Man in the middle attacks demo

+ Reverse Engineering

Attacks on users - what they do, what their credentials are

Communication types:

- Browser + HTTP
- Browser + HTTPS
- Browser + HTTPS + HSTS (HTTP Strict Transport Security)
- App + HTTP/HTTPS
- App + certificate pinning

Simplest how-to

1. You must earn the power of hacking by passing the steps with your own skills and knowledge
2. You must have two network adapters - either ethernet+WiFi, or a WiFi dongle
3. Download and install Burp, understand how Burp certificate works
4. Either set up Burp proxy (it's IP and host) to your phone's advanced WiFi settings

   Or setup transparent proxying via Iptables (on Linux):

   ```
   iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8080
   iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-ports 443
   ```

   And in Burp, listen on these ports on the interface of your WiFi adapter

Attacks on applications - functionality, structure, logic, security, keys..

- App + installed certificate
- Reverse Engineering
- MITM

Reverse Engineering

Apktool
http://ibotpeaches.github.io/Apktool/

A tool for Reverse Engineering resources nearly to their original form

- Layouts, colors, strings...
- Android Manifest
- Smali code - debuggable, better readable than byte code

Cmd command: *apktool d app_name.apk*

Reverse Engineering

Dex2jar

- Decompiles compiled Java classes to .jar files to nearly their original form
- Jar files can be opened in tools like *Luyten*

https://github.com/pxb1988/dex2jar

https://github.com/deathmarine/Luyten/releases

1. Unzip .apk file
2. sh dex2jar/d2j-dex2jar.sh unzipped_dir/classes.dex
3. Drag and drop created classes.jar to Luyten and read :)

Reverse Engineering

Where to hide secret keys in Android apps?

Native code!!!
Is not Java and is decompiled much harder :)