[原创]拼夕夕anti-token分析

那年没下雪 7 1

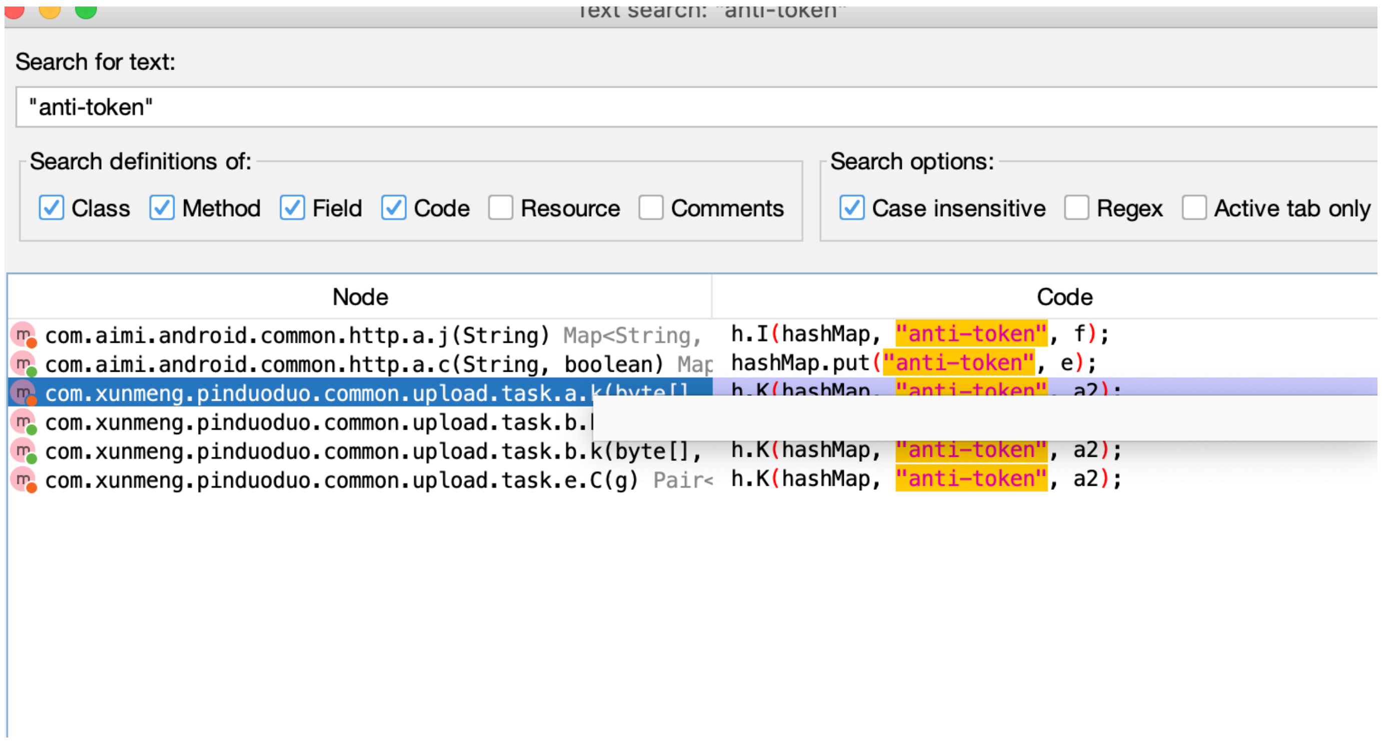
大侠

举报

1天前 1023

前言：拼夕夕charles抓包分析发现跟商品相关的请求头里都带了一个anti-token的字段且每次都不一样,那么下面的操作就从分析anti-token开始了

1.jadx反编译直接搜索



Text search: "anti-token"

Search for text:

"anti-token"

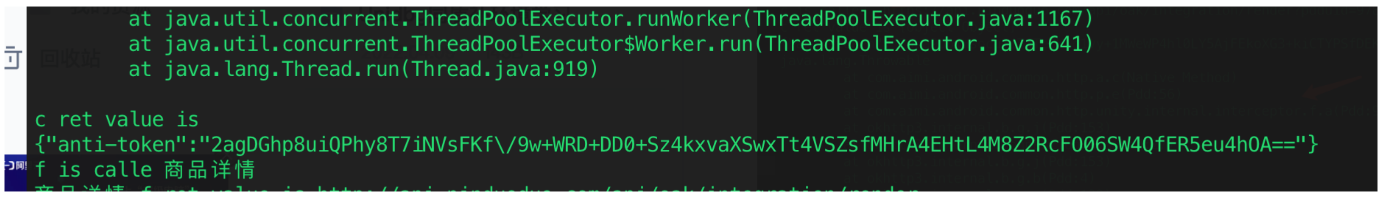
Search definitions of: ☑ Class ☑ Method ☑ Field ☑ Code ☐ Resource ☐ Comments

