

# hook从未如此丝滑

---

原创 海岸线突击队 酒仙桥六号部队

2020-12-29原文

这是 酒仙桥六号部队 的第 **138** 篇文章。

全文共计**2143**个字，预计阅读时长**7**分钟。

## 朋友们好啊

我是葫芦兄弟掌门人铁头娃！

刚才有个朋友问我：“铁老师，发生甚么事了？”

我说怎么回事？

给我发了几张截图，我一看！

奥！原来是昨天有个app，54多兆。

不仅有壳抓包还加密。

塔门说：

有一个说是我对这个app做测试，脱壳脱的头发都没了。

铁老师你能不能教教我简单的测法？

帮我缓解下我的工作量。

我说可以~

我说你一个一个个脱壳练死劲不好用，他不服气。

我说小朋友，你用脱壳来比我hook。

他说比不过，

他说你这个没用，

我说我这个有用。

这是化劲，app测试是讲化劲的，四两拨千斤。

他非要和我试试！

## 我啪一下就站起来，上来就先搭建环境

本次测试采用模拟器环境。

frida环境搭建略过，\*度资料很多。

使用的是mumu模拟器（adb连接比\*神模拟器方便很多）。

以及httpdecrypt

（<https://github.com/lyxhh/lxhToolHTTPDecrypt>）

首先连接adb并运行frida-server；

```
C:\Users\xxx>adb connect 127.0.0.1:7555
```

```
connected to 127.0.0.1:7555
```

```
C:\Users\xxx>adb shell
```

```
root@x86:/ #
```

启动httpdecrypt;

python3 app.py

在运行时可能遇到报错;

[ERROR] device not found, please wait for few seconds and retry.

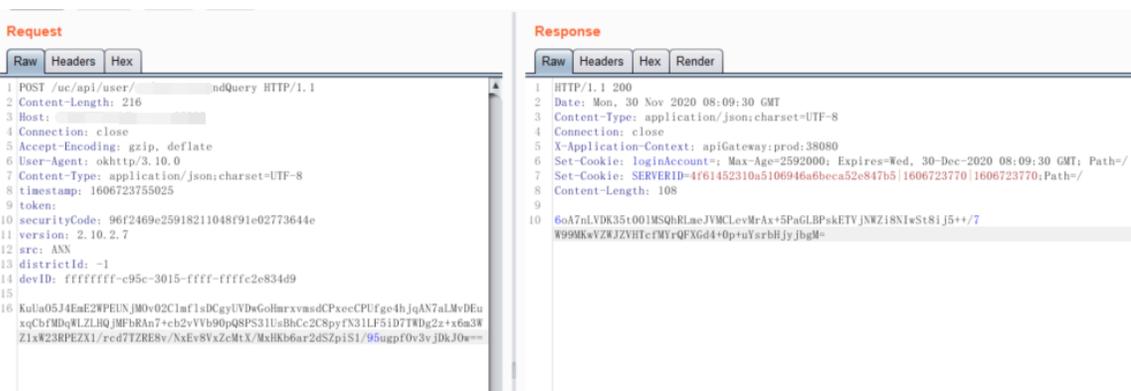
解决: 全局搜索get\_usb\_device, 修改为get\_usb\_device (timeout=1000) 即可。

