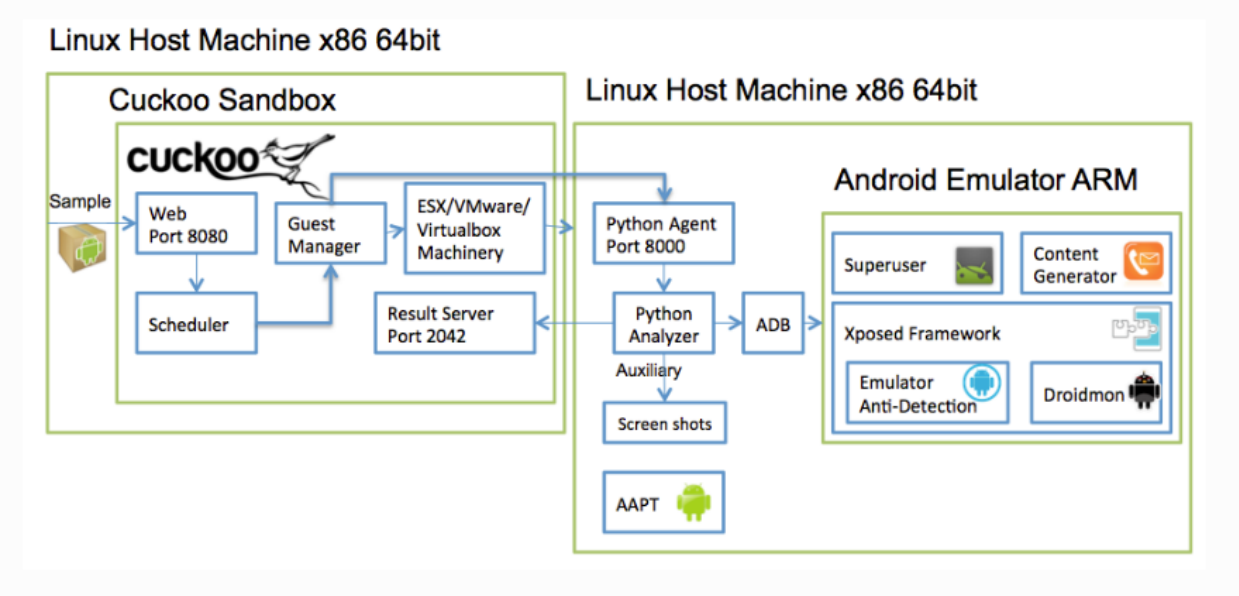
# CuckooDroid

## Installazione e Requisiti

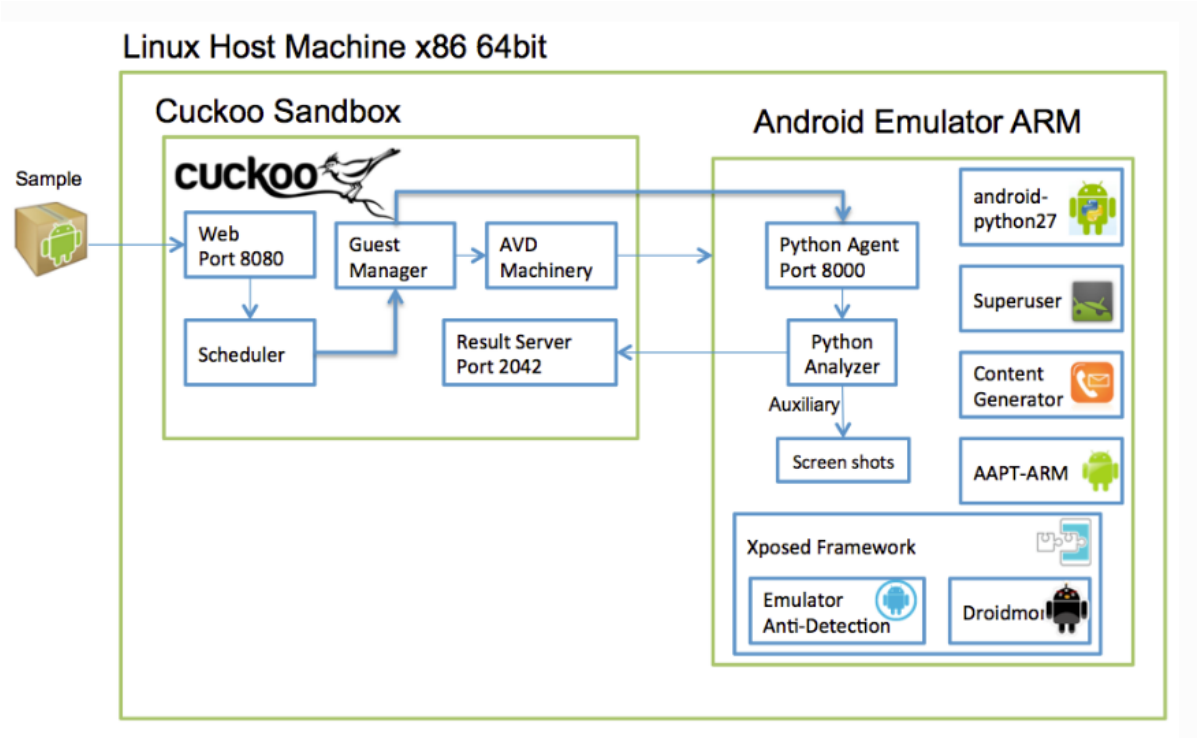
Configurazione di test : Ubuntu 18.04 e Android 4.1 per la guest machine.

