

# A very insecure Android CTF App

## Lab Setup Guide

### Beetlebug



Beetlebug v1.0

Author & Developer: Hafiz Aziz

Email: [contact@hafiz.ng](mailto:contact@hafiz.ng)

March, 2022

## Tools Required

To get started with Beetle CTF challenges, you will need to set up your Lab environment. These tools are requirements for completing the CTF challenges in the application.

- Genymotion
- ADB (Android Debug Bridge)
- MobSF (optional)
- JADX – Java Decompiler
- Drozer (optional)
- apktool

# Android Debug Bridge

ADB is command line utility that facilitates communication between your Android device and a PC. This tool allows you to send a wide array of terminal commands—including but not limited to [basic Linux shell commands](#) to your phone at just about any time (as long as you have debugging enabled on your phone).

## How to set up ADB

1. To use ADB on your system, you first need to download and unzip the Android [SDK Platform Tools](#) from the Android developer website.

Mac users can use Homebrew to install ADB using the command:

```
brew install/homebrew/cask/android-platform-tools
```

2. On your Android device, you will need to enable USB debugging in the developer settings before you can connect the device over ADB.
3. Go to **Setting > About phone** then tap on the Build number several times and wait to the message "You are now a developer!". Once again, go to **Settings > Developer options**, then toggle on **USB debugging**.
4. Now connect the Android device to your PC using USB cable. Always ensure to allow the USB debugging prompt that will appear on your Android device. Open the command line or Terminal and navigate to the **Platform Tools** folder.

<https://github.com/hafiz-ng/beetlebug>

By default, you'll either have to navigate to the folder where you extracted the Platform Tools package and run any ADB command from there, or write out the full path where ADB is every single time you want to run a command. For example, this is a simple command to see what devices are attached to your system:

```
adb devices
```

If your command prompt isn't open to the location where you extracted the Platform Tools, you would have to type something like this:

```
c:\User\Downloads\platform-tools\adb.exe devices
```

That's a pain to go through every single time you want to tweak something on your phone. To fix this, we can modify something called the PATH variable so that you can run ADB no matter which folder you're in.

## Edit your PATH Variable

PATH is used by Windows to specify the location of important executables.

1. Open Windows Explorer and right click "My PC". Select "Properties" and you will be greeted with a screen showing some system information..
2. Select "Advanced System Settings".
3. Select "Environment Variables".
4. Look for the variable named "Path" and double click it.
5. Click "Browse" and navigate to the folder where you extracted your adb files. Next "okay" out of all of the Windows you have open. Start a new PowerShell or command prompt and type "adb" to verify the location has been added. If not, reboot your PC and try again.

6. Now type `adb devices` in the command prompt and hit Enter. If everything works the way it should, you will see your device's serial number under the list of attached devices.

```
% adb devices
List of devices attached
RF8M31YY46J device
```

## How to use Android ADB wirelessly (optional)

Once you have set up the ADB and connected the Android device to your PC, follow the steps below to establish a wireless connection to your PC via ADB.

1. Type `adb tcpip 5555` in the command line or Terminal and press Enter.

```
(base) hafiz@Hafizs-MacBook-Pro Beetlebug % adb tcpip 5555
restarting in TCP mode port: 5555
(base) hafiz@Hafizs-MacBook-Pro Beetlebug % █
```

2. Find your phone's IP address in **Settings > About Phone > Status > IP Address**.
3. Back in the command line or Terminal, type `adb connect [your Android's IP address]`
4. Press Enter again, the Android device should now be connected

# Genymotion

If you do not have a physical rooted Android device, Genymotion is a great option to get you started. Genymotion Desktop is an Android emulator you can use to test your Android applications on a wide range of virtual devices for development, test and demonstration purposes.

## Setting Up Genymotion

1. In order to use Genymotion Desktop, you will need to create and activate a Genymotion account. Create your account at <https://genymotion.com/account/create>. When prompted for license type, select "Personal License" and make sure to click on the verification email sent to the email address to activate your account.
2. Ensure you have the latest version of VirtualBox installed on your system. Visit <https://www.virtualbox.org/wiki/Downloads> to download the latest binary of Virtualbox VM.
3. Download and install Genymotion Desktop from <https://www.genymotion.com/download/>, select your OS
4. Launch Genymotion, Sign In and create a new virtual device.
5. To install Android applications from the Play Store, you will need to install some additional tools.

# MobSF

MobSF framework is an awesome tool for the security analysis of mobile applications. This tool supports both static and dynamic analysis. This section covers MobSF installation on Linux-based distributions (e.g. Ubuntu) and Windows systems.

