# Frida-brida之hook抓包初探

## 环境准备

1、安装npm

<https://nodejs.org/en/download/> 官网下载后直接安装

2、安装frida：

pip install frida

pip install frida-tools

npm install [frida-compile@9.3.0](mailto:frida-compile@9.3.0)

注：Frida-compile较新版本缺少brida编译参数会报错，具体那个版本后去除参数的未进行测试，有兴趣的可自行尝试。

3、安装birda和依赖：

Github下载v.0.4导入burp扩展中:https://github.com/federicodotta/Brida/releases/tag/v0.4

pip install Pyro4

注：burp商店自带的也有brida，当不介意在商店直接安装，笔者在burp商店直接安装brida时hook的js脚本无法解释参数，最后换官网的包解决的问题，本萌新也不知道原因，求大佬指点。

4、测试apk环境搭建

测试apk下载地址：https://pan.baidu.com/s/1r2pKkbsB22FMfu\_bD2vhtw

安装：Adb install -t eseBrida.apk

注：本着最快入手讲解使用的原则使用了大佬的测试app。（好吧其实就是本人菜对加解函数的查找还不太会，先用测试app练练手，后续学会加密姿势后可能会继续写个学习过程）

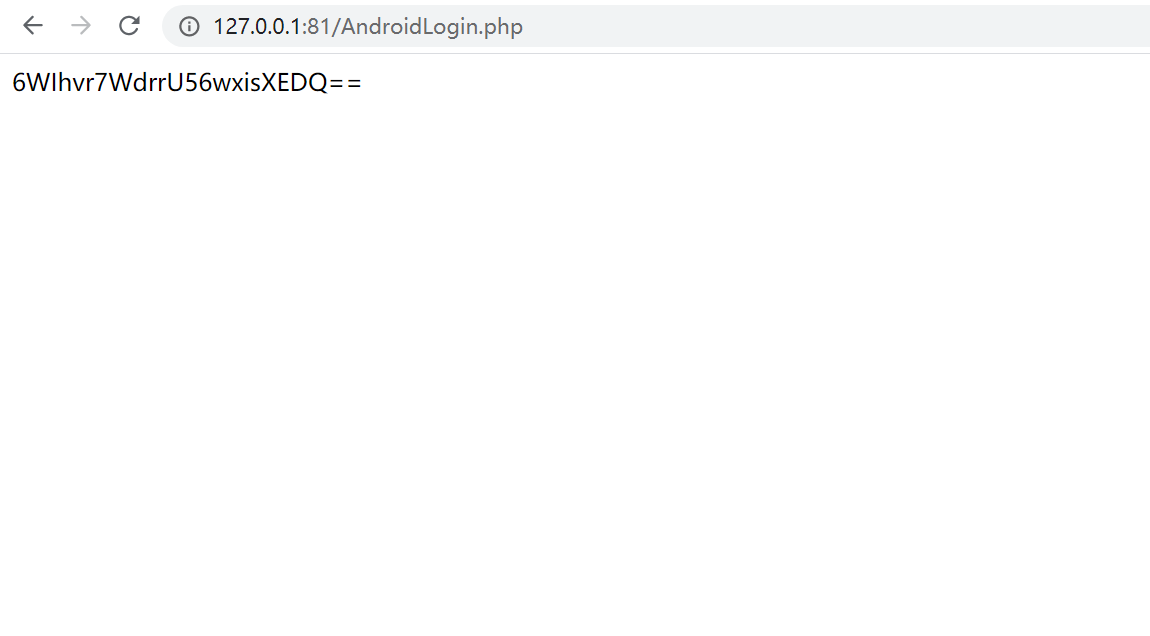
服务端环境搭建：



如上图所示，下载后四个文件，有hook.js脚本和服务端的处理代码，还有个就是开启frida服务的脚本，如需使用请参考实际环境。

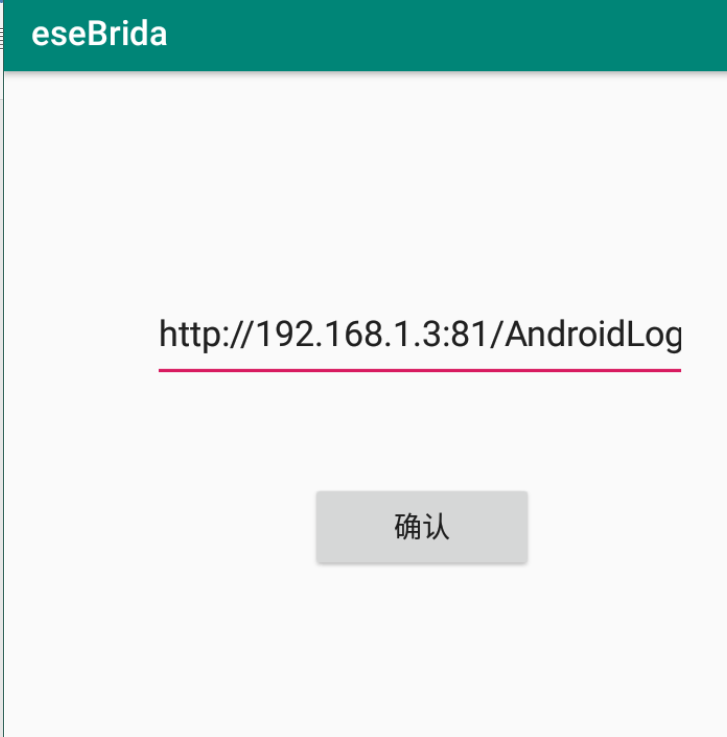
一切为了效率，我们可以直接使用phpstudy搭建服务，将php文件放进phpstudy的web根目录即可。

访问结果如下：



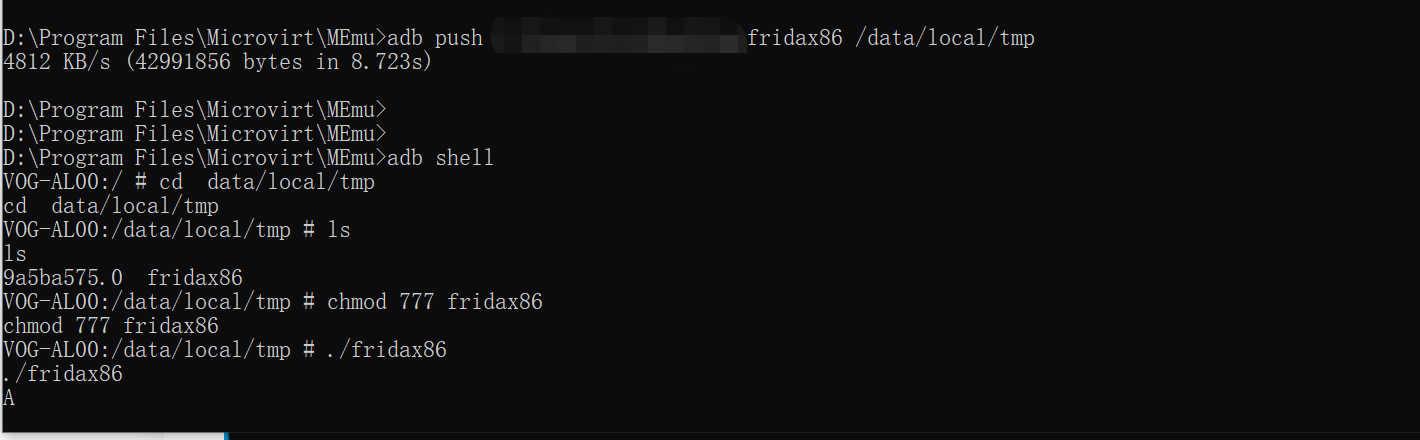
注：如果访问看到的是源代码或空白页可能是php版本问题，更改版本即可，此处我用的是5.4.45