Per la preparazione della macchina guest, è possibile effettuare tre diverse configurazioni:

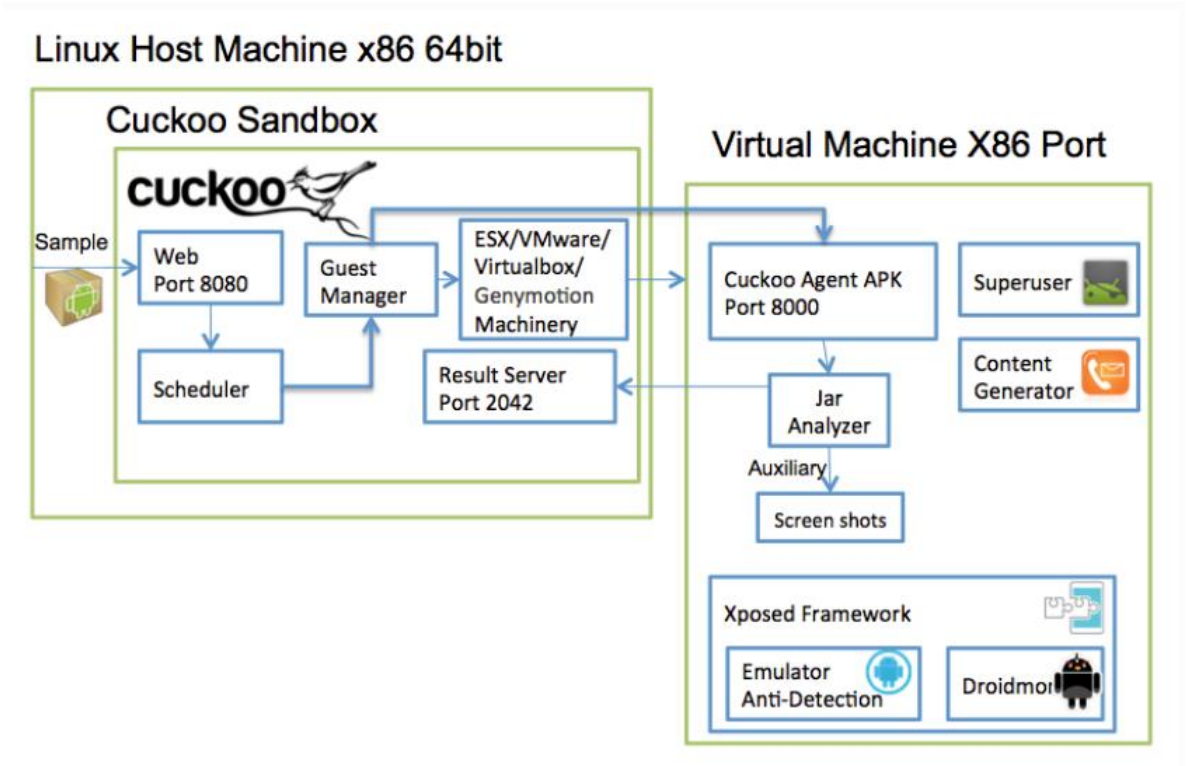
1. Android su macchina Linux




2. Emulatore Android



3. Dispositivo Android multi-piattaforma




**Download**

```
$ git config --global user.email "you@example.com"
$ git config --global user.name "Your Name"
$ git clone --depth=1 https://github.com/cuckoobox/cuckoo.git
cuckoo -b 1.2
$ cd cuckoo
$ git remote add droid https://github.com/idanr1986/cuckoo-droid
$ git pull --allow-unrelated-histories --no-edit -s recursive -X
theirs droid master
$ cat conf-extra/processing.conf >> conf/processing.conf
$ cat conf-extra/reporting.conf >> conf/reporting.conf
$ rm -r conf-extra
$ echo "protobuf" >> requirements.txt
```

