主要是根据

https://bbs.pediy.com/thread-266377.htm

https://bbs.pediy.com/thread-272100.htm

这两个文章来分析jd的sign

先抓包获得sign,

unidbg模拟执行 分析version2

unicorn 模拟执行 分析 version 1

京东sign 分析

来源于java的getsignfromjni,先frida hook获取参数

```
签名入参为:  str => uniformRecommend str2=> {"areaCode":0,"curPos":"118.927907,32.114909","dlvAd
.927994,32.114775","eventId":"MyJD_NavigationIcon","filteredPages":0,"newUIStyle":true,"page":1,
e":10,"source":0,"tabIndex":"0","verOld":"2"} str3=> 44cffbd68857d33e str4=> android str5=> 10.0
getSignFromJni sign value is st=1662628635840&sign=7be830c816315c65a5a089370da8be90&sv=112
[Pixel 3::京东 ]->
```

之后unidbg模拟执行调用

```
public class JD extends AbstractJni {


    private static final String SO_PATH =
"C:/Users/24657/Desktop/proj1/jd/libjdbitmapkit.so";
    private static final String APK_PATH =
"C:/Users/24657/Desktop/proj1/jd/jd.apk";

    private static final String INSTALL_ID = "55a9c688729bb118";
    private static final String PLAT_FROM = "android";
    private static final String VERSION = "10.4.6";
    private AndroidEmulator androidEmulator;

    public static void main(String[] args) {

Logger.getLogger("com.github.unidbg.AbstractEmulator").setLevel(Level.DEBUG);

Logger.getLogger("com.github.unidbg.linux.android.dvm.DalvikVM").setLevel(Level.DE
BUG);

Logger.getLogger("com.github.unidbg.linux.android.dvm.BaseVM").setLevel(Level.DEBU
G);

Logger.getLogger("com.github.unidbg.linux.android.dvm").setLevel(Level.DEBUG);
        JD jd = new JD();
        jd.start();
    }
```

```
    public void start() {
        androidEmulator =
AndroidEmulatorBuilder.for32Bit().setProcessName("com.jingdong.android")
                .build();
        Debugger debugger = androidEmulator.attach();

        Memory androidEmulatorMemory = androidEmulator.getMemory();
        androidEmulatorMemory.setLibraryResolver(new AndroidResolver(23));
        VM dalvikVM = androidEmulator.createDalvikVM(new File(APK_PATH));
        DalvikModule module = dalvikVM.loadLibrary(new File(SO_PATH), false);
        dalvikVM.setJni(this);
        Module moduleModule = module.getModule();
        dalvikVM.callJNI_OnLoad(androidEmulator, moduleModule);
        debugger.addBreakPoint(moduleModule, 0x126ac);
        List<Object> params = new ArrayList<>();
        params.add(dalvikVM.getJNIEnv());
        params.add(0);
        DvmClass context = dalvikVM.resolveClass("android/content/Context");
        params.add(dalvikVM.addLocalObject(context.newObject(null)));
        params.add(dalvikVM.addLocalObject(new StringObject(dalvikVM,
"personinfoBusiness")));
        params.add(dalvikVM.addLocalObject(new StringObject(dalvikVM, "
{\"callCJH\":\"1\",\"callNPS\":\"1\",\"closeJX\":\"0\",\"headTaskRefresh\":\"1\",\
"locationArea\":\"0_0_0_0\",\"menuStaticSource\":\"0\",\"menuTimeStamp\":\"1631586
010000\"}")));
        params.add(dalvikVM.addLocalObject(new StringObject(dalvikVM,
INSTALL_ID)));
        params.add(dalvikVM.addLocalObject(new StringObject(dalvikVM,
PLAT_FROM)));
        params.add(dalvikVM.addLocalObject(new StringObject(dalvikVM, VERSION)));
        Number numbers = moduleModule.callFunction(androidEmulator, 0x028B4 + 1,
params.toArray());
        DvmObject<?> object = dalvikVM.getObject(numbers.intValue());
        System.out.println("加密结果为:" + object.getValue());
    }

    @Override
    public DvmObject<?> getStaticObjectField(BaseVM vm, DvmClass dvmClass, String
signature) {
        if ("com/jingdong/common/utils/BitmapkitUtils-
>a:Landroid/app/Application;".equals(signature)) {
            //返回appliation
            return vm.resolveClass("android/app/Activity",
                    vm.resolveClass("android/content/ContextWrapper",

vm.resolveClass("android/content/Context"))).newObject(null);
        }
        return super.getStaticObjectField(vm, dvmClass, signature);
    }

    @Override
    public DvmObject<?> callStaticObjectMethod(BaseVM vm, DvmClass dvmClass,
String signature, VarArg varArg) {
```

```
        if ("com/jingdong/common/utils/BitmapkitZip-
>unZip(Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;)
[B".equals(signature)) {
            byte[] unzip = vm.unzip("META-INF/JINGDONG.RSA");
            System.out.println("unzip " + new String(unzip));
            return new ByteArray(vm, unzip);
