

Forum:[Hack Forums](#)

Thread: [My current stack for android emulator bot testing](#)

99overall - March 20, 2024, 2:55 pm

Emulator automation is very fucking hard to do and nobody wants to share how they do it so I thought I'd tell you guys what I've been working with so far. I'm trying to get traffic from dating apps so maybe this will help someone. I'm honestly nowhere close to a finished product but this could help you get started.

Basic:

- Android studio emulator (flamingo) 2022.2.1
- Nexus 4 AVD
- "Factory" fingerprint and props from XDA (I'll put it in the comments if anybody wants to use it)
- MagiskOnEmulator
- MagiskHide props config
- Appium with python

Why:

- Funny story for the emulator, I found that the latest version of the android studio one's proxy function was broken so I spent a whole 72 hours of my life scouring the internet for why this was the case. When I started this project I honestly didn't even know how proxies worked so I thought maybe I did something wrong on my end. Ended up finding through some random discord that someone found 2022.2.1 had a working proxy function so I just started using that. It was all in vain though because android emulator only officially supports http proxies which are useless to us because most apps won't pass through them so we need to use a third party solution for SOCKS.
- No reason, just was the device I was able to find a fingerprint for
- So yeah, you can't just make up some random bs to add for your fingerprint, you need to find what the real device you're emulating uses. You can find these on forums pretty easily and if you use magiskhide props config I think they have a built in DB of verified fingerprints but I haven't been able to access that through the emulator. Maybe I'm wrong too, documentation on this module isn't great and I still barely know what I'm doing.
- This is just a different way to install magisk for the emulator, as you can't do it the normal way. Just google this on github, it's pretty straightforward but they don't tell you in the docs that your APK has to be named just "magisk.apk", not the one you'll download initially that has the version number and everything. Took me a while to figure this out as the console window that pops up and tells you what's wrong disappears so fast that you can't even see it.
- Android studio emulators won't allow you to change "ro" or read only props by default, such as "ro.build.fingerprint", and magiskhide props is a module for magisk that's supposed to be a solution to this. It's finicky and I can go into detail on how I've got it (kind of) working if you want, but I'm actually still testing this and seeing what I can and can't change and how reliable it actually is. The syntax in the configuration file you'll have to add is completely unclear so it was a real pain to get this set up. If anyone knows a better way to change props please share.
- Actually haven't even tested any scripting with appium yet but I plan to use this library once I have a working emulator that can bypass detection on my target apps. The furthest I've gotten scripting-wise is just using python's subprocess module to launch the emulator from command line when I run the program and to get the coordinates of my proxy based on its IP and add that as the device's location.

That's about all I've got. This project is very difficult and I still don't really know wtf is going on or

what I'm doing. Bonus tip, you can use the program "MobSF" to check apk's for vulnerabilities and to some surface level extent how they check for emulators. If anybody is working on something similar I'd love to get in touch and maybe team up idk. This is really more of a resume project for me at this point.

Fingerprint and props for nexus 4:

```
ro.product.model: Nexus 4 >
ro.product.brand: google >
ro.product.manufacturer: LGE >
ro.product.board: MAKO - Test to see if it fucks anything up after making sure everything else
works
ro.product.name: occam >
ro.product.device: mako >
ro.build.product: mako > added to custom props to be safe (obsolete, use above device
parameter instead)
ro.build.version.release: 4.2.1 >
ro.build.id: JOP40D >
ro.build.display.id: JOP40D - Test to see if its "CONFDISPLAY" next round
ro.build.tags: release-keys - in magisk sensitive props
ro.build.date: Tue Nov 20 00:56:01 UTC 2012
ro.build.date.utc: 1353372961
ro.build.description: occam-user 4.2.1 JOP40D 533553 release-keys >

ro.build.fingerprint: google/occam/mako:4.2.1/JOP40D/533553:user/release-keys >
```

BrowningL - May 6, 2024, 9:02 am

Thanks for being the one person happy to share their knowledge. This is very interesting!

Do you have to do device emulation? Or is some System On Chip solution not viable for dating apps?

Honeydew826 - October 24, 2024, 1:11 pm

<https://hackforums.net/showthread.php?tid=6283236&pid=62553573#pid62553573>

Is this usable for KYC bro?