**Modifica dei files Config**

conf/cuckoo.conf

# Specify the name of the machinery module to use, this module will
# define the interaction between Cuckoo and your virtualization software
# of choice.
machinery = avd

[resultserver]
# The Result Server is used to receive in real time the behavioral logs
# produced by the analyzer.
# Specify the IP address of the host. The analysis machines should be able
# to contact the host through such address, so make sure it's valid.
# NOTE: if you set resultserver IP to 0.0.0.0 you have to set the option
# `resultserver_ip` for all your virtual machines in machinery configuration.
ip = 127.0.0.1

conf/avd.conf

[avd]
#Path to the local installation of the android emulator
emulator_path = <add>  ( /home/USER/Android/Sdk/emulator/emulator )

#Path to the local installation of the adb - android debug bridge utility.
adb_path = <add>  ( /home/USER/Android/Sdk/platform-tools/adb )

#Path to the emulator machine files is located
avd_path = <add home_path>/.android/avd  ( /home/USER/.android/avd )

#name of the reference machine that is used to duplicate
reference_machine = aosx

```
# Specify a comma-separated list of available machines to be used. For each
# specified ID you have to define a dedicated section containing the details
# on the respective machine. (E.g. aosx_1,aosx_2,aosx_3)
#currently supports only 1 machine for network limitations
machines =aosx_1

[aosx_1]
# Specify the label name of the current machine as specified in your
# aosx_1 configuration.
label = aosx_1

# Specify the operating system platform used by current machine
platform = android

# Specify the IP address of the current virtual machine. Make sure that the
# IP address is valid and that the host machine is able to reach it. If not,
# the analysis will fail.
# its always 127.0.0.1 because android emulator networking configurations this the loopback of the host machine
ip = 127.0.0.1

#Specify the port for the emulator as your adb sees it.
emulator_port=5554

#10.0.2.2 is the loopback of the host machine very importent!!!
resultserver_ip = 10.0.2.2

resultserver_port = 2042
```

conf/auxiliary.conf

```
[sniffer]
# Enable or disable the use of an external sniffer (tcpdump) [yes/no].
enabled = yes
```

conf/processing.conf

[droidmon]
enabled = yes

[googleplay]
enabled = no
android_id = <add android_id>
google_login = <add google_login>
google_password = <add google_password>

[apkinfo]
enabled = yes
#Decompiling dex with androguard in a heavy operation and for a big dex's
#he can really consume performance from the cuckoo host ,so it's recommended to limit the size of dex that you will decompile
#decompilation_threshold=2000000

conf/reporting.conf

[reporthtml]
enabled = no
[reportandroidhtml]
enabled = yes

**Requisiti**

$ sudo apt-get install openjdk-8-jre libstdc++6:i386 libgcc1:i386 zlib1g:i386 libncurses5:i386

