0x28 Mobile Security

Exercise Session

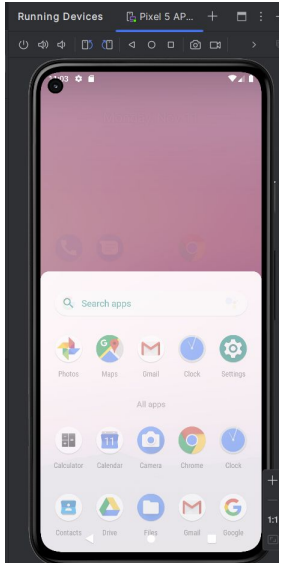
hexhive

EPFL

Lab requirements

- jadx
  - Used to decompile Android APKs

- [Optional] Android Studio
  - (used to for android app development)
  - Can emulate Android devices! (→ good for testing your solutions)
  - 1.1G download size
    - can take a while so feel free to start now :)



Recap

Android Application Package (APK)

An APK is a zip file (with a magic format)

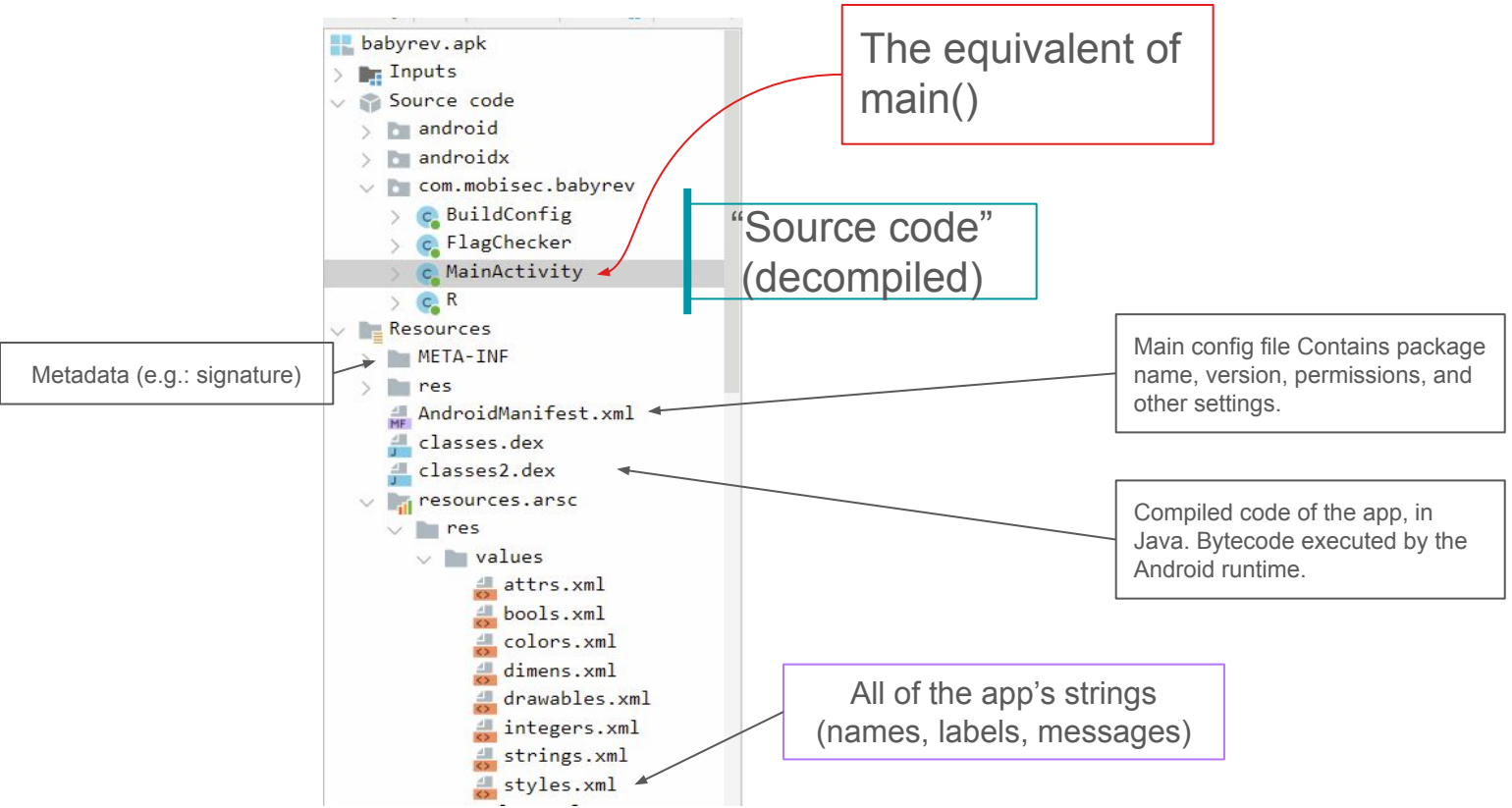
```
$ unzip app.apk
```

Content:

- AndroidManifest.xml (compressed)
- classes.dex (raw Dalvik bytecode)
- resources.arsc (compressed)
- res/*.xml (compressed)

Understanding APKs

1. Decompile using jadx



Reverse engineering

3 challenges

3 flags

Find them all :)

Ex. 1: babyrev

```
String flag = flagWidget.getText().toString();
boolean result = FlagChecker.checkFlag(MainActivity.this, flag);
if (result) {
	msg = "Valid flag!";
} else {
	msg = "Invalid flag";
}
```

