13) nojumpstarts

There was no text for this challenge on Moodle.

Solution

We are given an apk file an a Python script. We decompile the apk with jadx via command

*jadx -d out victim.apk*

In the usual output folder, this time around, apart from the Main Activity, there are four classes: A, B, C and the Main file. Let’s analyze each one of these.

Inside Main file, there are the private and the public key of RSA, which gets encoded in two different public cryptography schemas. Basically, it seems we are signing the message and building the intent via message this way, signing with author and message itself.

Basically, the A/B/C classes get called in order via the creation of an Intent and its extra, creating the flag and then signing both the message and the author; if everything is done correctly, the authentication is done well, considerig the activity result each time.

Infact, first there is the buildIntent from A to B, then from B to C and if we go into C, we reply the correct flag from the Main Activity.

At the end of the day, also looking inside the MainActivity, it seems like we are calling the Intent setting the flag this way, then expecting a possible result, hence getting a possibly right flag. So, the logic would be to call these methods in order and then get the flag inside our malicious app implementation. What we have to do would be use the signing logic to get at least to C.

Here is a possible code outline for getting the flag:

1. Create a new app with the same package name as the legitimate app.
2. Generate the private and public keys for each activity in the chain.
3. Use the correct keys to sign the messages from each activity.
4. Pass the signed messages to the next activity in the chain.
5. Repeat steps 3 and 4 until you reach the C activity.
6. The C activity will return the flag.

So, let’s start from building the chain; we would need to copy the entire code of Main in order to get to the solution, otherwise, we wouldn’t be able of recostructing the flag properly.

Basically, a right code would follow this implementation:  
  
package com.example.nojumpstarts;  
import android.app.Activity;  
import android.content.ComponentName;  
import android.content.Intent;  
import android.os.Bundle;  
import android.util.Log;  
  
import java.util.Objects;  
  
public class MainActivity extends Activity {  
  
 private static final String *TAG* = "MOBIOTSEC";  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 Log.*i*(*TAG*, "Created the activity");  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
  
 String msg = "Main-to-A/A-to-B/B-to-C";  
  
 // Use the buildIntent method from Main to create the intent  
 Intent data = null;  
 try {  
 data = Main.*buildIntent*("Main", "C", null);  
 } catch (Exception e) {  
 throw new RuntimeException(e);  
 }  
  
 Log.*i*(*TAG*, "Signed the message and created intent");  
  
 if (Objects.*requireNonNull*(data).resolveActivity(getPackageManager()) != null) {  
 // We then set authmsg and authsign as extras  
 data.putExtra("authmsg", msg);  
 // We can then sign the message with the code already present inside "Main"  
 // The following has to be try-catch surrounded  
 try {  
 data.putExtra("authsign", Main.*sign*(msg));  
 } catch (Exception e) {  
 Log.*e*(*TAG*, Log.*getStackTraceString*(e));  
 }  
  
 Log.*i*(*TAG*, "Sent intent successfully");  
 startActivityForResult(data, 1);  
 }  
 }  
  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data){  
 super.onActivityResult(requestCode, resultCode, data);  
  
 if (data != null) {  
 String flagValue = data.getStringExtra("flag");  
  
 if (flagValue != null) {  
 Log.*i*(*TAG*, flagValue);  
 } else {  
 Log.*e*(*TAG*, "Flag not present in the intent");  
 }  
 } else {  
 Log.*e*(*TAG*, "Received null intent");  
 }  
 }  
}

Infact, when launching the command via the Python script checker, the flag gets printed:  
  
> python3 nojumpstarts\_checker.py victim.apk app-debug.apk

…….

12-13 09:29:29.923 13801 13801 I MOBIOTSEC: Created the activity

12-13 09:29:29.936 13801 13801 I MOBIOTSEC: Signed the message and created intent

12-13 09:29:29.939 13801 13801 I MOBIOTSEC: Sent intent successfully

12-13 09:29:30.071 13801 13801 I MOBIOTSEC: FLAG{virtus\_unita\_fortior}