在apk内设置服务地址。

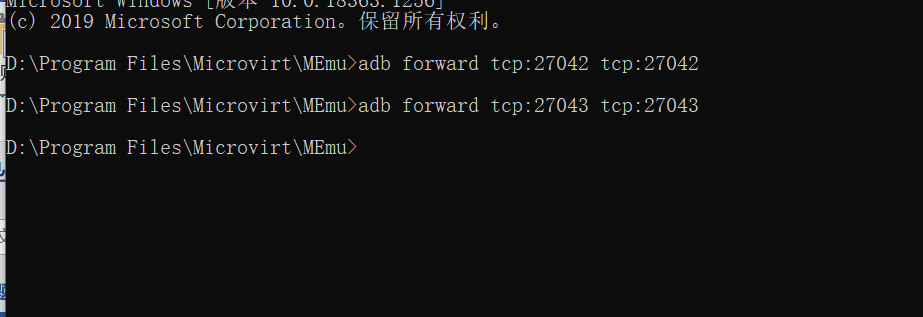


## 开始

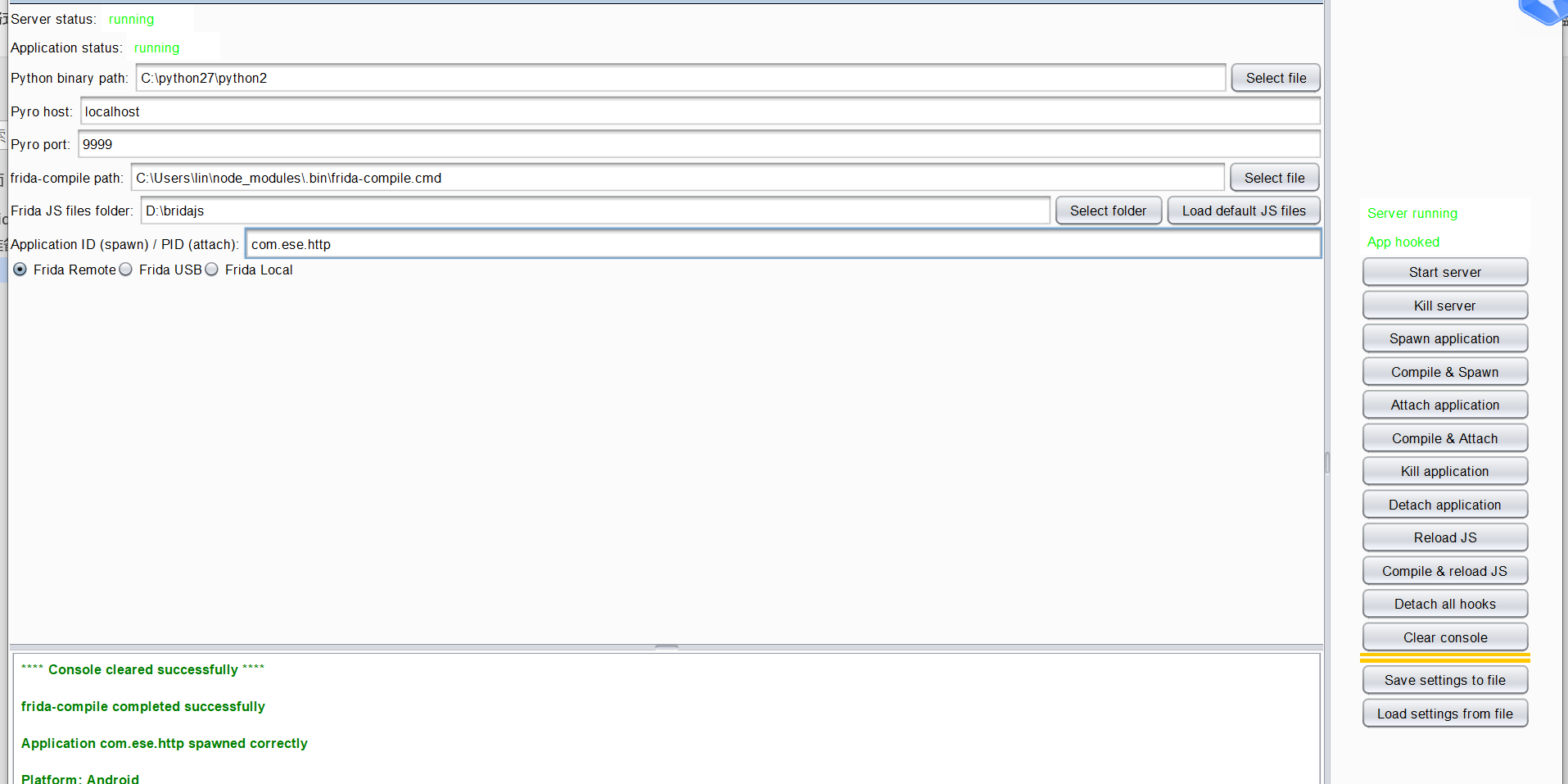