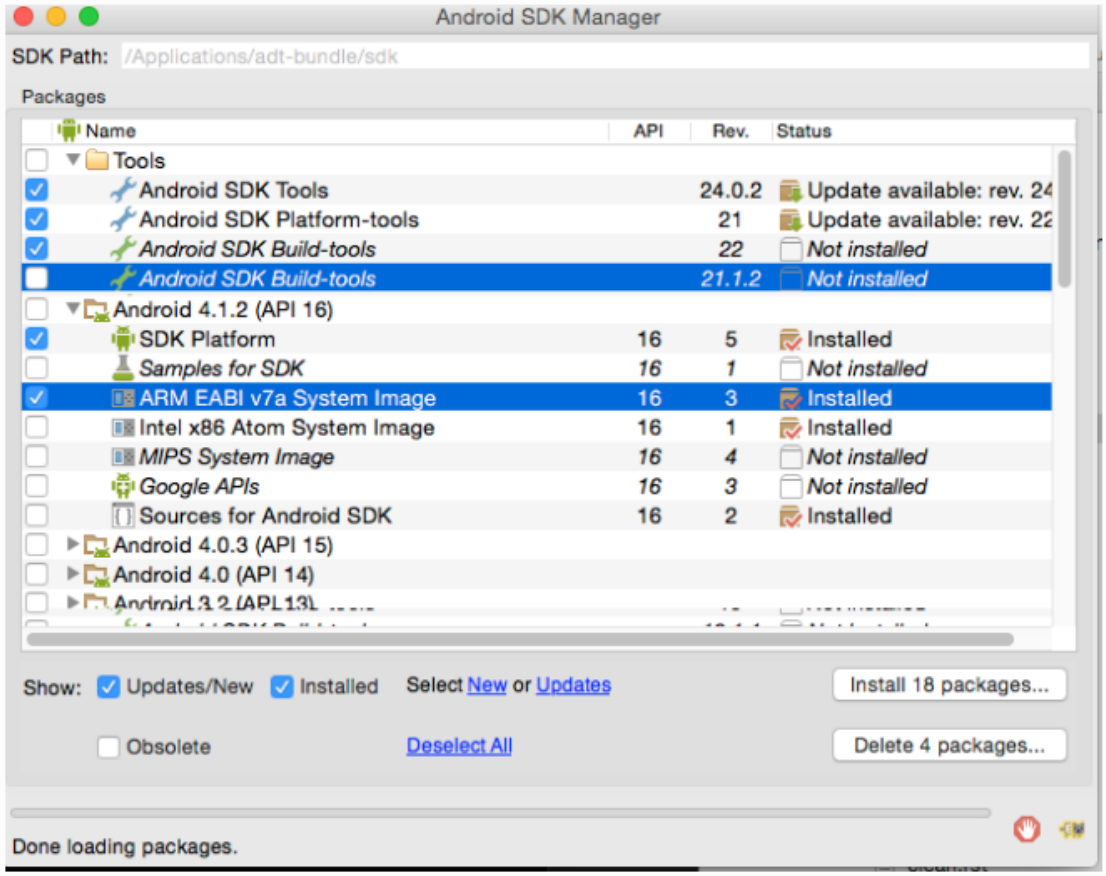
**Installazione Android SDK**

Effettuare il download della versione più recente delle SDK dal sito ufficiale di Android
Dopo aver effettuato il download, spostarsi nella cartella contente il file .tgz e usare i comandi:

$ tar -xvf android-sdk_r24.0.2-linux.tgz
$ cd android-sdk
$ tools/android

All'interno dell'Android SDK Manager, installare i seguenti componenti:

- Tools
  - Android SDK Tools
  - Android Platform-tools Tools
  - newest Android SDK Tools
- Android 4.1.2 (API 16)
  - SDK Platform
  - ARM EABI v7a System Image



**Aggiornamento dei Path**

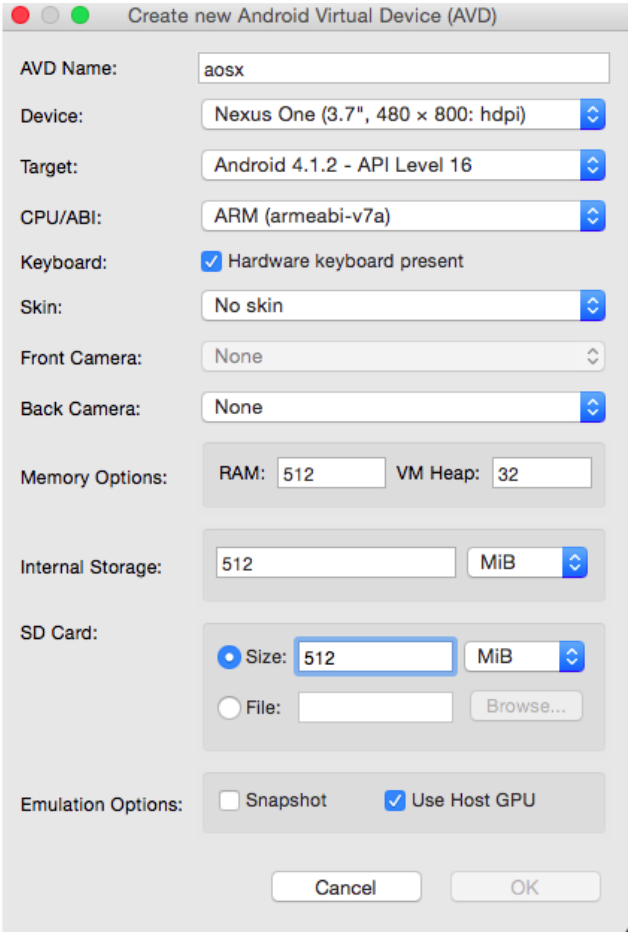
Scrivere nel file .profile

PATH="/home/USER/Android/Sdk/emulator:/home/USER/Android/Sdk/tools:/home/USER/Android/Sdk/build-tools/27.0.3:/home/USER/Android/Sdk/platform-tools:$PATH"

