Appendix 2

1)   K3 five card stud (pname: com.showhand.k3)

This app has some malicious activities, such as obtain SIM operator name, obtain SIM serial number, obtain SIM status, obtain geo-location information, obtain WiFi configuration, obtain Bluetooth information and obtain device information,etc.

a)   obtain SIM operator name

```
try {
    Object v41 = context.getSystemService("phone");
    ((Map)v4).put("operator", ((TelephonyManager)v41).getSimOperatorName());
    ((Map)v4).put("carrier", ((TelephonyManager)v41).getNetworkOperatorName());
}
catch(Exception v2_1) {
}
```

b)   obtain SIM serial number

```
public static String i(Context arg1) {
    String v0_1;
    try {
        v0_1 = arg1.getSystemService("phone").getSimSerialNumber();
    }
    catch(Exception v0) {
        v0_1 = null;
    }
```

c)   obtain SIM serial number

```
if(v0_2 == null) {
    goto label_182;
}
else if(((TelephonyManager)v0_2).getSimState() == 5) {
    v0_3 = ((TelephonyManager)v0_2).getNetworkOperatorName();
}
```

d)   obtain geo-location information

```
Float localFloat = Float.valueOf(1000.0F * paramLocation.getAccuracy());
Long localLong1 = Long.valueOf(1000L * paramLocation.getTime());
Long localLong2 = Long.valueOf((10000000.0D * paramLocation.getLatitude()));
Long localLong3 = Long.valueOf((10000000.0D * paramLocation.getLongitude()));
localHashMap.put("radius", localFloat);
localHashMap.put("lat", localLong2);                          s
```

e)   obtain WiFi configuration

```
str10 = localb.c();
WifiInfo localWifiInfo1 = ((WifiManager)localContext1.getSystemService("wifi")).getConnectionInfo();
if (localWifiInfo1 == null) {
    break label768;
}
```

f)   obtain Bluetooth infomation

```
try
{
    BluetoothAdapter localBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
    if ((localBluetoothAdapter != null) && (!localBluetoothAdapter.isEnabled())) {
        return "";
    }
    String str = localBluetoothAdapter.getAddress();
    return str;
```

g)   obtain device information

```
if (localTelephonyManager != null)
{
    String str1 = localTelephonyManager.getDeviceId();
    str2 = str1;
    localObject = str2;
```

2) 天堂戰記 (pname:com.enjoygame.ttwyft)

This app has some malicious activities, such as obtain SIM status, obtain geo-location information, obtain WiFi configuration, obtain device information, read phone number, and connect to the Internet, etc.

a) obtain SIM status

```java
private static boolean u(Context paramContext)
{
    return ((TelephonyManager)paramContext.getSystemService("phone")).getSimState() == 5;
}
```

b) read phone number

```java
if (h(paramContext))
{
    TelephonyManager localTelephonyManager = (TelephonyManager)paramContext.getSystemService("phone");
    if (localTelephonyManager != null) {
        return localTelephonyManager.getLine1Number();
    }
}
return "";
```

c) obtain geo-location information

```java
HashMap localHashMap = new HashMap();
Float localFloat = Float.valueOf(1000.0F * paramLocation.getAccuracy());
Long localLong1 = Long.valueOf(1000L * paramLocation.getTime());
Long localLong2 = Long.valueOf((10000000.0D * paramLocation.getLatitude()));
Long localLong3 = Long.valueOf((10000000.0D * paramLocation.getLongitude()));
localHashMap.put("radius", localFloat);
localHashMap.put("lat", localLong2);
localHashMap.put("long", localLong3);
localHashMap.put("time", localLong1);
```

d) obtain WiFi configuration

```java
WifiManager localWifiManager = (WifiManager)paramContext.getSystemService("wifi");
if (localWifiManager == null) {}
for (WifiInfo localWifiInfo = null;; localWifiInfo = localWifiManager.getConnectionInfo())
{
    String str = null;
    if (localWifiInfo != null) {
        str = localWifiInfo.getMacAddress();
    }
    return str;
}
```