下载对应的frida-server版本并传入测试机中运行：



进行端口转发：

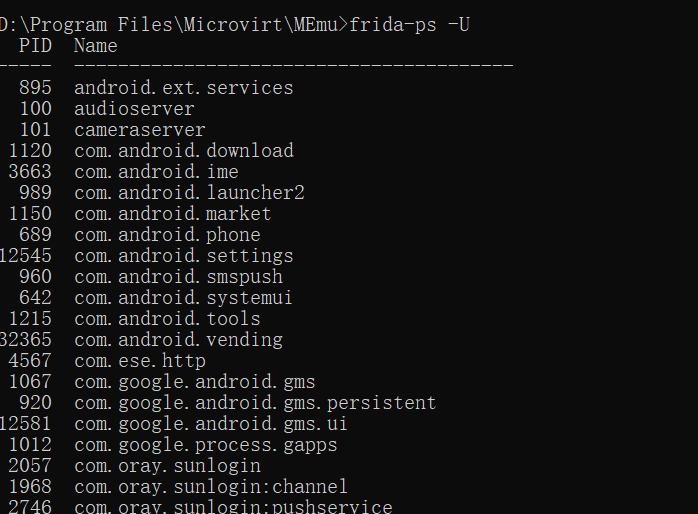


进入burp开启服务和应用



