagram.android
Data dir: /data/data/com.instagram.android
Tree View
Package Information
Shared Preferences 203
Serialization
Crypto
Hash 23
SQLite
HTTP 6
File System 9724
Misc.
WebView
IPC 16
+ Hooks
Exported Activities
Start Activity
com.instagram.android.activity.MainTabActivity
com.instagram.android.activity.ShareHandlerActivity
com.instagram.android.activity.UrlHandlerActivity
Non Exported Activities
com.instagram.android.nux.SignedOutFragmentActivity
com.instagram.android.creation.activity.MediaCaptureActivity
com.instagram.bugreporter.BugReporterActivity
com.instagram.osversionblock.OsVersionBlockingActivity
com.instagram.android.activity.TwitterOAuthActivity
com.instagram.android.activity.FoursquareAuthActivity
com.instagram.android.activity.FlickrAuthActivity
com.instagram.android.activity.TumblrAuthActivity
com.instagram.share.vkontakte.VkontakteAuthActivity
com.instagram.android.activity.AmebaAuthActivity
Requested Permissions
android.permission.INTERNET
android.permission.ACCESS_NETWORK_STATE
android.permission.WAKE_LOCK
android.permission.GET_ACCOUNTS
android.permission.RECEIVE_BOOT_COMPLETED
android.permission.VIBRATE
android.permission.CAMERA
android.permission.READ_CONTACTS
android.permission.READ_PROFILE
android.permission.ACCESS_FINE_LOCATION
android.permission.RECORD_AUDIO
android.permission.WRITE_EXTERNAL_STORAGE
.permission.C2D_MESSAGE
.permission.RECEIVE_ADM_MESSAGE
com.google.android.c2dm.permission.RECEIVE
com.amazon.device.messaging.permission.RECEIVE
com.htc.launcher.permission.READ_SETTINGS
com.htc.launcher.permission.UPDATE_SHORTCUT


Inspeckage - Package Inspector - Google Chrome
192.168.25.23:8008
Settings
OFF Disable FLAG_SECURE
OFF SSL uncheck
192.168.25.8
4443
OFF
Add Proxy
Restart App
Finish App
Start App
192.168.1.1
01:23:45:67:89:ab
Set ARP Entry
Disable/Enable
ON Shared Preferences
ON Serialization
ON Crypto
ON Hash
ON SQLite
ON HTTP
OFF File System
ON Misc.
ON WebView
ON IPC
ON + Hooks


Inspeckage - Package Inspector - Google Chrome
192.168.25.23:8008
Inspeckage
Download
ON
LogCat
App is running: true
Module enable: true | V 1.4
Package Information
Shared Preferences 203
Serialization
Crypto
Hash 23
SQLite
HTTP 9
File System 9738
Misc.
WebView
IPC 16
+ Hooks
Exported Activities
Start Activity
com.instagram.android.activity.MainTabActivity
com.instagram.android.activity.ShareHandlerActivity
com.instagram.android.activity.UrlHandlerActivity
Non Exported Activities
com.instagram.android.nux.SignedOutFragmentActivity
com.instagram.android.creation.activity.MediaCaptureActivity
com.instagram.bugreporter.BugReporterActivity
com.instagram.osversionblock.OsVersionBlockingActivity
com.instagram.android.activity.TwitterOAuthActivity
com.instagram.android.activity.FoursquareAuthActivity
com.instagram.android.activity.FlickrAuthActivity
com.instagram.android.activity.TumblrAuthActivity
com.instagram.share.vkontakte.VkontakteAuthActivity
com.instagram.android.activity.AmebaAuthActivity
com.instagram.android.activity.ActivityInTab
com.instagram.inapbrowser.BrowserActivity
com.facebook.browser.lite.BrowserLiteActivity
com.instagram.creation.photo.crop.CropActivity
com.instagram.android.people.activity.PeopleTagActivity
com.facebook.FacebookActivity
com.instagram.android.business.MobileInterfacesHybridWebViewActivity
Requested Permissions
android.permission.INTERNET
android.permission.ACCESS_NETWORK_STATE
android.permission.WAKE_LOCK
android.permission.GET_ACCOUNTS
android.permission.RECEIVE_BOOT_COMPLETED
android.permission.VIBRATE
android.permission.CAMERA
android.permission.READ_CONTACTS
android.permission.READ_PROFILE
android.permission.ACCESS_FINE_LOCATION
android.permission.RECORD_AUDIO
android.permission.WRITE_EXTERNAL_STORAGE
.permission.C2D_MESSAGE
.permission.RECEIVE_ADM_MESSAGE
com.google.android.c2dm.permission.RECEIVE
com.amazon.device.messaging.permission.RECEIVE
com.htc.launcher.permission.READ_SETTINGS
com.htc.launcher.permission.UPDATE_SHORTCUT
com.huawei.android.launcher.permission.CHANGE_BADGE
com.sonyericsson.home.permission.BROADCAST_BADGE
android.permission.READ_EXTERNAL_STORAGE
App Permissions


Inspeckage - Package Inspector - Google Chrome <2>
192.168.25.23:8008
Inspeckage
Download
Module enable: true | V 1.4
Messenger
78.0.0.22.70
Data dir: /data/data/com.facebook.orca/
lib
cache
app_state_logs
files
app_acra-reports
app_minidumps
app_traces
app_traces_cpuspin
crash_log - 320 B
crash_lock - 0 B
lib-main
lib-assets
lib-xzs
dex
shared_prefs
com.google.android.gms.measurement.prefs.xml - 295 B
QuickerExperimentUniqueDeviceIdPrefs.xml - 148 B
ni.mqtt.analytics.xml - 280 B
WebViewChromiumPrefs.xml - 127 B
ni.mqtt.flags.xml - 380 B
ni.mqtt.registrations.xml - 65 B
Exported Activities
com.facebook.messenger.intents.IntentHandlerActivity
com.facebook.messenger.intents.SecureIntentHandler
com.facebook.messenger.intents.SecureSameTaskIntent
com.facebook.messenger.intents.SameTaskIntentHand
com.facebook.messenger.intents.ShareIntentHandler
com.facebook.rtc.activities.WebrtcIncallActivity
com.facebook.platform.common.activity.PlatformActivity
com.facebook.camera.activity.CameraActivity
com.facebook.messaging.sms.defaultapp.ComposeSm
com.facebook.orca.auth.StartScreenActivity
Non Exported Activities
com.facebook.messaging.auth.StartScreenActivity
com.facebook.messenger.app.MessengerSplashScree
com.facebook.messaging.chatheads.view.ChatHeadFo
com.facebook.messaging.about.MessengerAboutLicen


Inspeckage - Package Inspector - Google Chrome
192.168.25.23:8008/#
Inspeckage
Create Hooks
mobi.acpm.example
mobi.acpm.example.providers
mobi.acpm.example
mobi.acpm.example.util
StringHelper
computeMd5MessageDigest
mobi.acpm.example.http
PubKeyManager
checkClientTrusted
HttpUtil
mobi.acpm.example.crypt
AESCrypt
bytesToHex
Class
mobi.acpm.example.util.StringHelper
Method
computeMd5MessageDigest
Hook constructors too?
leave blank for all
Add hook
className
method
constructor
mobi.acpm.example.crypt.AESCrypt
bytesToHex
false
Delete
mobi.acpm.example.http.PubKeyManager
checkClientTrusted
false
Delete