下载 burp 插件 HTTPDecrpyt 并进行安装。

<https://github.com/lyxhh/lxhToolHTTPDecrypt/releases>

## 这个app不讲武德

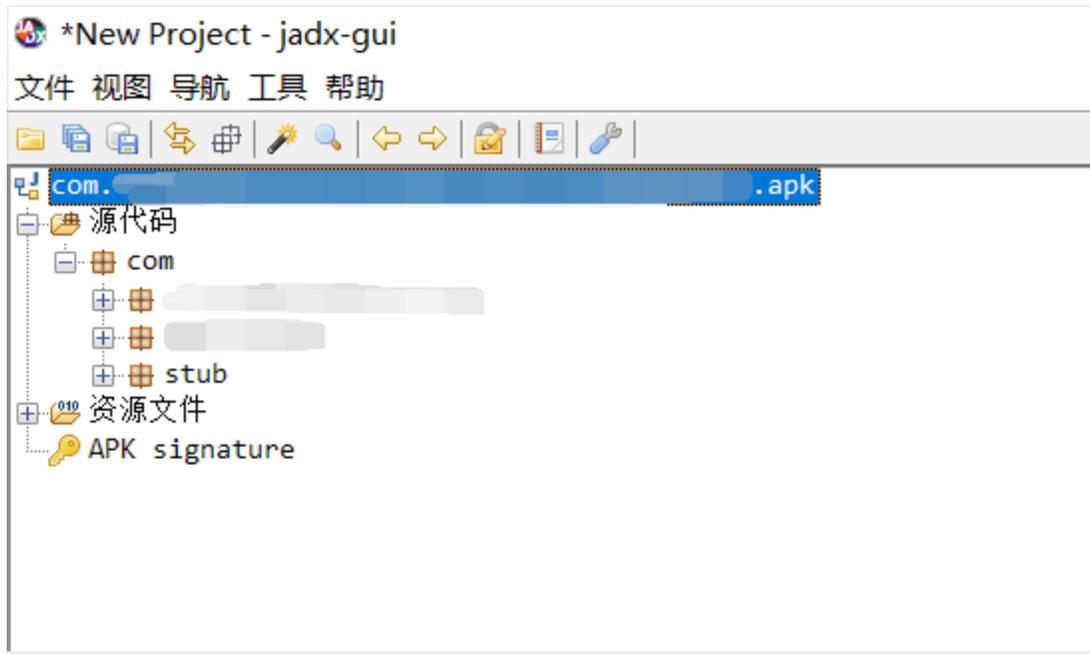
首先正常对app抓包;



The screenshot shows a network capture in Burp Suite. On the left, the 'Request' tab is active, displaying a POST request to /uc/api/user/... with various headers and a body of encrypted data. On the right, the 'Response' tab is active, displaying an HTTP 200 response with headers including Date, Content-Type, and Set-Cookie, and a body of encrypted data.

可以看到app对所有参数都进行了加密, 不论是请求包还是返回包;

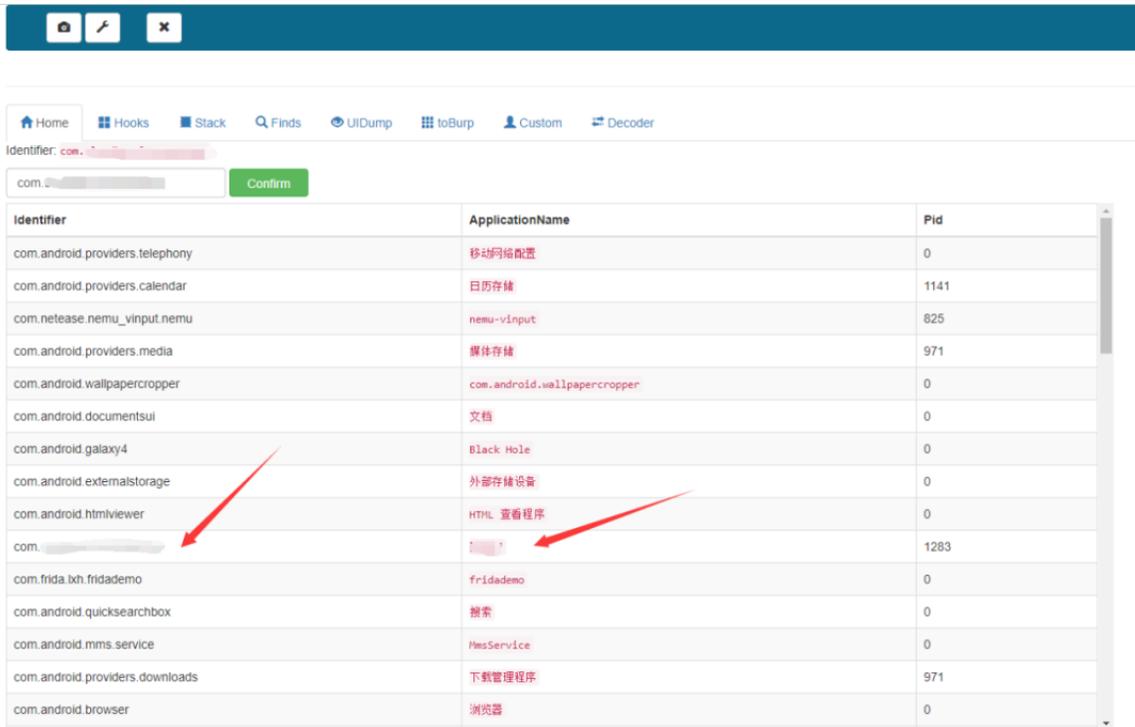
用神器jadx对app进行逆向试试;