Per verificare il corretto valore della variabile d'ambiente PATH e della corretta posizione dell'emulatore:
$ echo $PATH

$ which emulator

**Creazione AVD (Android Virtual Device)**



Dopo aver creato l'avd, bisogna recarsi nel percorso "/home/USER/Android/Sdk/system-images/android-16/default/armeabi-v7a" e copiare il file system.img all'interno della cartella "/home/USER/.android/avd/aosx.avd" e rinominare il file in system-qemu.img.

Fatto ciò, bisogna avviare l'emulatore con il seguente comando:

$ emulator @aosx -writable-system -qemu -nand -system,size=0x1f400000,file=/home/USER/.android/avd/aosx.avd/system-qemu.img&

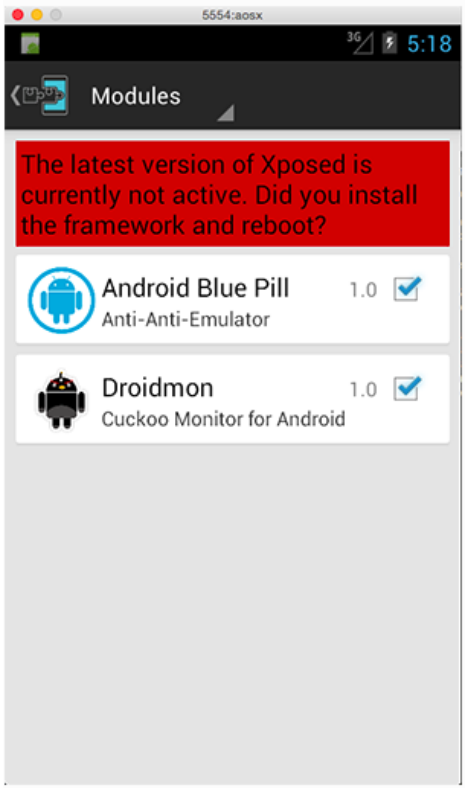
Quando l'emulatore ha completato l'avvio, bisogna avviare lo script shell : "/home/USER/cuckoo/utils/android_emulator_creator/create_guest_avd.sh" solo dopo aver modificato la riga 47 con " $ADB push ../../agent/android/python_agent/. /data/local/ " e attendere che vengano installati tutte le componenti all'interno dell'AVD.

**Rooting AVD (Android Virtual Device)**

All'interno dell'emulatore, effettuare le seguenti modifiche:

- Andare su settings→security→screenlock→none
- Andare su settings→display→sleep→30 minutes
- Avvia l'app Generate contacts
- Avvia l'app Superuser
- Avvia l'app xposedinstallar
- Andare in Modules, mettere le spunte in Droidmon e Android Blue Pill

A questo punto, andare su Framework→install
Dopo averlo installato, cliccare su cancel quando viene chiesto di riavviare il dispositivo, e poi cliccare su soft reboot e attendere che il dispositivo venga riavviato.
Al termine dell'avvio, chiudere l'emulatore.




**Fix per lanciare CuckooDroid**

Per fare funzionare correttamente CuckooDroid, è necessario effettuare alcune modifiche al codice per evitare errori e warning durante l'analisi.

- Modificare il file "/home/USER/cuckoo/analyzer/android/lib/api/adb.py" , in particolare la funzione *execute_sample* alla riga 111 con la seguente: **proc = subprocess.Popen("/system/bin/am start -n"+ package+"/"+activity, stdout=subprocess.PIPE, stderr=subprocess.PIPE, shell=True, executable="/system/bin/sh")** e commentare la riga sottostante.
- Modificare il file "/home/USER/cuckoo/modules/processing/network.py" , in particolare alla riga 596 con la seguente: **results = Pcap(self.pcap_path).run()**
- Modificare il file "/home/emiliano/cuckooTest/cuckoo/modules/reporting/mongodb.py" , in particolare all'ultima riga con la seguente: **self.conn.close()**