Assignment #6

CPEN 442

November 30

Kaibo Ma (32400129)
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, Canada

I. Problem #1

From looking at the Android manifest upon decompiling the zip, if you take the app id link and append it to google play store you obtain the app: https://play.google.com/store/apps/details?id=com.geniemobile.app1744009&hl=en

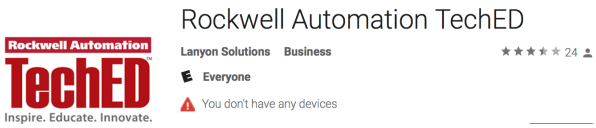

Fig.1 The Android App

II. Problem #2

To find the number of classes in the application, I did a word out for the amount of smali files inside the decompiled root folder with the bash command:

find ./ -name ".smali" | wc –l.*

This outputted 11011 smali class files, thus there is 11011 separate classes in the application.

III. Problem #3

Like that of java, when looking at smali files, they must construct an instance of a Cipher in java crypto library. Thus in the terminal I used command:

grep -r "Cipher;->getInstance(" ./

To find the instances of cipher usage. I found 7 instacnes of this call.

IV. Problem #4

I chose the instance:

.//smali/com/genie_connect/android/utils/crypt/Crypt2.smali: invoke-static {v8}, Ljavax/crypto/Cipher;->getInstance(Ljava/lang/String;)Ljavax/crypto/Cipher;

By looking at that file and where the instance is called, there is this code right above it:

const-string v8, "AES/CBC/PKCS5Padding"

Now we know it is AES CBC Encryption with PKCS5 Padding.

Then when we look at where this function is called in the class, we see that it resides inside another function called:

.method public static decrypt(Ljava/lang/String;Ljava/lang/String;)Ljava/lang/String;

Clearly it shows that it is in DECRYPT_MODE.

From going up the stack calls of functions we go through multiple calls. I found that the decrypt function is called within another function called LocalDecrypt(), this function is a function that overrides an Abstract class. In this function there is a call for:

calculateLocalKey(Landroid/content/Context;)Ljava/lang/String;

In this function, there is an init function call:

.method static constructor <clinit>()V

In this function we can clearly see that the Key is passed in:

```
# direct methods
.method static constructor <clinit>()V
    .locals 1

    .prologue
    .line 12
    const/16 v0, 0x10

    new-array v0, v0, [B

    fill-array-data v0, :array_0

    sput-object v0, Lcom/julysystems/appx/AppXAESEncription;->keyValue:[B

    return-void

    :array_0
    .array-data 1
        0x46t
        0x41t
        0x41t
        0x41t
        0x37t
        0x35t
        0x39t
        0x44t
        0x36t
        0x37t
        0x41t
        0x35t
        0x30t
        0x39t
        0x44t
        0x32t
    .end array-data
.end method
```

Fig. 2 The Key is shown in array-data 1

For the IV:

Back in the original function call of getInstance(), there is a local variable iv stored in buffer v1. This is where the iv is passed into the cipher:

```
.local v1, "iv":[B
    invoke-virtual {v4, v1}, Ljava/security/SecureRandom;->nextBytes([B]V
```

This uses secureRandom and nextBytes() which creates a random IV like what I used in Assignment 3 VPN.

V. Problem 5

It uses AES CBC mode with PKCS5 Padding and has a Random IV. Even if the Key is hardcoded into the program, it would be secure as the IV is different and unpredictable every time. Because it is using CBC and assuming there is no integrity checking , it is possible for Trudy to change messages without Alice or Bob knowing.

VI. Problem 6

To fix this issue. We can either create integrity checking by using HMAC. Or even better, using AES GCM with PKCS5 Padding, it would already contain integrity checking as GCM already handles integrity checking within the algorithm