有壳，还有加密。

他上来就是一个加壳，一个加密，我全都防出去了嗷

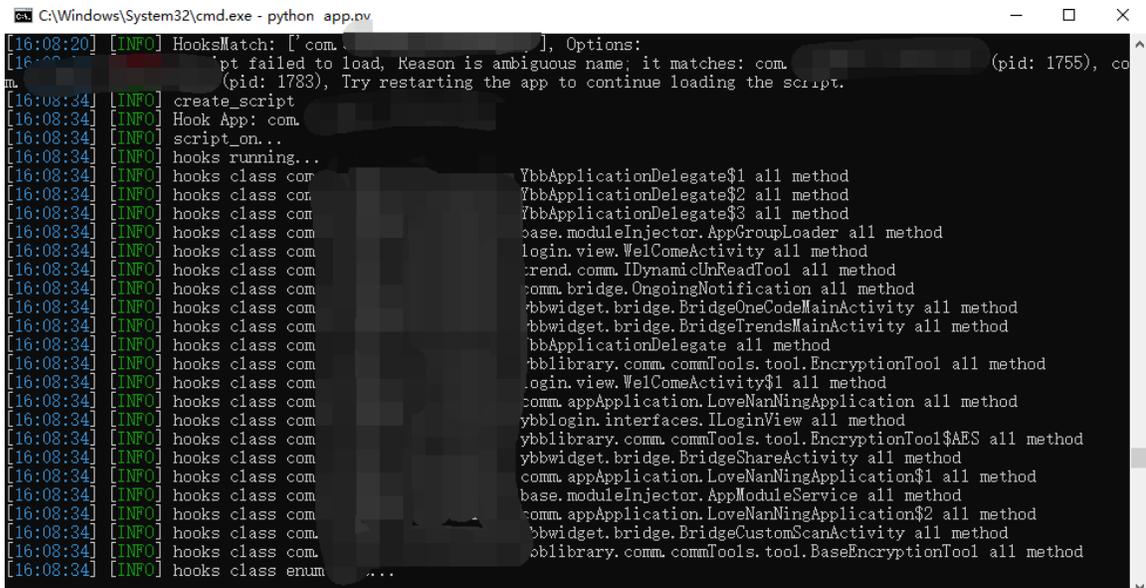
浏览器访问<http://127.0.0.1:8088/>



找到目标app，将app包名com.xxx.xxx填入框中并点击Confirm；

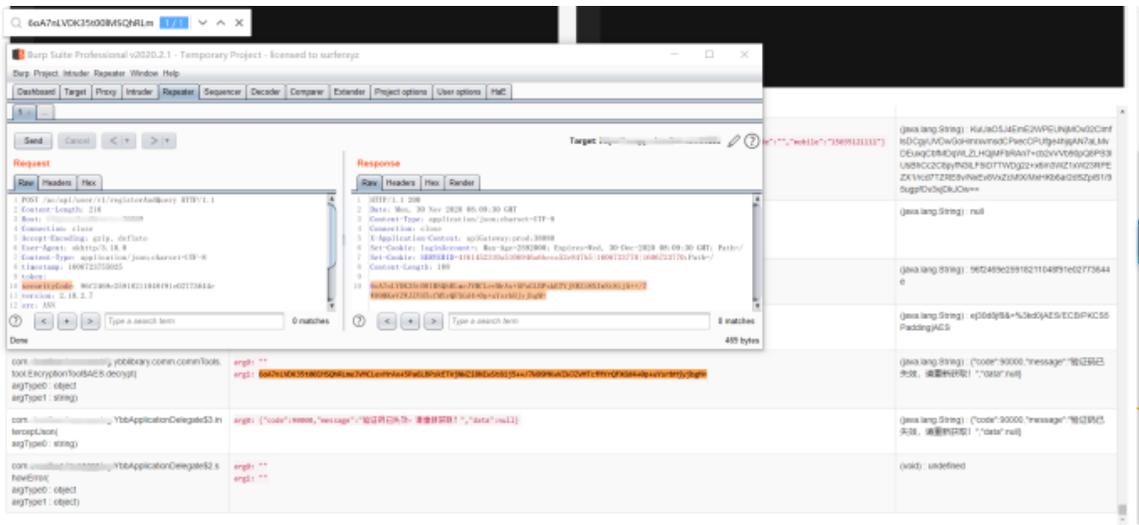
然后点击Hooks功能；

在Match中输入刚才的包名，点击Confirm。



等待输出：





```
com.xxx.xxx.ybblibrary.comm.commTools.tool.EncryptionTool$AES.decrypt(argType0 : object argType1 : string)
```

是返回包的解密函数。

我们看下加密函数：

```
com.xxx.xxx.ybblibrary.comm.commTools.tool.EncryptionTool$AES.encrypt(argType0 : object argType1 : string)
```