Yay, easy, just look at what `checkFlag()` does

Ex. 1: babyrev

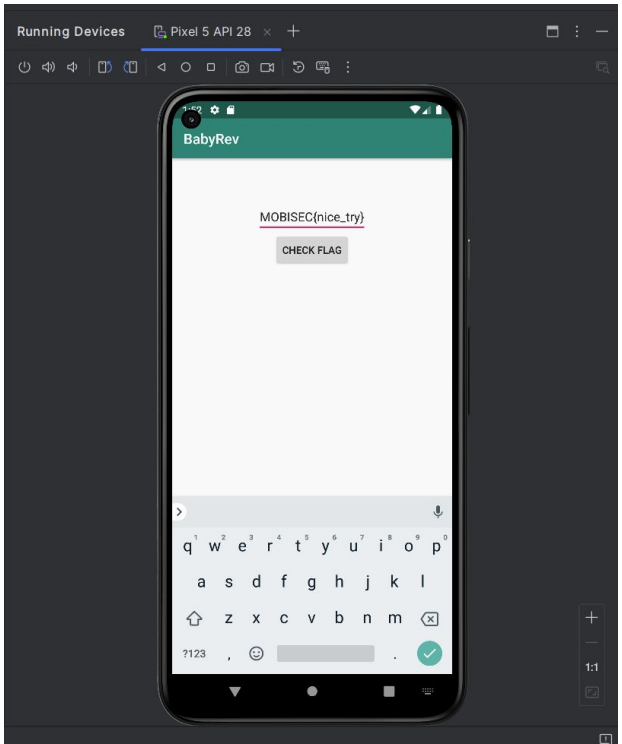
```
public static boolean checkFlag(Context ctx, String flag) {

      if (!flag.startsWith("MOBISEC{") || new
StringBuilder(flag).reverse().toString().charAt(0) != '}' || flag.length() != 35 ||
!flag.toLowerCase().substring(8).startsWith("this_is_") || !new
StringBuilder(flag).reverse().toString().toLowerCase().substring(1).startsWith(ctx.get
String(R.string.last_part)) || flag.charAt(17) != '_' || flag.charAt((int) (getY() *
Math.pow(getX(), getY()))) != flag.charAt(((int) Math.pow(Math.pow(2.0d, 2.0d), 2.0d))
+ 1) || !bam(flag.toUpperCase().substring(getY() * getX() * getY(), (int)
(Math.pow(getZ(), getX()) - 1.0d))).equals("ERNYYL") || flag.toLowerCase().charAt(16)
!= 'a' || flag.charAt(16) != flag.charAt(26) || flag.toUpperCase().charAt(25) !=
flag.toUpperCase().charAt(26) + 1) {
          return false;
      }
      String r = getR();
      return flag.substring(8, flag.length() - 1).matches(r);
   }
```

Oh.

Validating your solution - 3 options

1. **Try it on the app itself!**
2. Submit to the actual challenge (need to create an account)
3. Ask your TAs :)

Start a device emulation under SDK 28



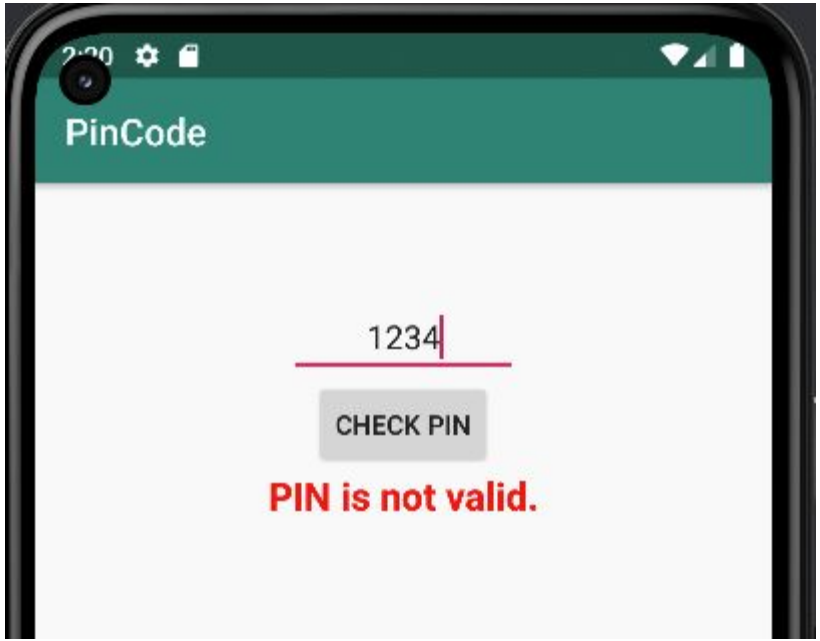
and just drag&drop the .apk

Ex. 1 - Solution

Will be discussed during the exercise session :)

Ex. 2: pincode

- Conceptually less difficult
- Exposes (yet again) the risk of client-side verification



Ex. 2: pincode

```
if (pin.length() != 6) {
    return false;
}
try {
    byte[] pinBytes = pin.getBytes();
    for (int i = 0; i < 25; i++) {
        for (int j = 0; j < 400; j++) {
            MessageDigest md = MessageDigest.getInstance("MD5");
            md.update(pinBytes);
            byte[] digest = md.digest();
            pinBytes = (byte[]) digest.clone();
        }
    }
    String hexPinBytes = toHexString(pinBytes);
    return hexPinBytes.equals("d04988522ddfed3133cc24fb6924eae9");
}
```

So many iterations ! This has to be unbreakable, right?

Ex. 2 - Solution

Will be discussed during the exercise session :)

Ex. 3: gnirts (bonus)

- More complicated, because of obfuscation!
  - Conceptually the same as ex. 1
  - Introduces a lot of creative techniques

- Thoroughly follow every lead, and you'll get there :)