e) connect to the Internet

```java
localURLConnection = localURL.openConnection();
if ((localURLConnection instanceof HttpURLConnection)) {
    ((HttpURLConnection)localURLConnection).setInstanceFollowRedirects(true);
}
localURLConnection.setRequestProperty("Prefer-Html-Meta-Tags", "al");
localURLConnection.connect();
if ((localURLConnection instanceof HttpURLConnection))
{
    HttpURLConnection localHttpURLConnection = (HttpURLConnection)localURLConnection;
    if ((localHttpURLConnection.getResponseCode() >= 300) && (localHttpURLConnection.g
```

f) obtain device information

```java
try
{
    String str = ((TelephonyManager)paramContext.getSystemService("phone")).getDeviceId();
    return str;
}
catch (Exception localException) {}
return "";
```

3)　億次元(com.jidongnet.oku.jdwan)

This app has some malicious activities, such as obtain SIM operator name, obtain SIM serial number, obtain SIM status, obtain geo-location information, obtain WiFi configuration, connect to the Internet, and send out phone information,etc.

a)   obtain SIM operator name

```
str1 = paramTelephonyManager.getSubscriberId();
break;
str2 = paramTelephonyManager.getSimSerialNumber();
break label50;
str3 = paramTelephonyManager.getSimOperator();
break label120;
str4 = paramTelephonyManager.getSimOperatorName();
break label140;
str5 = paramTelephonyManager.getSimCountryIso();
```

b)   obtain SIM serial number

```
if (an.a(paramContext, "android.permission.READ_PHONE_STATE"))
{
  if (a == null) {
    a(paramContext);
  }
  String str = a.getSimSerialNumber();
  return str;
}
```

c)   obtain SIM status

```
if (a == null) {
  a(paramContext);
}
int i = a.getSimState();
boolean bool = false;
if (i == 5) {
  bool = true;
}
return bool;
```

d)   obtain geo-location information

```
Locale localLocale = Locale.US;
Object[] arrayOfObject = new Object[2];
arrayOfObject[0] = Double.valueOf(paramLocation.getLatitude());
arrayOfObject[1] = Double.valueOf(paramLocation.getLongitude());
localBundle.putString("center", String.format(localLocale, "%f,%f", arrayOfObject)
localBundle.putInt("distance", paramInt1);
```

e)   obtain WiFi configuration

```
WifiManager localWifiManager = (WifiManager)paramContext.getSystemService("wifi");
if (!localWifiManager.isWifiEnabled()) {
  continue;
}
WifiInfo localWifiInfo = localWifiManager.getConnectionInfo();
if (localWifiInfo == null) {
  continue;
```

f)   connect to the Internet

```
localURLConnection = localURL.openConnection();
if ((localURLConnection instanceof HttpURLConnection)) {
  ((HttpURLConnection)localURLConnection).setInstanceFollowRedirects(true);
}
localURLConnection.setRequestProperty("Prefer-Html-Meta-Tags", "al");
localURLConnection.connect();
```

g)   send out phone information

```
boolean bool = this.mIntent.getAction().equals("android.intent.action.SEND_MULTIPLE");
if ((i == 0) && (bool))
{
  this.mIntent.setAction("android.intent.action.SEND");
  if ((this.mStreams == null) || (this.mStreams.isEmpty())) {
    break label219;
  }
}
```

4)   Lucky Patcher (pname:catalog.lucky.patcher.root.download.user)

This app has some malicious activities, such as obtain SIM operator name, obtain SIM status, obtain geo-location information, obtain WiFi configuration, obtain device information, and connect to the Internet, etc.

