**Day 3**

**Exploitation Analyst**

**Hacking the SSL Network protocol:**

**First understand what is SSL Pinning?**

SSL pinning is the process of associating a host (like an API server) with a specific SSL/TLS certificate or public key. Instead of trusting any certificate that is signed by a trusted Certificate Authority (CA), the app only accepts a specific certificate (or public key). This adds an extra layer of security.

**Why Use SSL Pinning?**

By default, HTTPS trusts any certificate issued by a trusted CA. If an attacker can:

* Install a malicious root certificate (on the device or network),
* Or intercept traffic using a forged certificate,

...they could decrypt and modify the data in transit.

SSL pinning stops this by rejecting all certificates except the one(s) you trust.

**SSL Pinning bypass by overwriting packaged CA certificate with custom CA certificate:**

Some mobile applications **bundle their own CA certificates** for SSL pinning (instead of using the system trust store). This helps them verify the server’s identity and block MITM attacks — but it can be bypassed if the cert is replaced.

Bypass Concept

If the app uses a custom CA certificate (typically found in the assets or res/raw directory of the APK), you can:

1. Extract the APK (e.g., using APK Studio or apktool).
2. Locate the bundled CA certificate inside directories like /assets/.
3. Replace it with your own custom CA certificate (generated with tools like Burp Suite or mitmproxy).
4. Repack and sign the APK, then install it on the device.

Result

The app will now trust your MITM proxy’s certificate, allowing you to intercept HTTPS traffic without triggering SSL pinning errors.

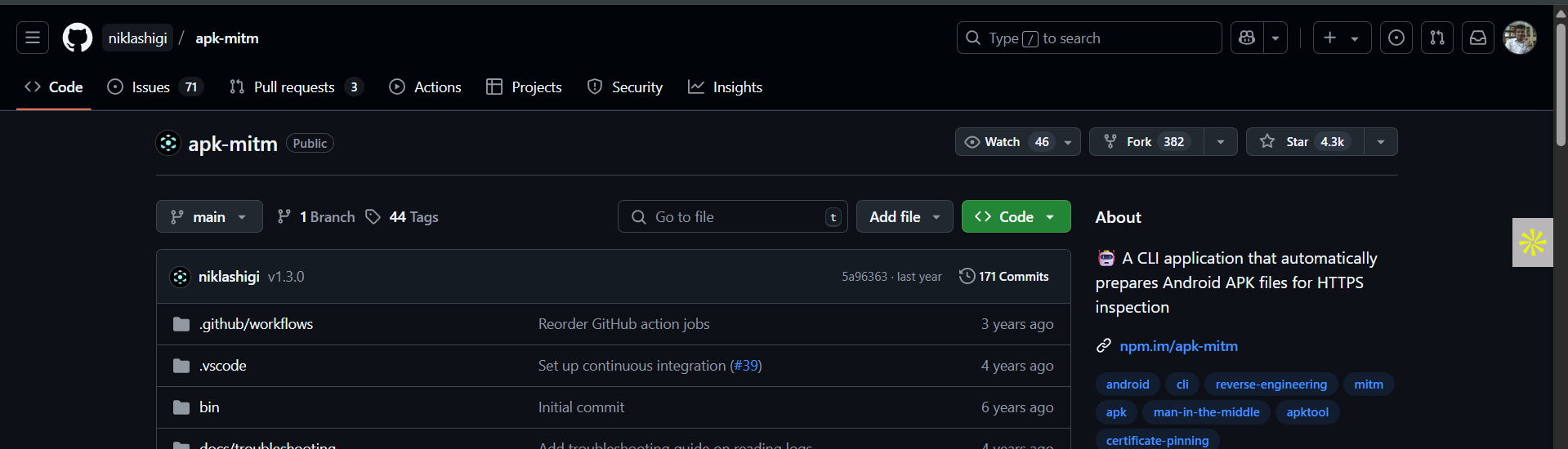
Tools Used

* apk-mitm: Automates certificate injection and patching of common pinning libraries.
* apktool/APK Studio: For manual unpacking and certificate replacement.