## MobSF installation on Linux

**Step 1: Download MobSF installer on system**

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

If git not installed on machine,

```
sudo apt-get install git-all
```

**Step 2: Change directory by using the cd command**

```
cd Mobile-Security-Framework-MobSF
```

**Step 3: Run**

```
./setup.sh
```

<https://github.com/hafiz-ng/beetlebug>

After successful installation. Use the below command to run MobSF.

```
./run.sh 127.0.0.1:8000
```

Now you can access MobSF by pointing your browser to <http://127.0.0.1:8080>

## MobSF installation on Windows

**Step 1:** Download by using the git command or else you can download by browsing URL.

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

**Step 2:** Change directory to the MobSF directory

```
cd Mobile-Security-Framework-MobSF
```

**Step 3:** Install and Run MobSF

```
setup.bat
```

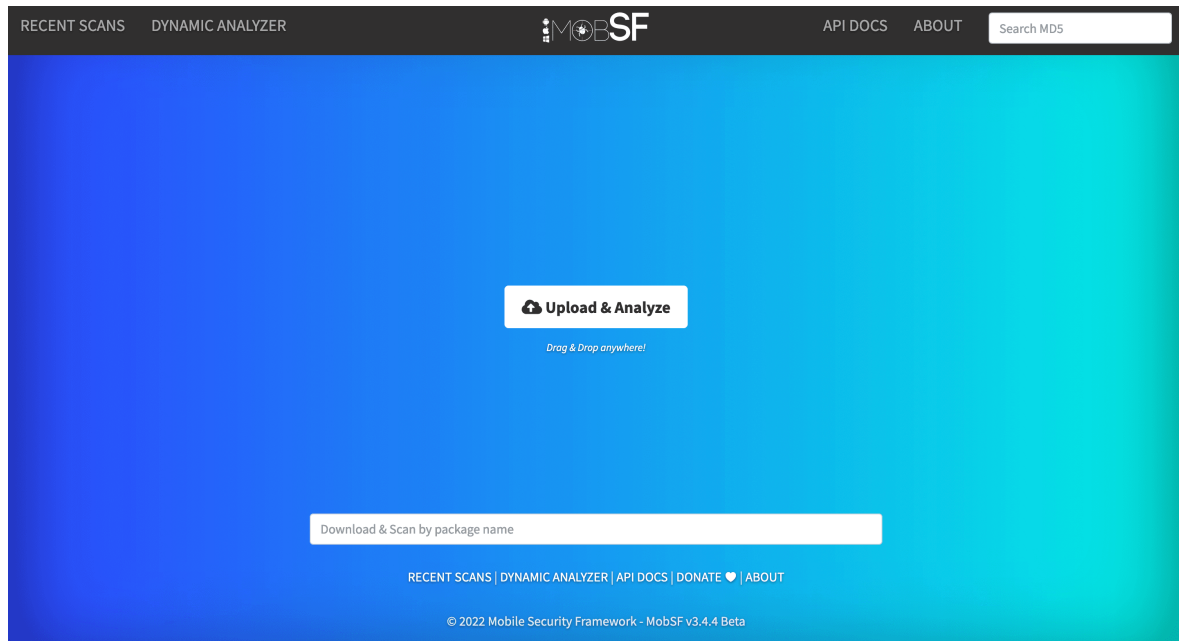
**Step 4:** Run MobSF

```
run.bat 127.0.0.1:8000
```

<https://github.com/hafiz-ng/beetlebug>



Now you can access MobSF using this URL <http://localhost:8000/>



## MobSF User Interface

<https://github.com/hafiz-ng/beetlebug>

# JADX






JADX is a Command line and GUI tools for producing Java source code from Android Dex and APK files.

## Install on Windows

This application is public and licensed under the Apache 2.0 License. The source code is available on Github and the binaries are available to download in the [releases window here](#). Download always the latest version.

### 1. Download the JADX binaries






#### ▼ Assets 5

 <a href="#">jadx-1.3.4.zip</a>	29.2 MB
 <a href="#">jadx-gui-1.3.4-no-jre-win.exe</a>	31.9 MB
 <a href="#">jadx-gui-1.3.4-with-jre-win.zip</a>	55.5 MB
 <a href="#">Source code</a> (zip)	
 <a href="#">Source code</a> (tar.gz)	

  25 25 people reacted

### 2. Extract zip content

Extract the content of the downloaded zip files into a folder of your preference, the content of the zip will be something like:

Name	Date modified	Type	Size
 bin	05.12.2016 08:18	File folder	
 lib	05.12.2016 08:18	File folder	
 LICENSE	05.12.2016 08:18	File	12 KB
 NOTICE	05.12.2016 08:18	File	11 KB
 README.md	05.12.2016 08:18	Markdown Source...	4 KB

<https://github.com/hafiz-ng/beetlebug>

### 3. Start Jadx-GUI

After download unpack zip file go to bin directory and run:

jadx-gui - UI version

jadx - command line version

## Install on macOS

Use the command below to install JADX using the Brew Package Manager on macOS.

```
brew install jadx
```

# Drozer

Drozer is a security testing framework for Android. It allows you to search for security vulnerabilities in apps and devices by assuming the role of an app and interacting with the Dalvik VM, other apps' IPC endpoints and the underlying OS.