        } else if ("com/jingdong/common/utils/BitmapkitZip-
>objectToBytes(Ljava/lang/Object;)[B".equals(signature)) {
            DvmObject<?> objectArg = varArg.getObjectArg(0);
            byte[] bytes = objectToBytes(objectArg.getValue());
            return new ByteArray(vm, bytes);
        }
        return super.callStaticObjectMethod(vm, dvmClass, signature, varArg);
    }

    public static byte[] objectToBytes(Object obj) {
        try {
            ByteArrayOutputStream byteArrayOutputStream = new
ByteArrayOutputStream();
            ObjectOutputStream objectOutputStream = new
ObjectOutputStream(byteArrayOutputStream);
            objectOutputStream.writeObject(obj);
            objectOutputStream.flush();
            byte[] byteArray = byteArrayOutputStream.toByteArray();
            objectOutputStream.close();
            byteArrayOutputStream.close();
            return byteArray;
        } catch (IOException e) {
            return null;
        }
    }

    @Override
    public DvmObject<?> newObject(BaseVM vm, DvmClass dvmClass, String signature,
VarArg varArg) {
        if ("sun/security/pkcs/PKCS7-><init>([B)V".equals(signature)) {
            ByteArray byteArray = varArg.getObjectArg(0);
            try {
                PKCS7 pkcs7 = new PKCS7(byteArray.getValue());
                return
vm.resolveClass("sun/security/pkcs/PKCS7").newObject(pkcs7);
            } catch (ParsingException e) {
                e.printStackTrace();
            }

        }
        return super.newObject(vm, dvmClass, signature, varArg);
    }

    @Override
    public DvmObject<?> callObjectMethod(BaseVM vm, DvmObject<?> dvmObject, String
signature, VarArg varArg) {
        if ("sun/security/pkcs/PKCS7->getCertificates()
[Ljava/security/cert/X509Certificate;".equals(signature)) {
```

```
            PKCS7 pkcs7 = (PKCS7) dvmObject.getValue();
            X509Certificate[] certificates = pkcs7.getCertificates();
            DvmObject<?> object = ProxyDvmObject.createObject(vm, certificates);
            return object;


        }
        return super.callObjectMethod(vm, dvmObject, signature, varArg);
    }

    @Override
    public DvmObject<?> getObjectField(BaseVM vm, DvmObject<?> dvmObject, String
signature) {
        //sourceDir 代表当前apk目录
        if ("android/content/pm/ApplicationInfo-
>sourceDir:Ljava/lang/String;".equals(signature)) {
            StringObject stringObject = new StringObject(vm, APK_PATH);
            return stringObject;
        }
        return super.getObjectField(vm, dvmObject, signature);
    }


    @Override
    public DvmObject<?> callObjectMethodV(BaseVM vm, DvmObject<?> dvmObject,
String signature, VaList vaList) {
        if ("java/lang/StringBuffer-
>append(Ljava/lang/String;)Ljava/lang/StringBuffer;".equals(signature)) {
            StringBuffer stringBuffer = (StringBuffer) dvmObject.getValue();
            DvmObject<?> objectArg = vaList.getObjectArg(0);
            stringBuffer.append(objectArg.getValue().toString());
            return
vm.resolveClass("java/lang/StringBuffer").newObject(stringBuffer);
        } else if ("java/lang/Integer-
>toString()Ljava/lang/String;".equals(signature)) {
            Integer integer = (Integer) dvmObject.getValue();
            return new StringObject(vm, integer.toString());
        }else if("java/lang/StringBuffer-
>toString()Ljava/lang/String;".equals(signature)){
            StringBuffer stringBuffer = (StringBuffer) dvmObject.getValue();
            return new StringObject(vm, stringBuffer.toString());
        }
        return super.callObjectMethodV(vm, dvmObject, signature, vaList);
    }

    @Override
    public DvmObject<?> newObjectV(BaseVM vm, DvmClass dvmClass, String signature,
VaList vaList) {
        if ("java/lang/StringBuffer-><init>()V".equals(signature)) {
            StringBuffer stringBuffer = new StringBuffer();
            return
vm.resolveClass("java/lang/StringBuffer").newObject(stringBuffer);
        } else if ("java/lang/Integer-><init>(I)V".equals(signature)) {
            int intArg = vaList.getIntArg(0);
```

```
            Integer integer = Integer.valueOf(intArg);
            return vm.resolveClass("java/lang/Integer").newObject(integer);
        }
        return super.newObjectV(vm, dvmClass, signature, vaList);
    }
}
```