a)   obtain device information

```
TelephonyManager localTelephonyManager = (TelephonyManager)hn.a().c().getSystemService("phone");
if (localTelephonyManager == null) {}
String str;
do
{
  return;
  str = localTelephonyManager.getDeviceId();
} while ((str == null) || (str.trim().length() <= 0));
```

b)   obtain SIM operator name

```
this.mNetworkOperatorName = localTelephonyManager.getNetworkOperatorName();
if (localTelephonyManager.getSimState() == 5) {
  this.mSimOperatorName = localTelephonyManager.getSimOperatorName();
}
this.mUdid = getDeviceIdFromContext(this.mContext);
return;
```

c)   obtain SIM status

```
this.mNetworkOperatorName = localTelephonyManager.getNetworkOperatorName();
if (localTelephonyManager.getSimState() == 5) {
  this.mSimOperatorName = localTelephonyManager.getSimOperatorName();
}
this.mUdid = getDeviceIdFromContext(this.mContext);
```

d)   obtain geo-location information

```
if ((getLocationDecimalDigits() != -1) && (paramLocation != null))
{
  double d = Math.pow(10.0D, getLocationDecimalDigits());
  paramLocation.setLatitude(Math.round(d * paramLocation.getLatitude()) / d);
  paramLocation.setLongitude(Math.round(d * paramLocation.getLongitude()) / d);
}
Settings.getSettings().location = paramLocation;
return;
```

e)   obtain WiFi configuration

```
WifiInfo localWifiInfo = ((WifiManager)paramContext.getSystemService("wifi")).getConnectionInfo();
if (localWifiInfo == null) {
  return null;
}
return localWifiInfo.getMacAddress();
```

f)   connect to the Internet

```
try
{
  localHttpURLConnection = (HttpURLConnection)localURL.openConnection();
  localHttpURLConnection.setConnectTimeout(3000);
  localHttpURLConnection.setReadTimeout(5000);
  localHttpURLConnection.setInstanceFollowRedirects(true);
}
catch (IOException localIOException)
```

5)   GoGoVan (pname:gogovan.GoGoVanClient2)

This app has some malicious activities, such as obtain SIM status, obtain geo-location information, obtain WiFi configuration, obtain network information, and send out phone information,etc.

a)   obtain SIM status

```
    if (k < 0) {
        k = this.ef.jdField_case;
    }
    locala.jdField_case = k;
    }
    this.eq = this.d9.getSimState();
  }
  catch (Exception localException1)
  {
    int j;
```

b)   obtain geo-location information

```
  if (c("android.permission.ACCESS_COARSE_LOCATION"))
  {
    Location localLocation1 = localLocationManager.getLastKnownLocation("network");
    if (localLocation1 != null)
    {
      p.a("get location from network:" + localLocation1.getLatitude() + "," + localLocation1.getLongitude(), new Object[0]);
      return localLocation1;
    }
  }
}
```

c)   obtain WiFi configuration

```
    if (c(paramContext, "android.permission.ACCESS_WIFI_STATE"))
    {
      localWifiInfo = ((WifiManager)paramContext.getSystemService("wifi")).getConnectionInfo();
      String str2 = localWifiInfo.getMacAddress();
      str1 = str2;
    }
  }
```

d)   obtain network information

```
  NetworkInfo localNetworkInfo1 = localConnectivityManager.getNetworkInfo(0);
  NetworkInfo localNetworkInfo2 = localConnectivityManager.getNetworkInfo(1);
  HttpURLConnection localHttpURLConnection;
  if ((localNetworkInfo2 != null) && (localNetworkInfo2.isAvailable()))
  {
    c.a("", "WIFI is available");
    localHttpURLConnection = (HttpURLConnection)localURL.openConnection();
  }
```

e)   send out phone information

```
  public static void sendSMS(String paramString1, String paramString2)
  {
    Intent localIntent = new Intent("android.intent.action.SENDTO", Uri.parse("smsto:" + paramString1));
    localIntent.putExtra("sms_body", paramString2);
    b.a().startActivity(localIntent);
  }
```