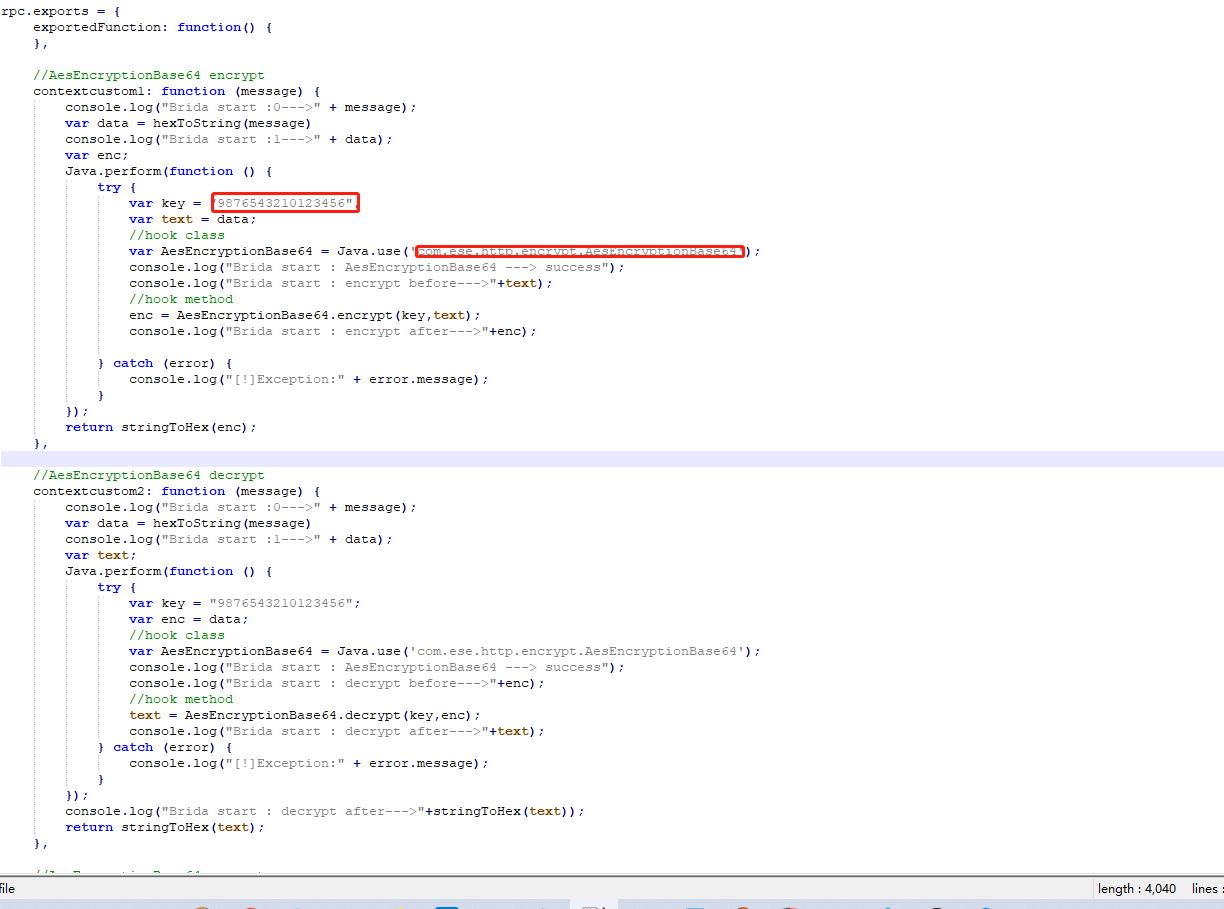
Js直接加载默认js脚本就好，在默认js内进行修改，两个路径选择安装时的路径即可，找不到可用everything等工具搜索文件名找到路径。