Search options: ☑ Case insensitive ☐ Regex ☐ Active tab only

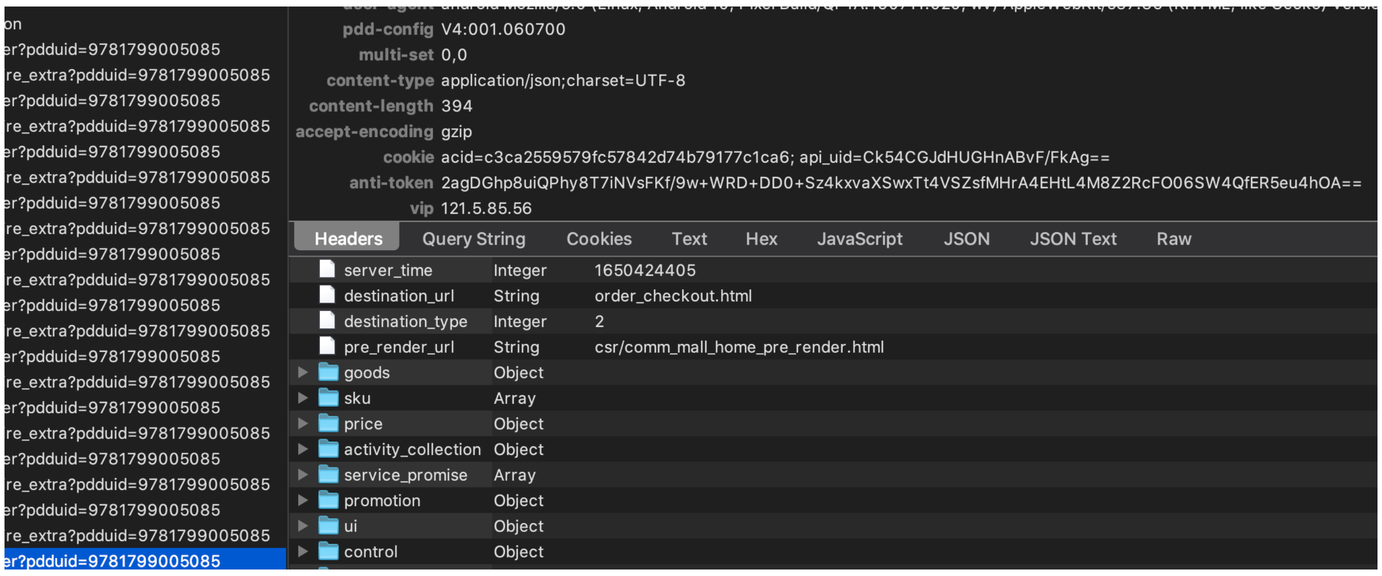
Node	Code
com.aimi.android.common.http.a.j(String) Map<String,	h.I(hashMap, "anti-token", f);
com.aimi.android.common.http.a.c(String, boolean) Map	hashMap.put("anti-token", e);
com.xunmeng.pinduoduo.common.upload.task.a.k(byte[]	h.K(hashMap, "anti-token", a2);
com.xunmeng.pinduoduo.common.upload.task.b.l	
com.xunmeng.pinduoduo.common.upload.task.b.k(byte[],	h.K(hashMap, "anti-token", a2);
com.xunmeng.pinduoduo.common.upload.task.e.C(g) Pair<	h.K(hashMap, "anti-token", a2);

选中跟http相关的类对这个方法进行打印堆栈

```
c ret value is
c is called  https://api.pinduoduo.com/api/oak/integration/render?pdduid=9781799005085 fals
java.lang.Throwable
        at com.aimi.android.common.http.a.c(Native Method)
        at com.aimi.android.common.http.p.e(Pdd:56)
        at com.aimi.android.common.http.unity.internal.interceptor.f.a(Pdd:94)
        at okhttp3.internal.b.g.j(Pdd:153)
        at okhttp3.internal.connection.a.a(Pdd:27)
        at okhttp3.internal.b.g.j(Pdd:153)
```



```
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1167)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:641)
        at java.lang.Thread.run(Thread.java:919)

c ret value is
{"anti-token":"2agDGhp8uiQPhy8T7iNVsFKf\/9w+WRD+DD0+Sz4kxvaXSwxTt4VSZsfMHrA4EHtL4M8Z2RcFO06SW4QfER5eu4hOA=="}
f is calle 商品详情
```



```
on
er?pdduid=9781799005085
re_extra?pdduid=9781799005085
er?pdduid=9781799005085
re_extra?pdduid=9781799005085
er?pdduid=9781799005085
re_extra?pdduid=9781799005085
er?pdduid=9781799005085
re_extra?pdduid=9781799005085
er?pdduid=9781799005085
re_extra?pdduid=9781799005085
er?pdduid=9781799005085
re_extra?pdduid=9781799005085
er?pdduid=9781799005085
re_extra?pdduid=9781799005085
er?pdduid=9781799005085
re_extra?pdduid=9781799005085
er?pdduid=9781799005085
re_extra?pdduid=9781799005085
er?pdduid=9781799005085
re_extra?pdduid=9781799005085
er?pdduid=9781799005085
```

```
      pdd-config V4:001.060700
       multi-set 0,0
    content-type application/json;charset=UTF-8
  content-length 394
 accept-encoding gzip
          cookie acid=c3ca2559579fc57842d74b79177c1ca6; api_uid=Ck54CGJdHUGHnABvF/FkAg==
      anti-token 2agDGhp8uiQPhy8T7iNVsFKf/9w+WRD+DD0+Sz4kxvaXSwxTt4VSZsfMHrA4EHtL4M8Z2RcFO06SW4QfER5eu4hOA==
             vip 121.5.85.56
```

server_time	Integer	1650424405
destination_url	String	order_checkout.html
destination_type	Integer	2
pre_render_url	String	csr/comm_mall_home_pre_render.html
goods	Object	
sku	Array	
price	Object	
activity_collection	Object	
service_promise	Array	
promotion	Object	
ui	Object	
control	Object	