## Prerequisites

The following prerequisites must be considered to set up Drozer:

1. **Python 2.7** – On Windows please ensure that the path to the Python installation and the Scripts folder under the Python installation are added to the PATH environment variable.
2. Java Development Kit 1.7
3. [Android Debug Bridge](#)

## Installation on Windows

1. Install Drozer Client on your host. Download it from the installer from the latest releases <https://github.com/FSecureLABS/drozer/releases>.

### 2.3.4

Updated missing tag & release as per #177

▼ Assets 8

 <a href="#">drozer-2.3.4-1.noarch.rpm</a>	21.6 MB
 <a href="#">drozer-2.3.4.tar.gz</a>	21.8 MB
 <a href="#">drozer-agent-2.3.4.apk</a>	618 KB
 <a href="#">drozer-installer-2.3.4.zip</a>	38 MB
 <a href="#">drozer_2.3.4.deb</a>	21.6 MB
 <a href="#">sieve.apk</a>	359 KB
 <a href="#">Source code</a> (zip)	
 <a href="#">Source code</a> (tar.gz)	



<https://github.com/hafiz-ng/beetlebug>








## 2. Download and Drozer Agent APK file from the latest releases

<https://github.com/FSecureLABS/drozer/releases>

### 2.3.4

Updated missing tag & release as per [#177](#)

#### ▼ Assets 8

 <a href="#">drozer-2.3.4-1.noarch.rpm</a>	21.6 MB
 <a href="#">drozer-2.3.4.tar.gz</a>	21.8 MB
 <a href="#">drozer-agent-2.3.4.apk</a>	618 KB
 <a href="#">drozer-installer-2.3.4.zip</a>	38 MB
 <a href="#">drozer_2.3.4.deb</a>	21.6 MB
 <a href="#">sieve.apk</a>	359 KB
 <a href="#">Source code</a> (zip)	
 <a href="#">Source code</a> (tar.gz)	



## 3. Install Drozer APK

```
adb install drozer.apk
```

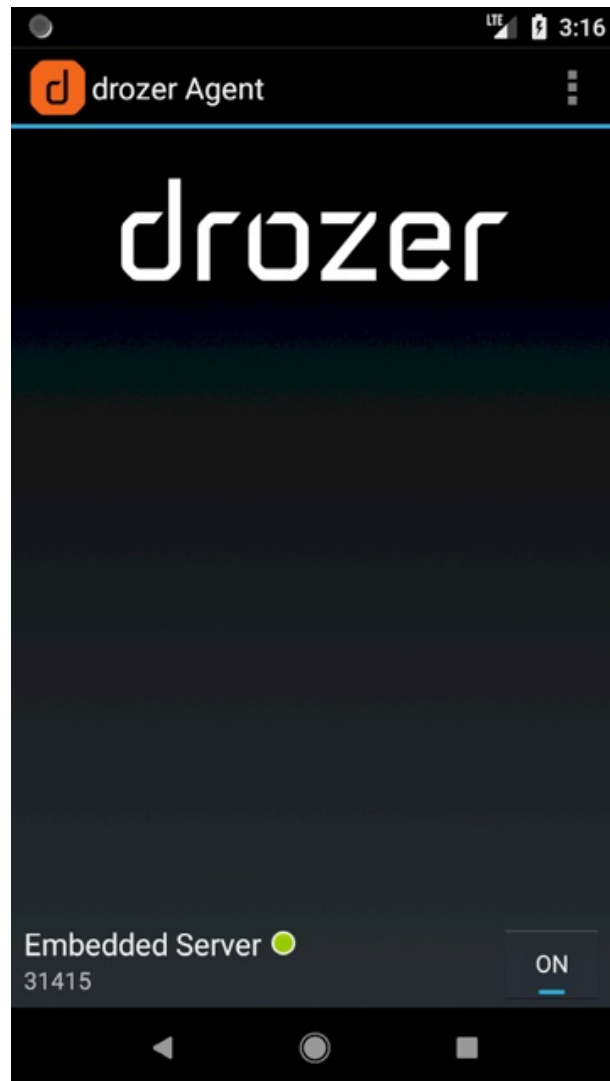
## Starting the Server

4. The Drozer Agent is running on port 31415, we need to port forward to establish a connection between the Drozer client and Agent.

```
adb forward tcp:31415 tcp:31415
```

<https://github.com/hafiz-ng/beetlebug>

5. Finally, launch the application and press the bottom "ON"



6. Connect to it with:

```
drozer console connect
```

<https://github.com/hafiz-ng/beetlebug>