包名可通过下图命令进行查找或在打开软件后使用adb shell dumpsys window w |findstr \/ |findstr name= 进行查看

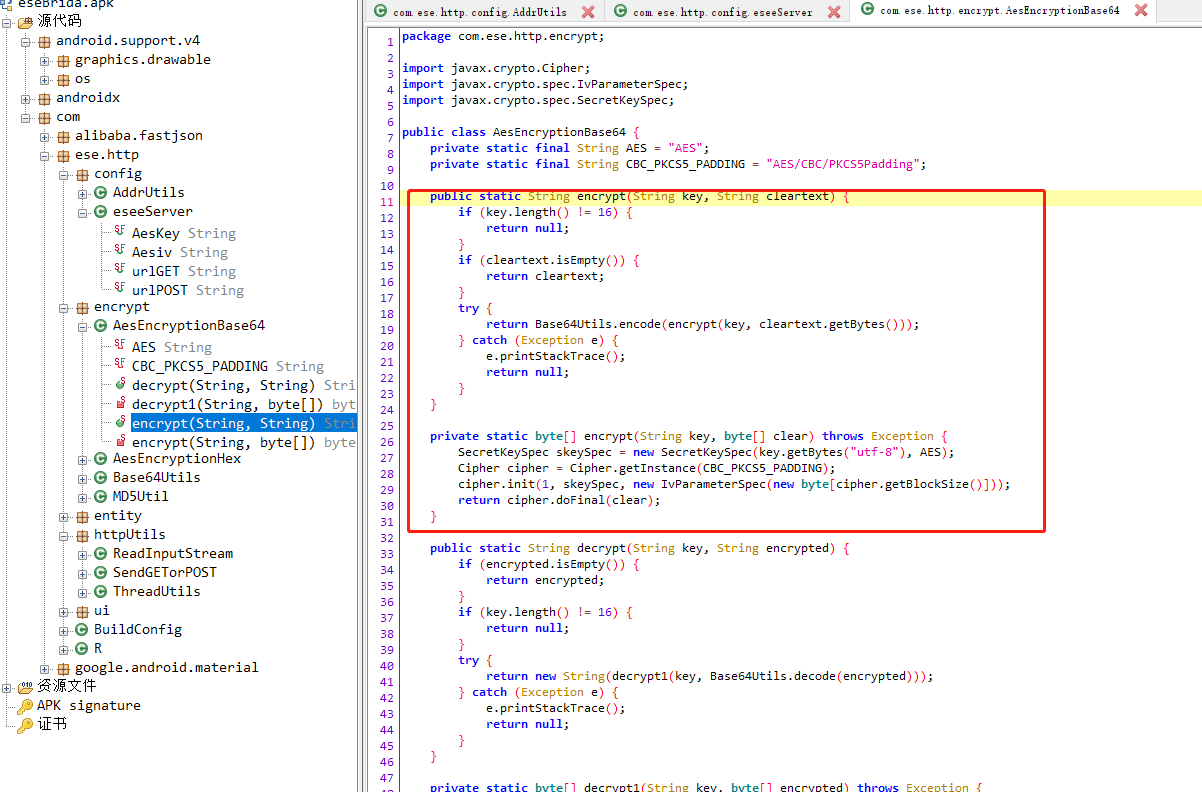


## 脚本调试