结合堆栈方法调用的情况找到具体anti-token是由拦截器类f.a方法调用的,在http.a.c()方法中生成并且http.p.e()方法中加入请求头

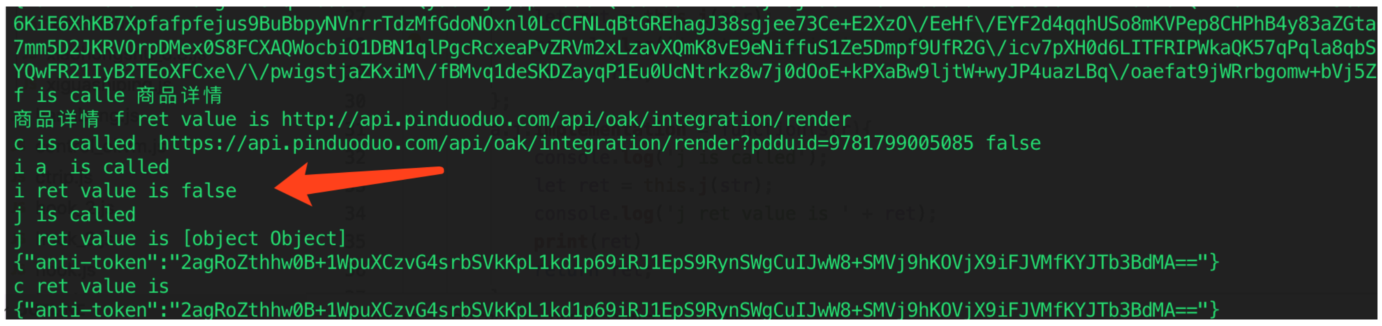
在http.a.c()方法中有个一个判断条件如果为true则走d.a().e()方法生成anti-token

```
        path = e.a(path, 1);
    }
    if (z || h(path)) {
        com.aimi.android.common.cmt.a.a().ag(91023, 2, true);
        Context c = com.xunmeng.pinduoduo.basekit.a.c();
        try {
            Logger.d("Pdd.AntiToken", "before get deviceInfo2:%s", path);
            long longValue = TimeStamp.getRealLocalTime().longValue();
            if (AbTest.instance().isFlowControl("ab_anti_token_use_timestamp_v2_5150", false) && f.b()) {
                long realLocalTimeV2 = TimeStamp.getRealLocalTimeV2();
                Logger.i("Pdd.AntiToken", "tsV2:%d, realTimestamp:%d, clientTime:%d", Long.valueOf(realLoca
                longValue = realLocalTimeV2;
            }
            e = d.a().e(c, Long.valueOf(longValue));
            Object[] objArr = new Object[2];
            if (e == null) {
                i = 0;
            } else {
                i = e.length();
            }
            objArr[0] = Integer.valueOf(i);
            objArr[1] = path;
            Logger.d("Pdd.AntiToken", "after get deviceInfo2, len:%s, api:%s", objArr);
        } catch (Throwable th) {
            Logger.e("Pdd.AntiToken", "obtainToken is null, url:%s, error:%s", str, th);
            k(str, th);
        }
        if (!TextUtils.isEmpty(e)) {
            HashMap hashMap = new HashMap();
```

如果为false则走j()方法生成anti-token

```
                longValue = realLocalTimeV2;
            }
            e = d.a().e(c, Long.valueOf(longValue));
            Object[] objArr = new Object[2];
            if (e == null) {
                i = 0;
            } else {
                i = e.length();
            }
            objArr[0] = Integer.valueOf(i);
            objArr[1] = path;
            Logger.d("Pdd.AntiToken", "after get deviceInfo2, len:%s, api:%s",
        } catch (Throwable th) {
            Logger.e("Pdd.AntiToken", "obtainToken is null, url:%s, error:%s",
            k(str, th);
        }
        if (!TextUtils.isEmpty(e)) {
            HashMap hashMap = new HashMap();
            hashMap.put("anti-token", e);
            Logger.i("Pdd.AntiToken", "obtainToken added:%s, len:%s", path, In
            Logger.v("Pdd.AntiToken", "token:%s", e);
            return hashMap;
        }
        Logger.e("Pdd.AntiToken", "obtainToken is null, url:%s, ", str);
        k(str, null);
        return null;
    } else if (!i(path)) {
        return j(path);
    } else {
        return null;
```

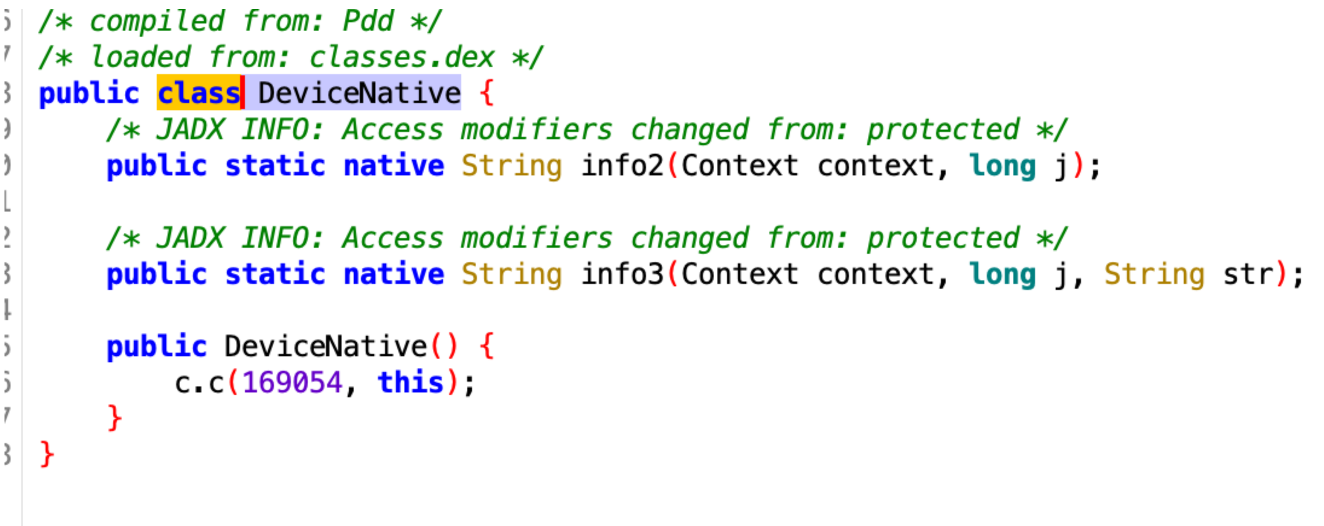
hook这个i()方法返回值可知获取商品详情接口返回值为false所以走的是j()方法进行计算anti-token。