可以看到加密函数需要两个参数，分别是对象和字符串。

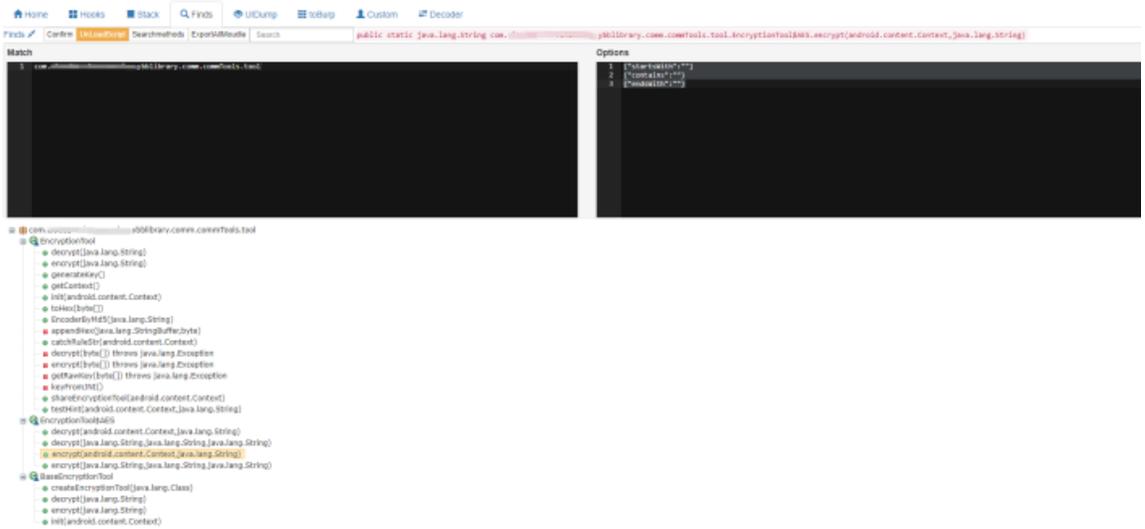
但是我们需要知道对象参数的具体类型。

现在转到Find功能中，搜索函数的类名。

```
com.xxx.xxx.ybblibrary.comm.commTools.tool
```

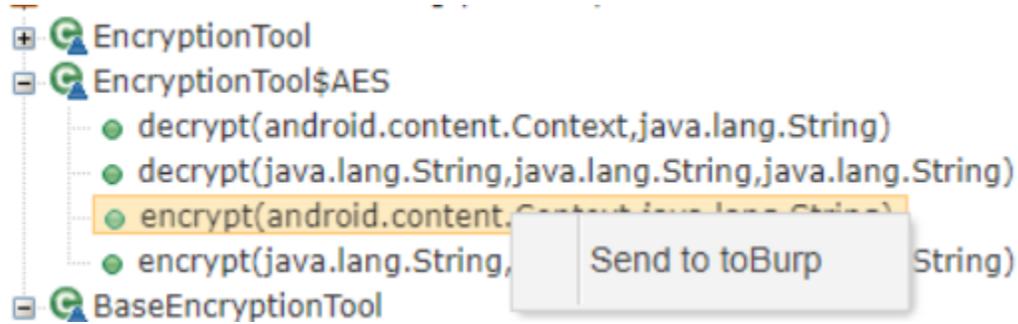
点击Confirm。

找到方法名EncryptionTool\$AES.encrypt。



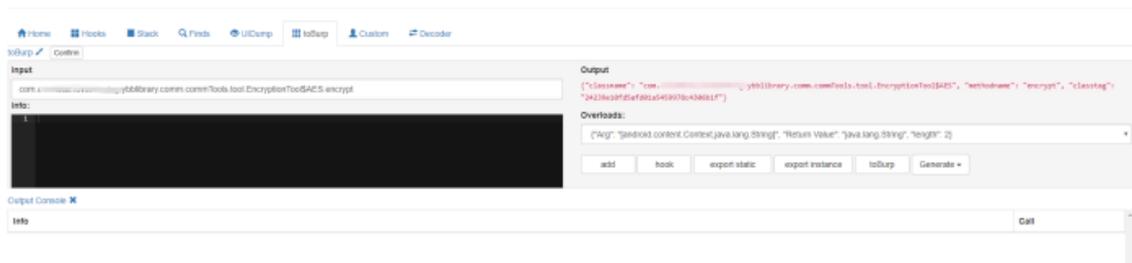
可以看到加密函数参数中，对象类型的参数具体类型为static。

选择到encrypt，然后右键发送到toBurp功能。



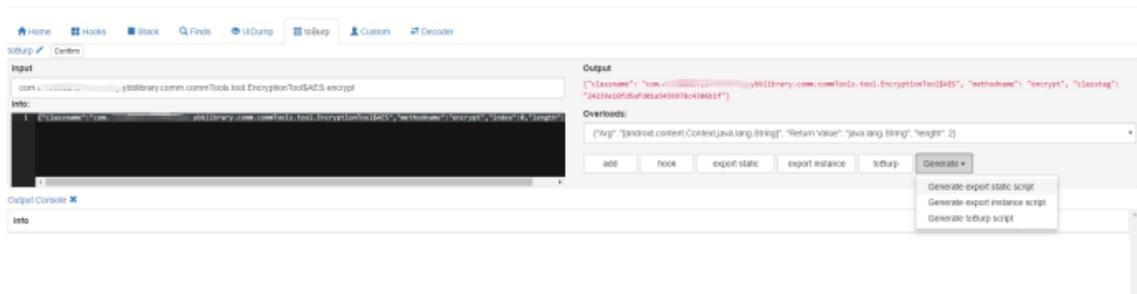
转到toBurp功能。

点击Confirm。



点击add。

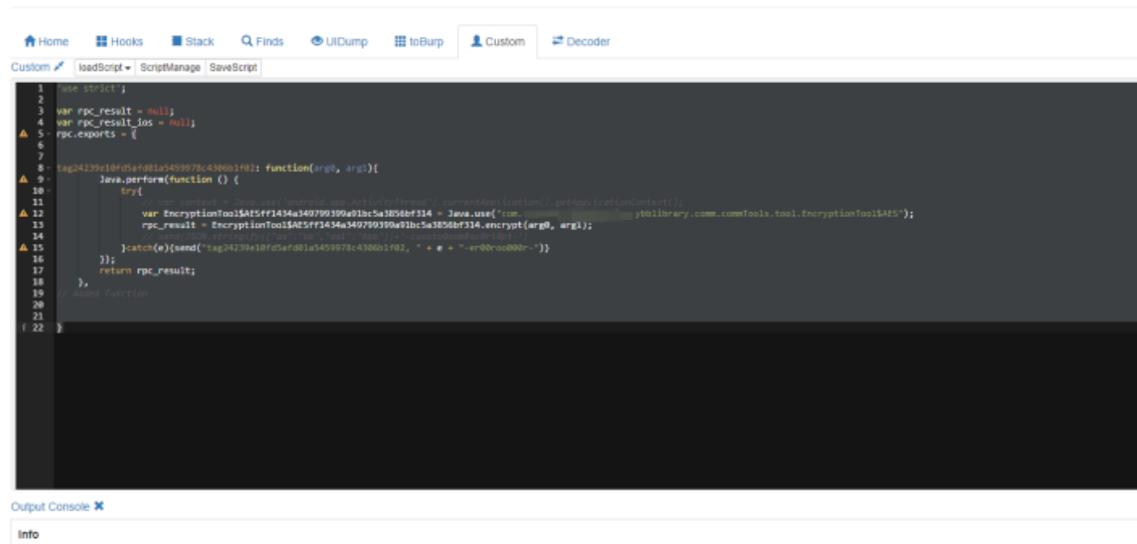
由于这个加密函数有两个参数，所以需要自己编辑代码（默认为一个参数）。



由于对象参数类型为static，所以我们选择Generate export static script。

如果参数为动态，那我们选择Generate export instance script。

现在可以看到，在custom中会生成代码。



代码中，arg0是传过来的加密字符串，arg1是Android的Context对象。

添加代码：

```
var context =
```

```
Java.use('android.app.ActivityThread').currentApplication().getApplicationContext();
```