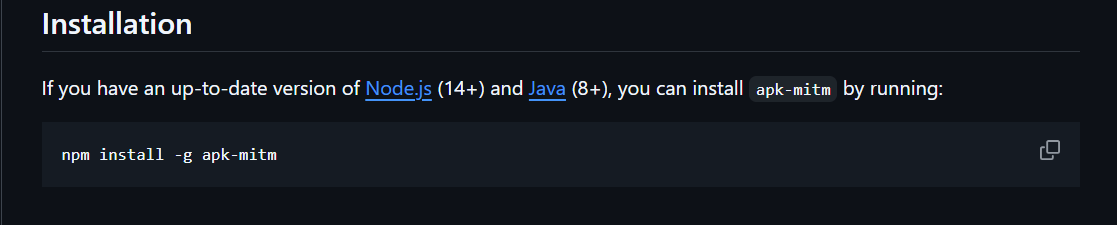
**Bypassing SSL pinning using the tool apk-mitm:**

<https://youtu.be/odGnlw4MZx0?si=glwPNCqeFDP22PRv>

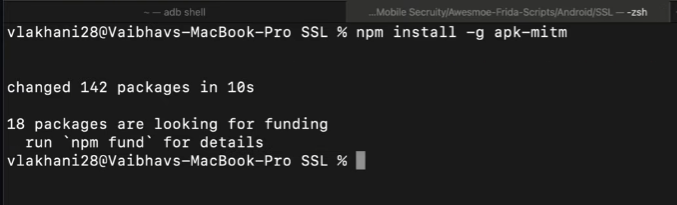
First visit the github repo of this: https://github.com/niklashigi/apk-mitm



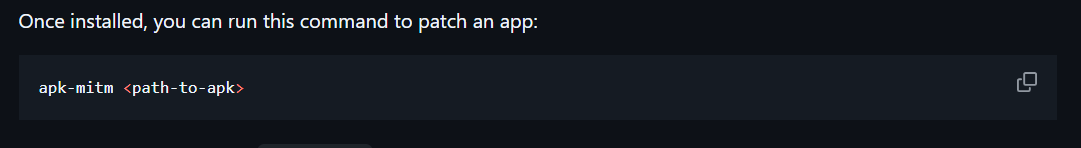
Install the tool using the following command: (in terminal as shown next)

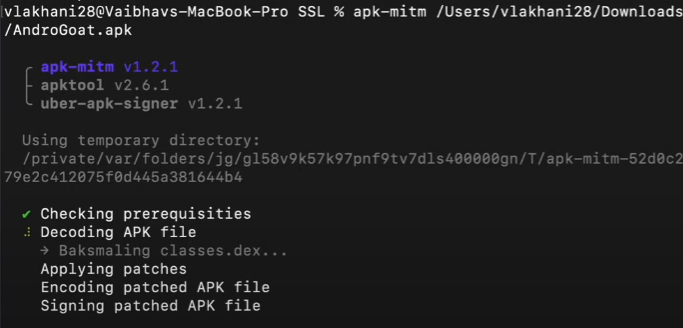


In terminal:

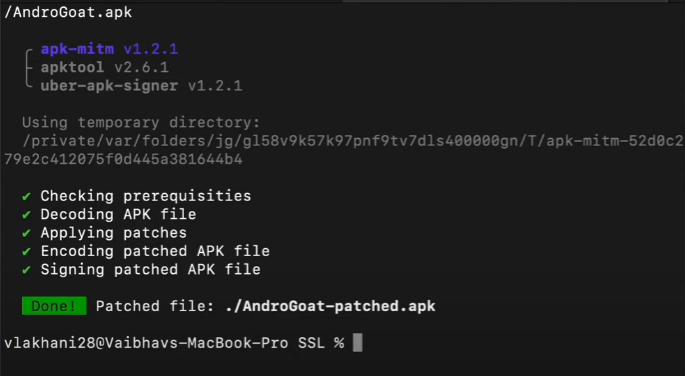


Now, you can run it on the .apk file using the following command: (shown next in terminal)