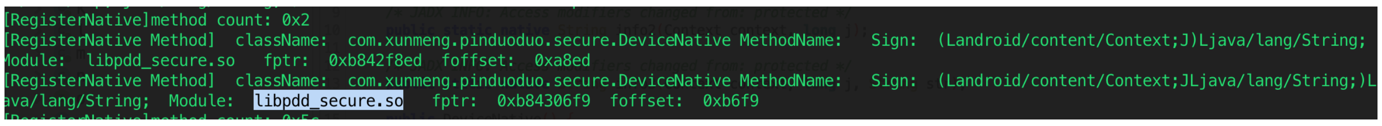
```
6KiE6XhKB7Xpfafpfejus9BuBbpyNVnrrTdzMfGdoNOxnl0LcCFNLqBtGREhagJ38sgjee73Ce+E2XzO\/EeHf\/EYF2d4qqhUSo8mKVPep8CHPhB4y83aZGta
7mm5D2JKRVOrpDMex0S8FCXAQWocbiO1DBN1qlPgcRcxeaPvZRVm2xLzavXQmK8vE9eNiffuS1Ze5Dmpf9UfR2G\/icv7pXH0d6LITFRIPWkaQK57qPqla8qbS
YQwFR21IyB2TEoXFCxe\/\/pwigstjaZKxiM\/fBMvq1deSKDZayqP1Eu0UcNtrkz8w7j0dOoE+kPXaBw9ljtW+wyJP4uazLBq\/oaefat9jWRrbgomw+bVj5Z
f is calle 商品详情
商品详情 f ret value is http://api.pinduoduo.com/api/oak/integration/render
c is called  https://api.pinduoduo.com/api/oak/integration/render?pdduid=9781799005085 false
i a  is called
i ret value is false
j is called
j ret value is [object Object]
{"anti-token":"2agRoZthhw0B+1WpuXCzvG4srbSVkKpL1kd1p69iRJ1EpS9RynSWgCuIJwW8+SMVj9hKOVjX9iFJVMfKYJTb3BdMA=="}
c ret value is
{"anti-token":"2agRoZthhw0B+1WpuXCzvG4srbSVkKpL1kd1p69iRJ1EpS9RynSWgCuIJwW8+SMVj9hKOVjX9iFJVMfKYJTb3BdMA=="}
```

SecureNative.deviceInfo3()方法生成,传入的str为pdd生成的固定id 一个字符串.

```
    @Override // com.aimi.android.common.service.c
    public String f(Context context, String str) {
        long j;
        if (com.xunmeng.manwe.hotfix.c.p(169369, this, context, str)) {
            return com.xunmeng.manwe.hotfix.c.w();
        }
        if (AbTest.instance().isFlowControl("ab_timestamp_v2_5590", true)) {
            j = TimeStamp.getRealLocalTimeV2();
        } else {
            j = k.c(TimeStamp.getRealLocalTime());
        }
        try {
            return SecureNative.deviceInfo3(context, Long.valueOf(j), str);
        } catch (Throwable th) {
            Logger.e("PDD.SecureServiceImpl", "deviceInfo3 error:" + th);
            return null;
        }
    }
```

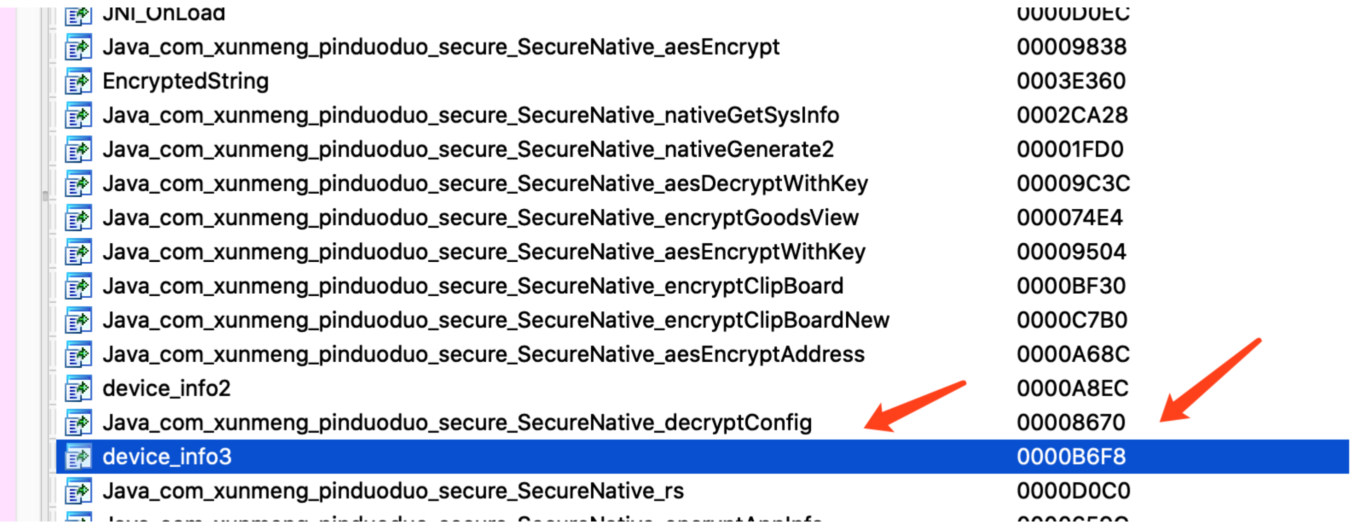


```
/* compiled from: Pdd */
/* loaded from: classes.dex */
public class DeviceNative {
    /* JADX INFO: Access modifiers changed from: protected */
    public static native String info2(Context context, long j);

    /* JADX INFO: Access modifiers changed from: protected */
    public static native String info3(Context context, long j, String str);

    public DeviceNative() {
        c.c(169054, this);
    }
}
```



```
[RegisterNative]method count: 0x2
[RegisterNative Method]  className:  com.xunmeng.pinduoduo.secure.DeviceNative MethodName:   Sign:  (Landroid/content/Context;J)Ljava/lang/String;
Module:  libpdd_secure.so   fptr:  0xb842f8ed  foffset:  0xa8ed
[RegisterNative Method]  className:  com.xunmeng.pinduoduo.secure.DeviceNative MethodName:   Sign:  (Landroid/content/Context;JLjava/lang/String;)L
ava/lang/String;  Module:  libpdd_secure.so   fptr:  0xb84306f9  foffset:  0xb6f9
```