再用同样的方法生成decrypt的代码。

最终代码为：

```
Custom loadScript ScriptManage SaveScript
1 'use strict';
2
3 var rpc_result = null;
4 var rpc_result_ios = null;
5 rpc.exports = {
6
7
8 tag24239e10fd5afd01a5459978c4306b1f02: function(arg0, arg1){
9   Java.perform(function () {
10     try{
11       var context = Java.use('android.app.ActivityThread').currentApplication().getApplicationContext();
12       var EncryptionTool$AES7ec509488cc1b5017b98358765dbc4f8 = Java.use('com.yblibrary.com.comTools.tool.EncryptionTool$AES');
13       rpc_result = EncryptionTool$AES7ec509488cc1b5017b98358765dbc4f8.encrypt(context, arg0);
14       // send(JSON.stringify({"tag": "aa", "bb": "aa1", "bbb": ""} + "-custo@oom@scdr@bpt-"))
15     }catch(e){send("tag24239e10fd5afd01a5459978c4306b1f02, " + e + "-er@roo@00r-")}
16     });
17     return rpc_result;
18   },
19   // Added Function
20 tagf0f1c91ca14d835ab8ae6e62346a447d02: function(arg0, arg1){
21   Java.perform(function () {
22     try{
23       var context = Java.use('android.app.ActivityThread').currentApplication().getApplicationContext();
24       var EncryptionTool$AES7ec509488cc1b5017b98358765dbc4f8 = Java.use('com.yblibrary.com.comTools.tool.EncryptionTool$AES');
25       rpc_result = EncryptionTool$AES7ec509488cc1b5017b98358765dbc4f8.decrypt(context, arg0);
26       // send(JSON.stringify({"tag": "aa", "bb": "aa1", "bbb": ""} + "-custo@oom@scdr@bpt-"))
27     }catch(e){send("tagf0f1c91ca14d835ab8ae6e62346a447d02, " + e + "-er@roo@00r-")}
28     });
29     return rpc_result;
30   },
31   // Added Function
32
33 }
f 34 }
```