st是时间戳

```
v16 = v15;
if ( v15 )
{
  *((_BYTE *)v15 + arrge_len) = 0;
  ((void (__fastcall *)(JNIEnv_ *, int))a1->functions->GetByteArrayRegion)(a1, arrgr);
  switch ( v12 )
  {
    case 1:
      v23 = a4_1;
      if ( a4_1 )
        v23 = 1;
      sub_10E18(v13, v23, v16, arrge_len);
      break;
    case 2:
      v22 = a4_1;
      if ( a4_1 )
        v22 = 1;
      sub_10DE4(v13, v22, v16, arrge_len);
      break;
    case 0:
      v17 = a4_1;
      if ( a4_1 )
        v17 = 1;
      sub_10E4C(v13, v17, v16, arrge_len);
      break;
```

接着看sign的生成 sub_126ac 根据random_46 % 3随机在3个模式中选一个 看了模式2 进入模式2的函数 r0为

```
0000: 38 30 33 30 36 66 34 33 37 30 62 33 39 66 64 35    80306f4370b39fd5
0010: 36 33 30 61 64 30 35 32 39 66 37 37 61 64 62 36    630ad0529f77adb6
0020: 00 00 00 00 00 00 00 00 E9 A8 CB DB A0 12 FE FF    ................
0030: A9 4C 8B 31 01 00 00 00 01 00 00 00 C3 E9 E6 32    .L.1...........2
0040: 54 0C F7 09 CE 03 DC 61 51 2E 48 29 73 2B 01 40    T......aQ.H)s+.@
0050: 01 00 00 00 01 00 00 00 BF F5 FF BF 0D 36 F8 50    .............6.P
0060: 19 2C A1 6E C4 7F 68 FA A2 B0 21 79 D9 E7 D2 66    .,.n..h...!y...f
^                                                                              ^
```

r1 为1 r2为 我的那个json参数的拼接字符串

r3 0x106长度

```
size: 112
0000: 66 75 6E 63 74 69 6F 6E 49 64 3D 70 65 72 73 6F    functionId=perso
0010: 6E 69 6E 66 6F 42 75 73 69 6E 65 73 73 26 62 6F    ninfoBusiness&bo
0020: 64 79 3D 7B 22 63 61 6C 6C 43 4A 48 22 3A 22 31    dy={"callCJH":"1
0030: 22 2C 22 63 61 6C 6C 4E 50 53 22 3A 22 31 22 2C    ","callNPS":"1",
0040: 22 63 6C 6F 73 65 4A 58 22 3A 22 30 22 2C 22 68    "closeJX":"0","h
0050: 65 61 64 54 61 73 6B 52 65 66 72 65 73 68 22 3A    eadTaskRefresh":
0060: 22 31 22 2C 22 6C 6F 63 61 74 69 6F 6E 41 72 65    "1","locationAre
^-------------------------------------------------------------------------^
```

经过查看，前面是固定生成 unidbg调试得到结果

后面就只是xor了