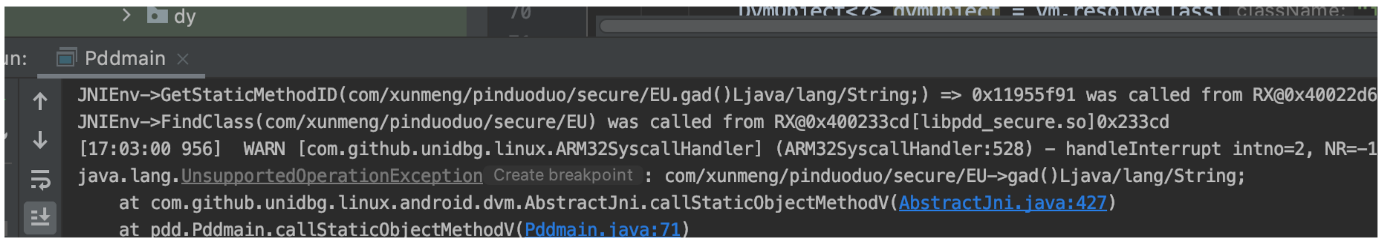
根据hook_libart 得到info3()方法是在libodd_secure.so中,那么ida打开看看这个so包



```
JNI_OnLoad                                                  0000D0EC
Java_com_xunmeng_pinduoduo_secure_SecureNative_aesEncrypt           00009838
EncryptedString                                             0003E360
Java_com_xunmeng_pinduoduo_secure_SecureNative_nativeGetSysInfo     0002CA28
Java_com_xunmeng_pinduoduo_secure_SecureNative_nativeGenerate2      00001FD0
Java_com_xunmeng_pinduoduo_secure_SecureNative_aesDecryptWithKey    00009C3C
Java_com_xunmeng_pinduoduo_secure_SecureNative_encryptGoodsView     000074E4
Java_com_xunmeng_pinduoduo_secure_SecureNative_aesEncryptWithKey    00009504
Java_com_xunmeng_pinduoduo_secure_SecureNative_encryptClipBoard     0000BF30
Java_com_xunmeng_pinduoduo_secure_SecureNative_encryptClipBoardNew  0000C7B0
Java_com_xunmeng_pinduoduo_secure_SecureNative_aesEncryptAddress    0000A68C
device_info2                                                0000A8EC
Java_com_xunmeng_pinduoduo_secure_SecureNative_decryptConfig        00008670
device_info3                                                0000B6F8
Java_com_xunmeng_pinduoduo_secure_SecureNative_rs                   0000D0C0
```

2.这部分我们采用unidbg+jnitrace+frida相结合的方式

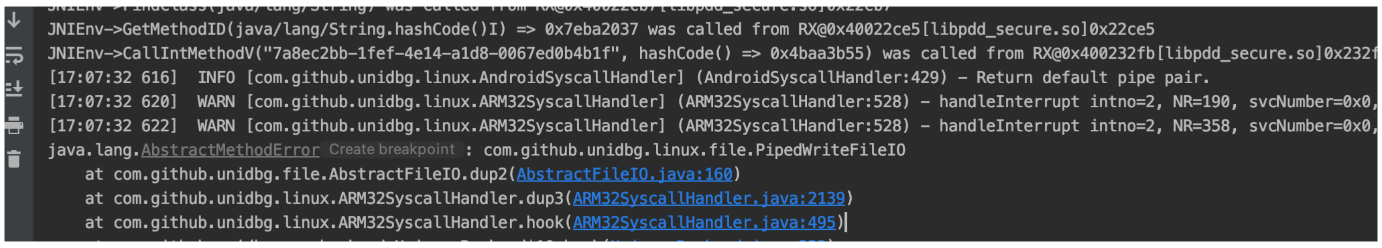
unidbg前期准备的代码这里就不发了直接调用这个info3方法



```
JNIEnv->GetStaticMethodID(com/xunmeng/pinduoduo/secure/EU.gad()Ljava/lang/String;) => 0x11955f91 was called from RX@0x40022d6
JNIEnv->FindClass(com/xunmeng/pinduoduo/secure/EU) was called from RX@0x400233cd[libpdd_secure.so]0x233cd
[17:03:00 956]  WARN [com.github.unidbg.linux.ARM32SyscallHandler] (ARM32SyscallHandler:528) - handleInterrupt intno=2, NR=-1
java.lang.UnsupportedOperationException Create breakpoint : com/xunmeng/pinduoduo/secure/EU->gad()Ljava/lang/String;
    at com.github.unidbg.linux.android.dvm.AbstractJni.callStaticObjectMethodV(AbstractJni.java:427)
    at pdd.Pddmain.callStaticObjectMethodV(Pddmain.java:71)
```