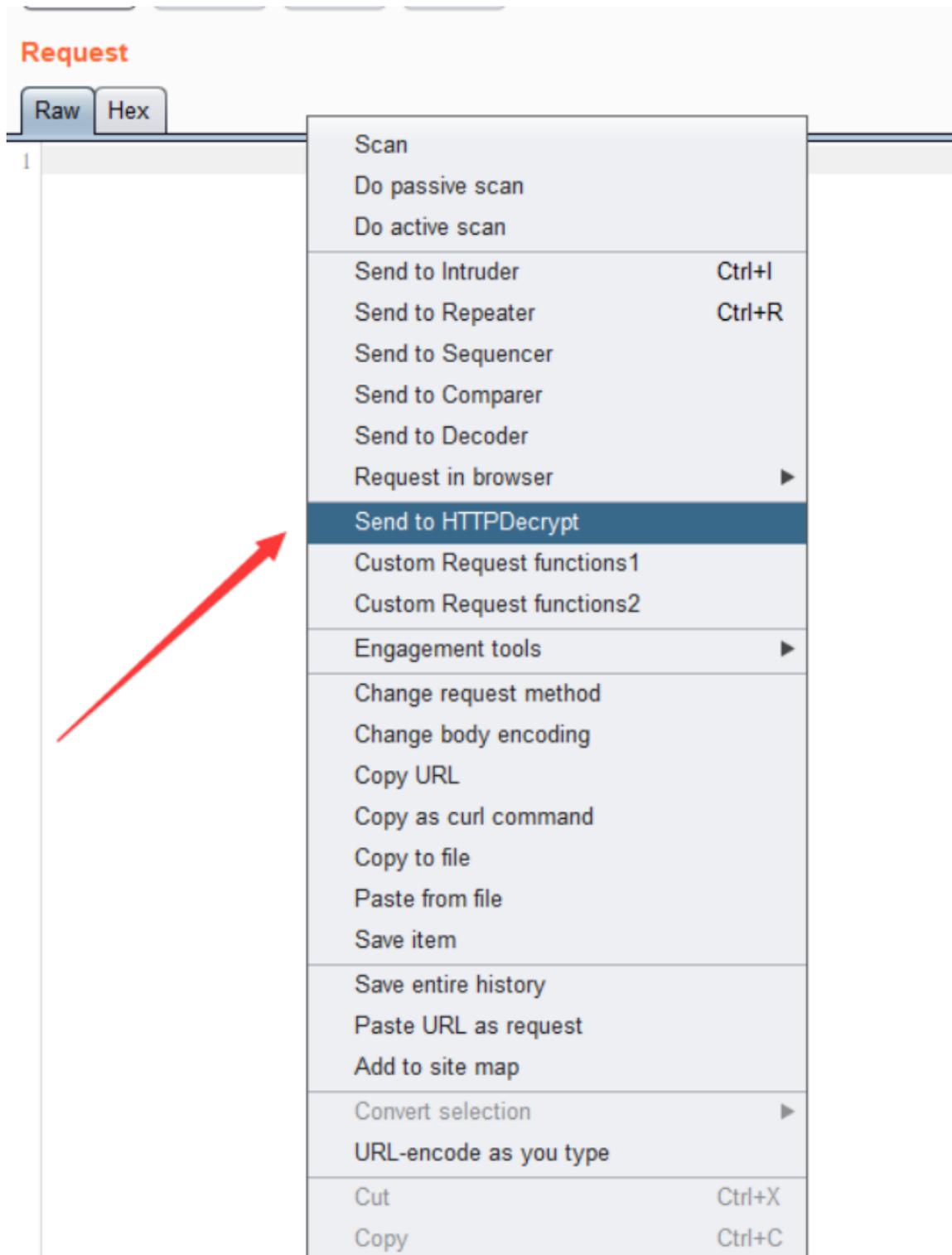
Output Console Info

点击左上角loadScript 将脚本发送到burp。

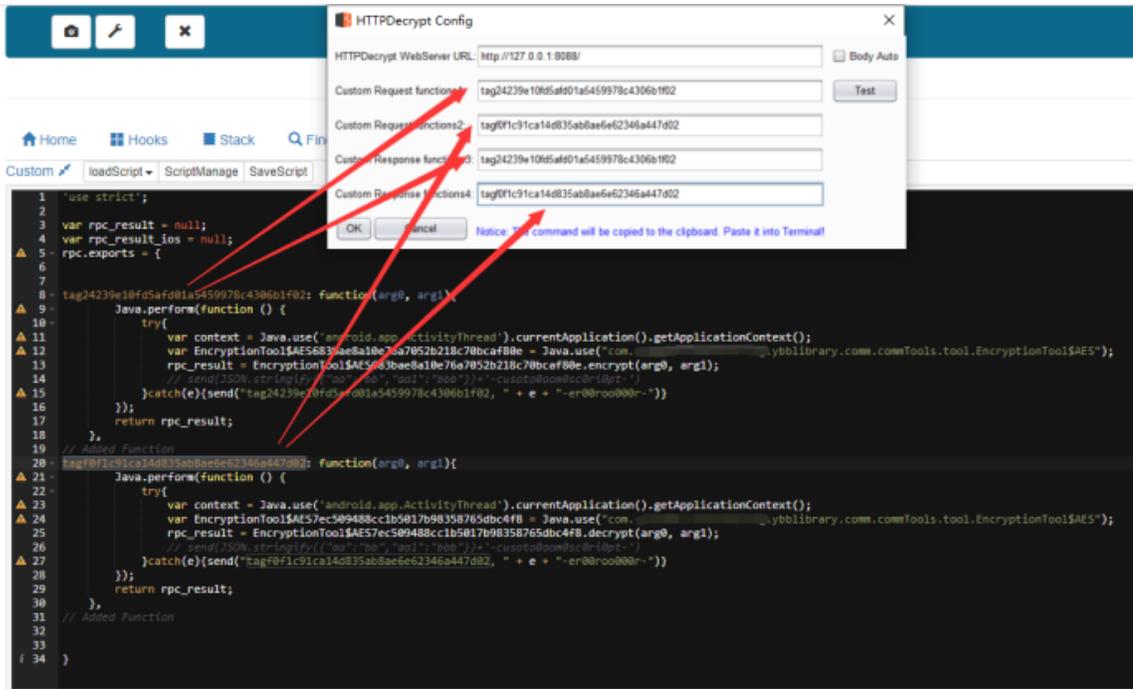
```
Home Hooks Stack Finds UIDump toBurp Custom Decoder
Custom loadScript ScriptManage SaveScript
1 'use strict';
2   loadScript
3   var
4   loadScript(SleepResumeApp)
5   rpc.exports = {
6
7
8 tag24239e10fd5afd01a5459978c4306b1f02: function(arg0, arg1){
9   Java.perform(function () {
10     try{
11       var context = Java.use('android.app.ActivityThread').currentApplication().getApplicationContext();
12       var EncryptionTool$AES7ec509488cc1b5017b98358765dbc4f8 = Java.use('com.yblibrary.com.comTools.tool.EncryptionTool$AES');
13       rpc_result = EncryptionTool$AES7ec509488cc1b5017b98358765dbc4f8.encrypt(arg0, arg1);
14       // send(JSON.stringify({"tag": "aa", "bb": "aa1", "bbb": ""} + "-custo@oom@scdr@bpt-"))
15     }catch(e){send("tag24239e10fd5afd01a5459978c4306b1f02, " + e + "-er@roo@00r-")}
16     });
17     return rpc_result;
18   },
19   // Added Function
20 tagf0f1c91ca14d835ab8ae6e62346a447d02: function(arg0, arg1){
21   Java.perform(function () {
22     try{
23       var context = Java.use('android.app.ActivityThread').currentApplication().getApplicationContext();
24       var EncryptionTool$AES7ec509488cc1b5017b98358765dbc4f8 = Java.use('com.yblibrary.com.comTools.tool.EncryptionTool$AES');
25       rpc_result = EncryptionTool$AES7ec509488cc1b5017b98358765dbc4f8.decrypt(arg0, arg1);
26       // send(JSON.stringify({"tag": "aa", "bb": "aa1", "bbb": ""} + "-custo@oom@scdr@bpt-"))
27     }catch(e){send("tagf0f1c91ca14d835ab8ae6e62346a447d02, " + e + "-er@roo@00r-")}
28     });
29     return rpc_result;
30   },
31   // Added Function
32
33 }
f 34 }
```