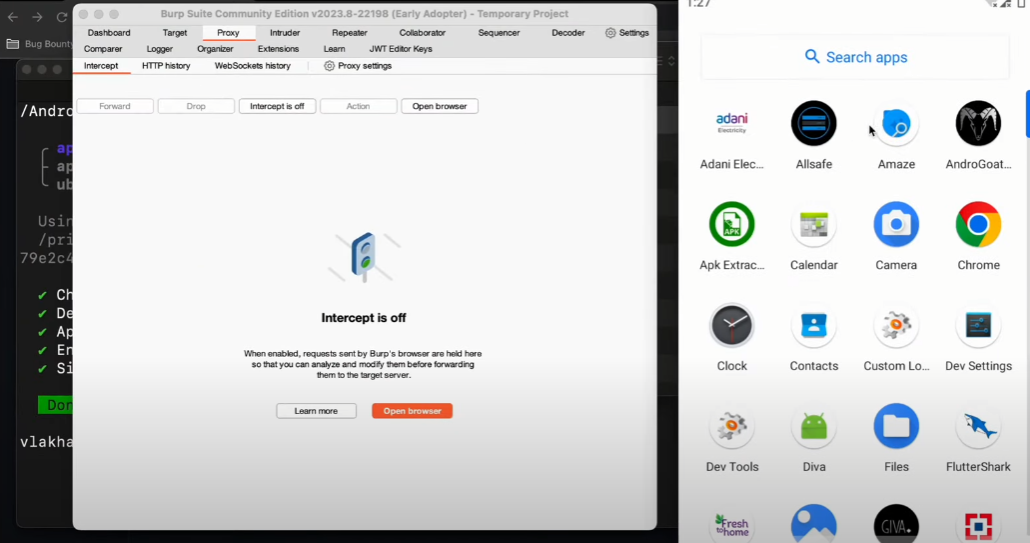




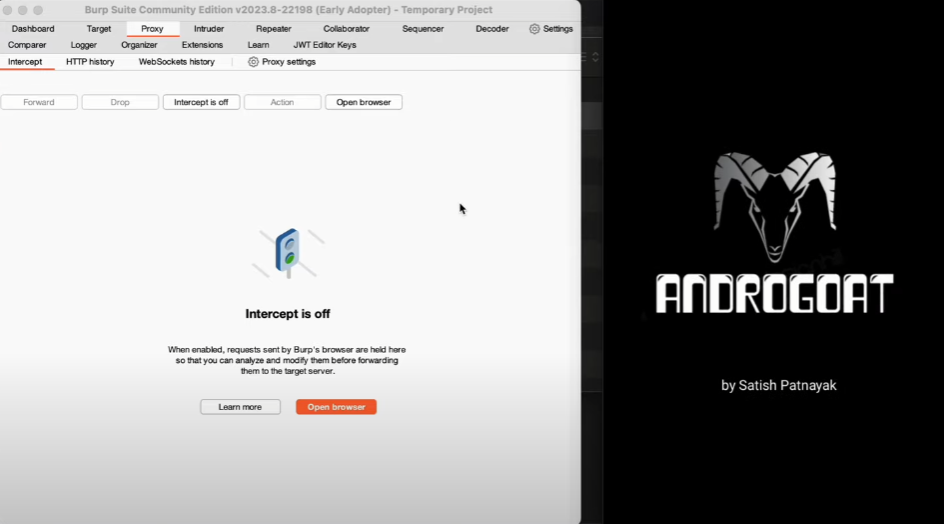
Once, it is done, it will give us a patched .apk file:

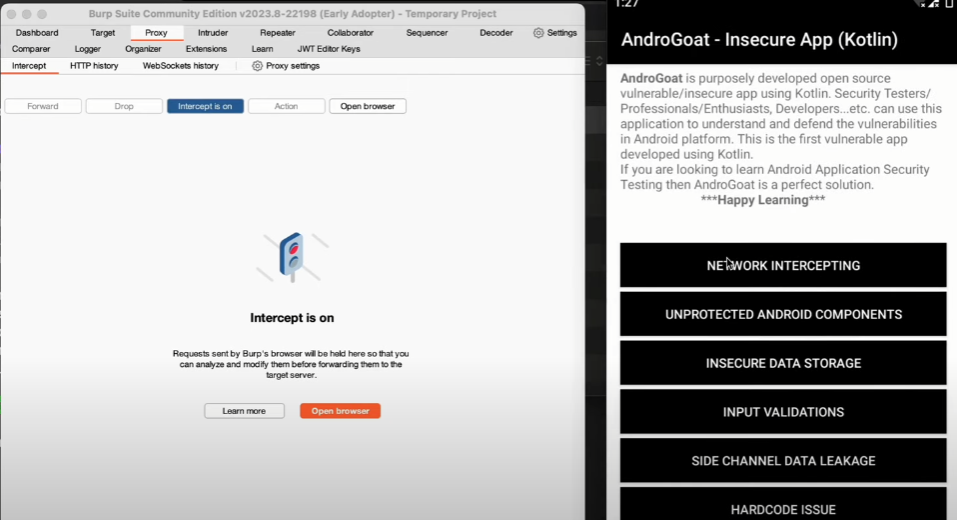


Now, we will try to un install the original .apk file from the android simulation. For this we will need the Burp Suite

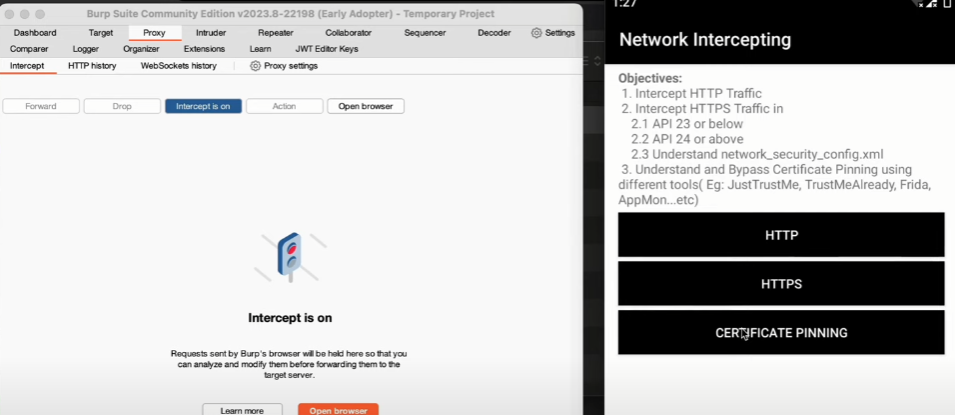


Here, open the Androgoat app:

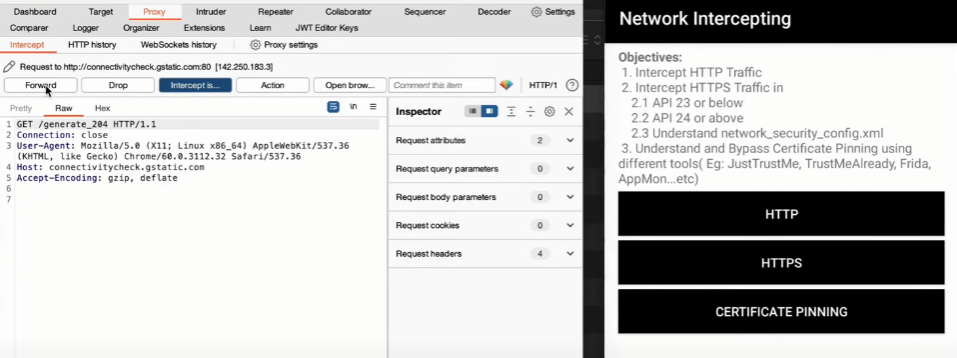


Keep your intercept on, and then click on the “Networking intercept” option:

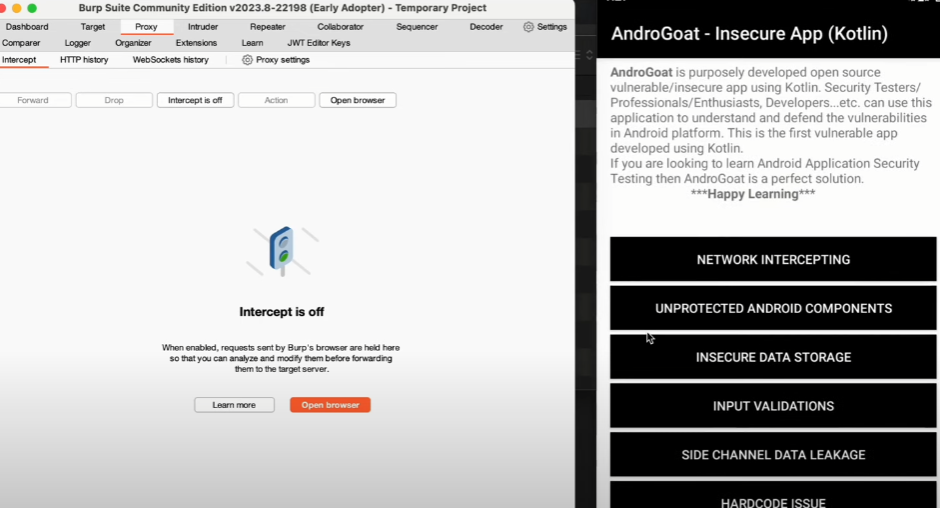
Following screen will appear:



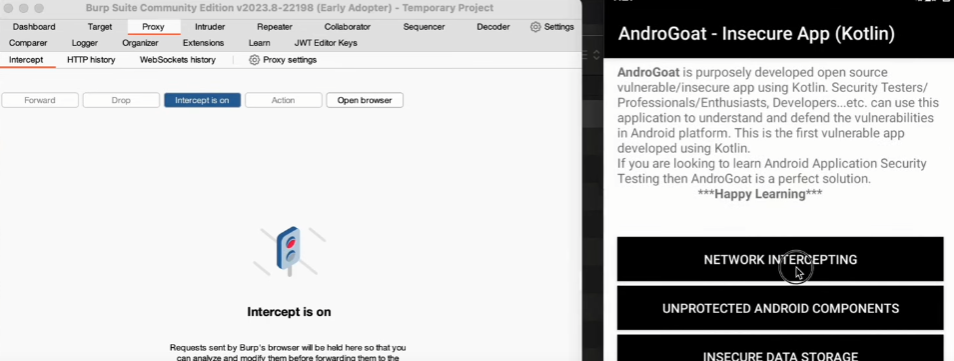
Click on the “Certificate pinning”, and then check if intercept is captured or not using the Burp Suite. As, we see below we are not able to intercept that:



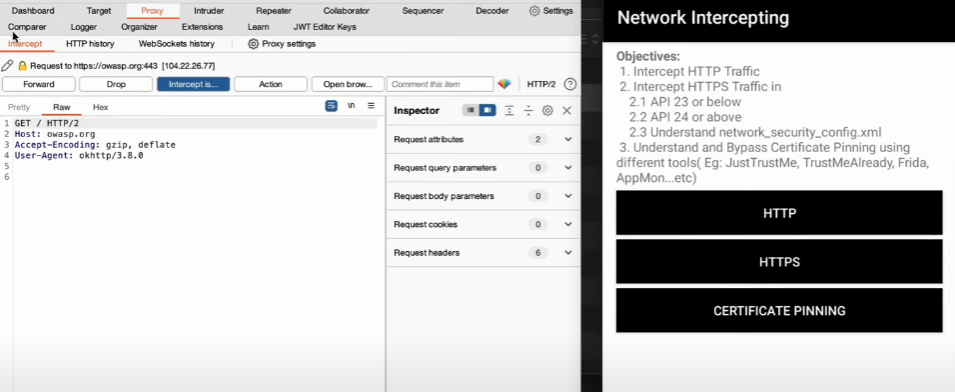
So, now, we can uninstall this old .apk file and install the new, manually:



Keep intercept on and then click on the network intercepting option:



Then, click on the certificate pinning option as shown: Clearly, intercept get captured and thus we managed to by pass the SSL pinning.



--The End--