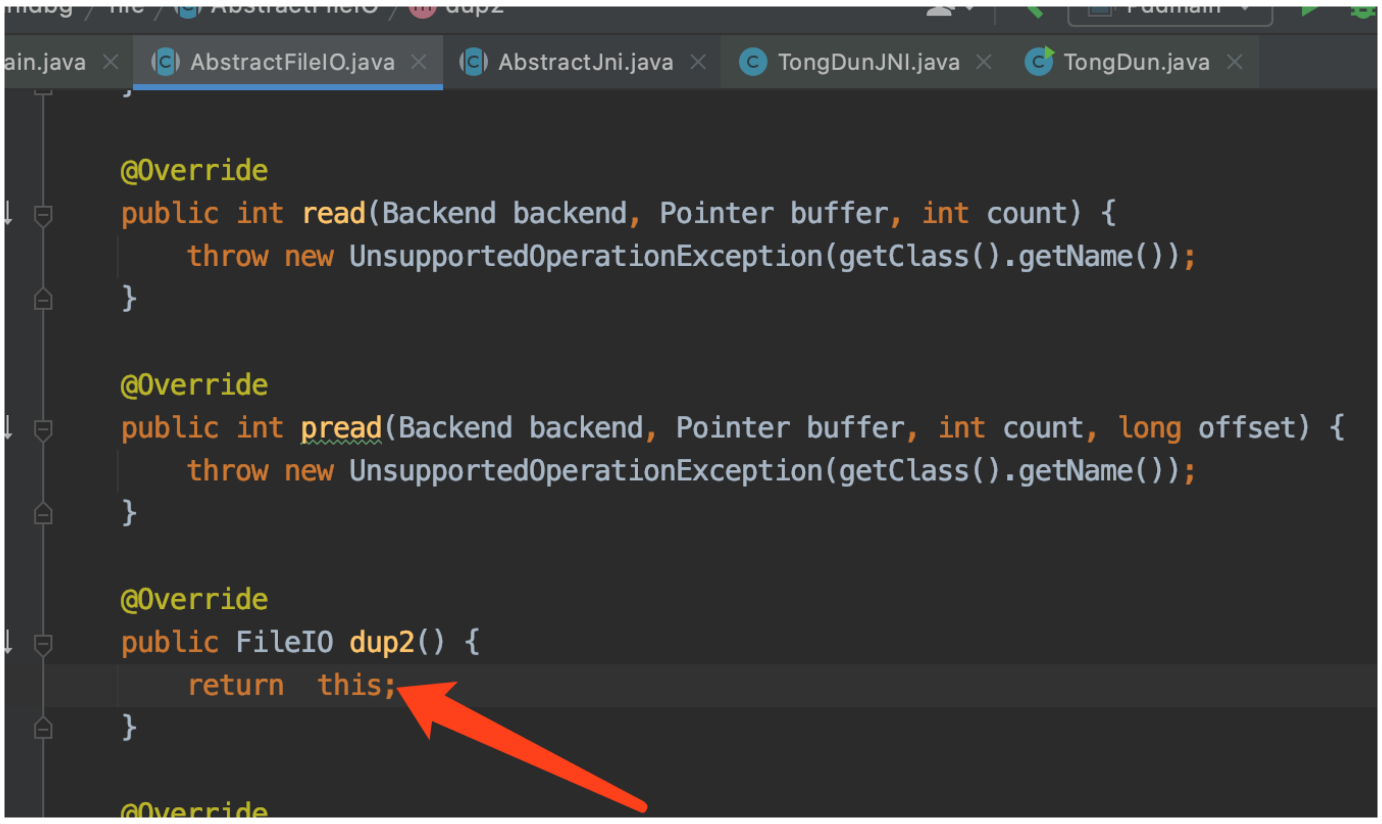
这里提示调用gad()方法返回一个字符串那么frida hook这个方法拿到这个值 如下图 一个固定的字符串16位长度看着像AES的密钥

```
i ret value is false
f is called dIrjGpkC
gad is called
gad ret value is cb14a9e76b72a627
f ret value is 2agcQ3ZrcFGepIPCUBHiHDk4SNnI40fS5zJUzQMGYx80/twDBcJnJebaCErqtsvQSPE3IxHh0OCnPo7Du8wDDrZ2Q==
```



```
JNIEnv->GetMethodID(java/lang/String.hashCode()I) => 0x7eba2037 was called from RX@0x40022ce5[libpdd_secure.so]0x22ce5
JNIEnv->CallIntMethodV("7a8ec2bb-1fef-4e14-a1d8-0067ed0b4b1f", hashCode() => 0x4baa3b55) was called from RX@0x400232fb[libpdd_secure.so]0x232f
[17:07:32 616]  INFO [com.github.unidbg.linux.AndroidSyscallHandler] (AndroidSyscallHandler:429) - Return default pipe pair.
[17:07:32 620]  WARN [com.github.unidbg.linux.ARM32SyscallHandler] (ARM32SyscallHandler:528) - handleInterrupt intno=2, NR=190, svcNumber=0x0,
[17:07:32 622]  WARN [com.github.unidbg.linux.ARM32SyscallHandler] (ARM32SyscallHandler:528) - handleInterrupt intno=2, NR=358, svcNumber=0x0,
java.lang.AbstractMethodError Create breakpoint : com.github.unidbg.linux.file.PipedWriteFileIO
    at com.github.unidbg.file.AbstractFileIO.dup2(AbstractFileIO.java:160)
    at com.github.unidbg.linux.ARM32SyscallHandler.dup3(ARM32SyscallHandler.java:2139)
    at com.github.unidbg.linux.ARM32SyscallHandler.hook(ARM32SyscallHandler.java:495)
```