Output Console Info

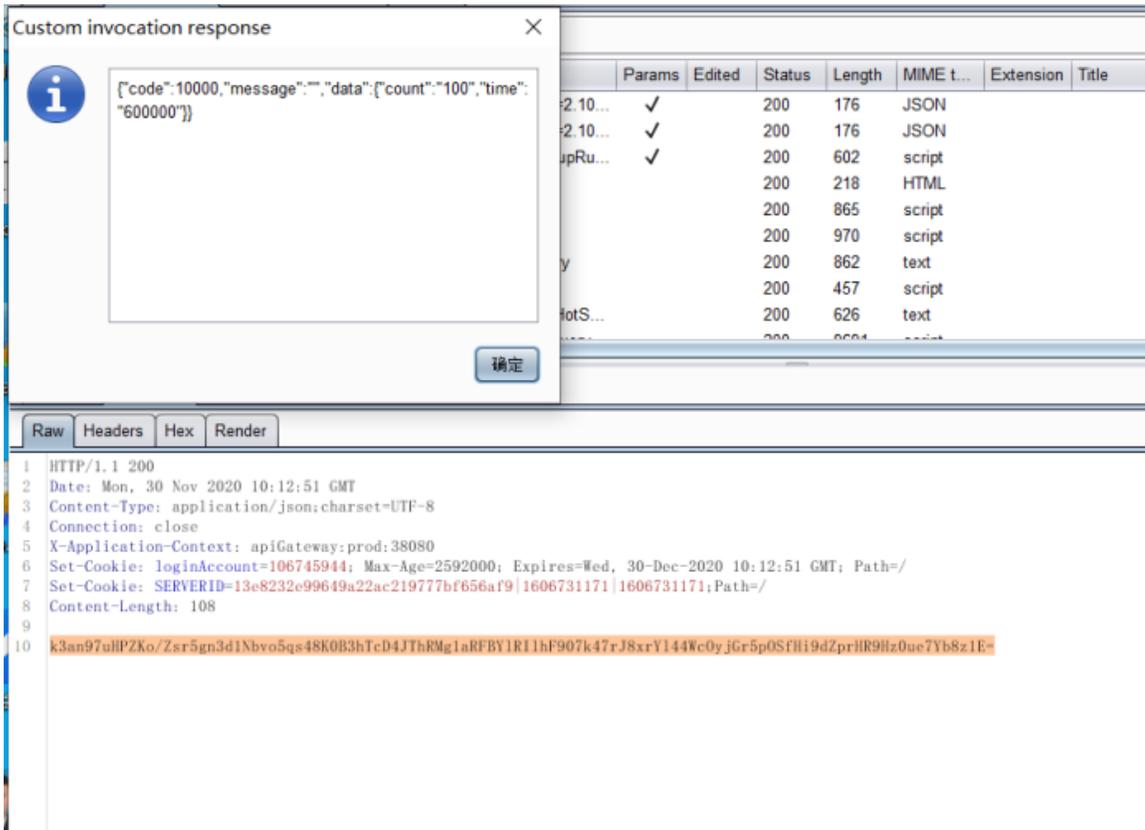
在burpsutie中进行配置。



点击send to HttpDecrypt 打开配置界面。



然后就可以对加密后的数据进行加解密了。



## 耗子为知，好好反思

app测试的时候，hook的难点在于定位所需要hook的函数和参数。然后就是根据函数名和参数名进行hook代码的编写。

httpDecrypt提供了较为方便的集成的环境。相比脱壳看代码，这种调试的方法会省下我们大量的时间。



## 招聘信息

[点击了解 >>](#)

**安全咨询热线**

**400-0309-360**



知其黑 守其白

分享知识盛宴，闲聊大院趣事，备好酒肉等你



长按二维码关注 酒仙桥六号部队

精选留言

---

用户设置不下载评论