补完简单的环境代码后,再次运行报这个错误 看错误应该是缺少文件 ,那么看看日志需要补那个文件继续运行,没有返回值报空指针。execve()函数执行的时候程序exit了这里我们返回对象本身.



```java
    @Override
    public int read(Backend backend, Pointer buffer, int count) {
        throw new UnsupportedOperationException(getClass().getName());
    }

    @Override
    public int pread(Backend backend, Pointer buffer, int count, long offset) {
        throw new UnsupportedOperationException(getClass().getName());
    }

    @Override
    public FileIO dup2() {
        return  this;
    }

    @Override
```

execve()函数执行的时候程序exit了

```
[17:21:08 639]  WARN [com.github.unidbg.linux.ARM32SyscallHandler] (ARM32SyscallHandler:528) - handleInterrupt intno=2, NR=190, svcNumber=0x0, PC=RX@0x40128b5c[
[17:21:08 641]  INFO [com.github.unidbg.linux.ARM32SyscallHandler] (ARM32SyscallHandler:1036) - execve filename=/system/bin/sh, args=[sh, -c, cat /proc/sys/kern
exit with code: 127
Exception in thread "main" java.lang.NullPointerException Create breakpoint
    at pdd.Pddmain.callInfo3(Pddmain.java:67)
    at pdd.Pddmain.create(Pddmain.java:47)
    at pdd.Pddmain.main(Pddmain.java:31)
```

execve filename=/system/bin/sh, args=[sh, -c, cat /proc/sys/kernel/random/boot_id]
这个函数相当于查看 boot_id这个文件信息

```
sailfish:/system/bin # cat /proc/sys/kernel/random/boot_id
05df5c42-7d0a-404a-bb38-63fe5c70dee4
sailfish:/system/bin #
```

捋顺下逻辑应该就是先fork进程 然后在子进程中读取这个文件 然后把他写入pip中
那么自定义syscallhandler后 再次运行成功拿到结果

```
JNIEnv->CallObjectMethodV(java.util.UUID@567d299b, toString() => "79868935-4d6f-4ec2-a714-85aa49459ae1") was called from RX
JNIEnv->NewStringUTF("-") was called from RX@0x4001614b[libpdd_secure.so]0x1614b
JNIEnv->NewStringUTF("") was called from RX@0x4001615b[libpdd_secure.so]0x1615b
JNIEnv->FindClass(java/lang/String) was called from RX@0x40022cb7[libpdd_secure.so]0x22cb7
JNIEnv->GetMethodID(java/lang/String.replaceAll(Ljava/lang/String;Ljava/lang/String;)Ljava/lang/String;) => 0x7e257efa was
JNIEnv->CallObjectMethodV("79868935-4d6f-4ec2-a714-85aa49459ae1", replaceAll("-", "") => "79868935-4d6f-4ec2-a714-85aa49459
JNIEnv->GetStringUtfChars("79868935-4d6f-4ec2-a714-85aa49459ae1") was called from RX@0x4001620b[libpdd_secure.so]0x1620b
JNIEnv->ReleaseStringUTFChars("79868935-4d6f-4ec2-a714-85aa49459ae1") was called from RX@0x40016233[libpdd_secure.so]0x1623
JNIEnv->NewStringUTF("79868935-4d6f-4ec2-a714-85aa49459ae1") was called from RX@0x4000b7d5[libpdd_secure.so]0xb7d5
JNIEnv->FindClass(java/lang/String) was called from RX@0x40022cb7[libpdd_secure.so]0x22cb7
JNIEnv->GetMethodID(java/lang/String.hashCode()I) => 0x7eba2037 was called from RX@0x40022ce5[libpdd_secure.so]0x22ce5
JNIEnv->CallIntMethodV("79868935-4d6f-4ec2-a714-85aa49459ae1", hashCode() => 0x10c716a1) was called from RX@0x400232fb[libp
pipe2 pipefd=unidbg@0xbffff460, flags=0x0, read=5, write=4, stdout=905df5c42-7d0a-404a-bb38-63fe5c70dee4

vfork pid=2285
[17:50:18 097]  WARN [com.github.unidbg.linux.ARM32SyscallHandler] (ARM32SyscallHandler:528) - handleInterrupt intno=2, NR=
[17:50:18 106]  WARN [com.github.unidbg.linux.ARM32SyscallHandler] (ARM32SyscallHandler:528) - handleInterrupt intno=2, NR=
JNIEnv->NewStringUTF("2agmZXgMRF110DQqK7zX4fV6VBBjOOCDI7nh9drkFfA0eI1TGAZCc9HaOmrju1K5TDe") was called from RX@0x4000b6b1[l
2agmZXgMRF110DQqK7zX4fV6VBBjOOCDI7nh9drkFfA0eI1TGAZCc9HaOmrju1K5TDe
```

全部代码如下:

```java
package pdd;

import com.github.unidbg.AndroidEmulator;
import com.github.unidbg.Emulator;
```

```
import com.github.unidbg.file.IOResolver;
import com.github.unidbg.file.linux.AndroidFileIO;
import com.github.unidbg.linux.android.AndroidARMEmulator;
import com.github.unidbg.linux.android.AndroidEmulatorBuilder;
import com.github.unidbg.linux.android.AndroidResolver;
import com.github.unidbg.linux.android.dvm.*;
import com.github.unidbg.linux.file.ByteArrayFileIO;
import com.github.unidbg.memory.Memory;
import com.github.unidbg.memory.SvcMemory;
import com.github.unidbg.spi.SyscallHandler;
import com.github.unidbg.unix.UnixSyscallHandler;

import java.io.File;
import java.nio.charset.StandardCharsets;
import java.util.ArrayList;
import java.util.List;
import java.util.UUID;

public class Pddmain extends AbstractJni implements IOResolver<AndroidFileIO> {

    private AndroidEmulator androidEmulator;
    private static final String APK_PATH = "/Users/huangchao/Downloads/com.xunmeng.pinduc
    private static final String SO_PATH = "/Users/huangchao/Downloads/com.xunmeng.pinduoc
v7a/libpdd_secure.so";
    private Module moduleModule;
    private VM dalvikVM;

    public static void main(String[] args) {
        Pddmain main = new Pddmain();
        main.create();
    }

    private void create() {
        AndroidEmulatorBuilder androidEmulatorBuilder = new AndroidEmulatorBuilder(false)
            @Override
            public AndroidEmulator build() {
                return new AndroidARMEmulator("com.xunmeng.pinduoduo",rootDir,backendFact
                    @Override
                    protected UnixSyscallHandler<AndroidFileIO> createSyscallHandler(SvcM
                        return new PddArmSysCallHand(svcMemory);
                    }
                };
            }
        };
        androidEmulator = androidEmulatorBuilder.setProcessName("").build();
        androidEmulator.getSyscallHandler().addIOResolver(this);
        Memory androidEmulatorMemory = androidEmulator.getMemory();
        androidEmulatorMemory.setLibraryResolver(new AndroidResolver(23));
        dalvikVM = androidEmulator.createDalvikVM(new File(APK_PATH));
        DalvikModule module = dalvikVM.loadLibrary(new File(SO_PATH), true);
        moduleModule = module.getModule();
        dalvikVM.setJni(this);
        dalvikVM.setVerbose(true);
        dalvikVM.callJNI_OnLoad(androidEmulator, moduleModule);
        callInfo3();
    }

    @Override
    public void callStaticVoidMethodV(BaseVM vm, DvmClass dvmClass, String signature, VaI
        if ("com/tencent/mars/xlog/PLog->i(Ljava/lang/String;Ljava/lang/String;)V".equals
            return;
        }
        super.callStaticVoidMethodV(vm, dvmClass, signature, vaList);
    }

    private void callInfo3() {
        List<Object> argList = new ArrayList<>();
        argList.add(dalvikVM.getJNIEnv());
        argList.add(0);
        DvmObject<?> context = dalvikVM.resolveClass("android/content/Context").newObject
        argList.add(dalvikVM.addLocalObject(context));
        argList.add(dalvikVM.addLocalObject(new StringObject(dalvikVM, "api/oak/integrati
        argList.add(dalvikVM.addLocalObject(new StringObject(dalvikVM, "dIrjGpkC")));
        Number number = moduleModule.callFunction(androidEmulator, 0xb6f9, argList.toArra
        String toString = dalvikVM.getObject(number.intValue()).getValue().toString();
        System.out.println(toString);
    }

    @Override
    public DvmObject<?> callStaticObjectMethodV(BaseVM vm, DvmClass dvmClass, String sigr
        if ("com/xunmeng/pinduoduo/secure/EU->gad()Ljava/lang/String;".equals(signature))
            return new StringObject(vm, "cb14a9e76b72a627");
        } else if ("java/util/UUID->randomUUID()Ljava/util/UUID;".equals(signature)) {
            UUID uuid = UUID.randomUUID();
            DvmObject<?> dvmObject = vm.resolveClass("java/util/UUID").newObject(uuid);
            return dvmObject;
        }
```

```
    @Override
    public DvmObject<?> callObjectMethodV(BaseVM vm, DvmObject<?> dvmObject, String signa
        if ("java/util/UUID->toString()Ljava/lang/String;".equals(signature)) {
            UUID uuid = (UUID) dvmObject.getValue();
            return new StringObject(vm, uuid.toString());
        } else if ("java/lang/String->replaceAll(Ljava/lang/String;Ljava/lang/String;)Lja
            String obj = dvmObject.getValue().toString();
            String arg0 = vaList.getObjectArg(0).toString();
            String arg1 = vaList.getObjectArg(1).toString();
            String replaceAll = obj.replaceAll(arg0, arg1);
            return new StringObject(vm, replaceAll);

        }
        return super.callObjectMethodV(vm, dvmObject, signature, vaList);
    }

    @Override
    public int callIntMethodV(BaseVM vm, DvmObject<?> dvmObject, String signature, VaList
        if ("java/lang/String->hashCode()I".equals(signature)) {
            return dvmObject.getValue().toString().hashCode();
        }
        return super.callIntMethodV(vm, dvmObject, signature, vaList);
    }

    @Override
    public FileResult<AndroidFileIO> resolve(Emulator<AndroidFileIO> emulator, String pat
        if ("/proc/stat".equals(pathname)) {
            String info = "cpu  15884810 499865 12934024 24971554 59427 3231204 945931 0
                     "cpu0 6702550 170428 5497985 19277857 45380 1821584 529454 0 0 0\n" +
                     "cpu1 4438333 121907 3285784 1799772 3702 504395 255852 0 0 0\n" +
                     "cpu2 2735453 133666 2450712 1812564 4626 538114 93763 0 0 0\n" +
                     "cpu3 2008473 73862 1699542 2081360 5716 367109 66860 0 0 0\n" +
                     "intr 1022419954 0 0 0 159719900 0 16265892 4846825 5 5 5 6 0 0 497 2
98 0 0 0 0 0 0 3212852 0 12195284 0 0 0 0 0 43 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 12513 2743129 375 12477726 0 0 0 0 37 1351794 0 36 8 0 0 0 0 0 0 584
18 0 18 0 0 0 0 0 0 66 0 0 0 0 0 0 0 77 0 166 0 0 0 0 0 394 0 0 0 0 0 1339137 0 0 0 0 0
 47 0 0 0 2 2 0 0 0 6 8 0 0 0 2 0 462 2952327 35420 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 495589 0 0 0 0 3 27 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 4760 0 0 97 0 0 0 0 0 0 0 0 0 243 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 22355451 0 0 0 14 0 24449357 96 49415 2 0 0 0
222 3211 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 649346 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                     "ctxt 1572087931\n" +
                     "btime 1649910663\n" +
                     "processes 230673\n" +
                     "procs_running 6\n" +
                     "procs_blocked 0\n" +
                     "softirq 374327567 12481657 139161248 204829 7276312 2275183 26796 12
            return FileResult.success(new ByteArrayFileIO(oflags, pathname, info.getBytes
        }
        return null;
    }
}
```

[【公告】看雪招聘大学实习生！看雪20年安全圈的口碑，助你快速成长！]

最后于 1天前 被那年没下雪编辑 ，原因：

收藏 · 1　　点赞　　打赏　　分享

最新回复 (0)

sun-shine

内容

回帖　　表情　　高级回复

返回

1

首页 论坛 课程 招聘 发现