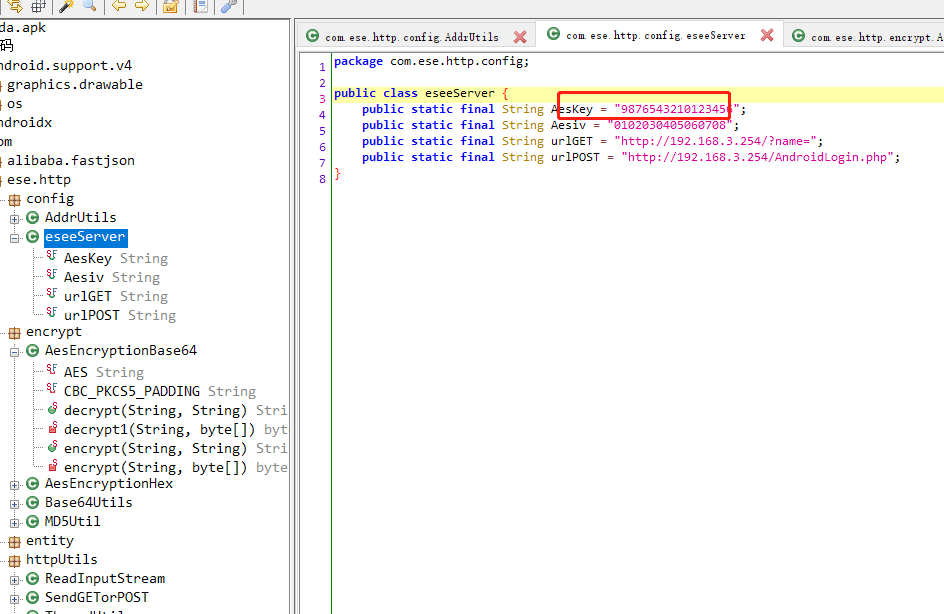
查看大佬的js文件：



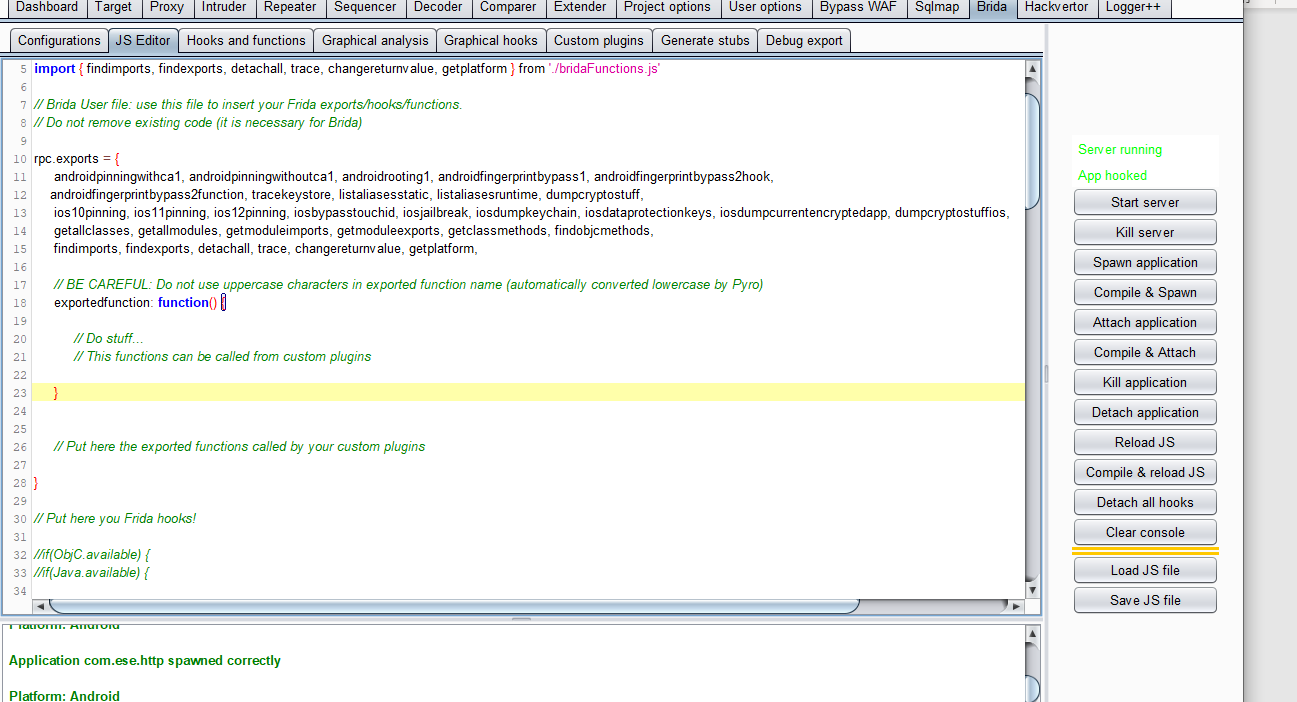
对比反编译的加密算法：



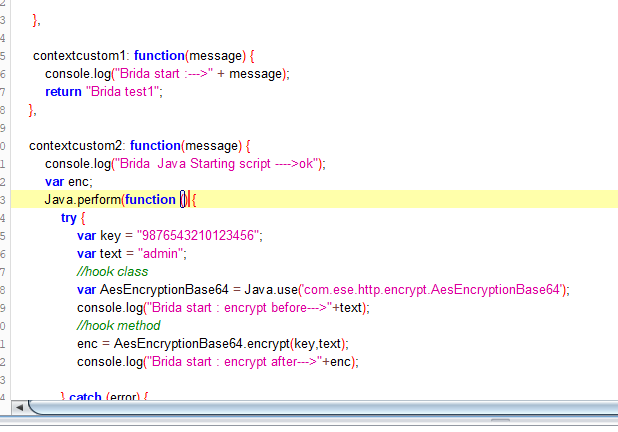
Key值：



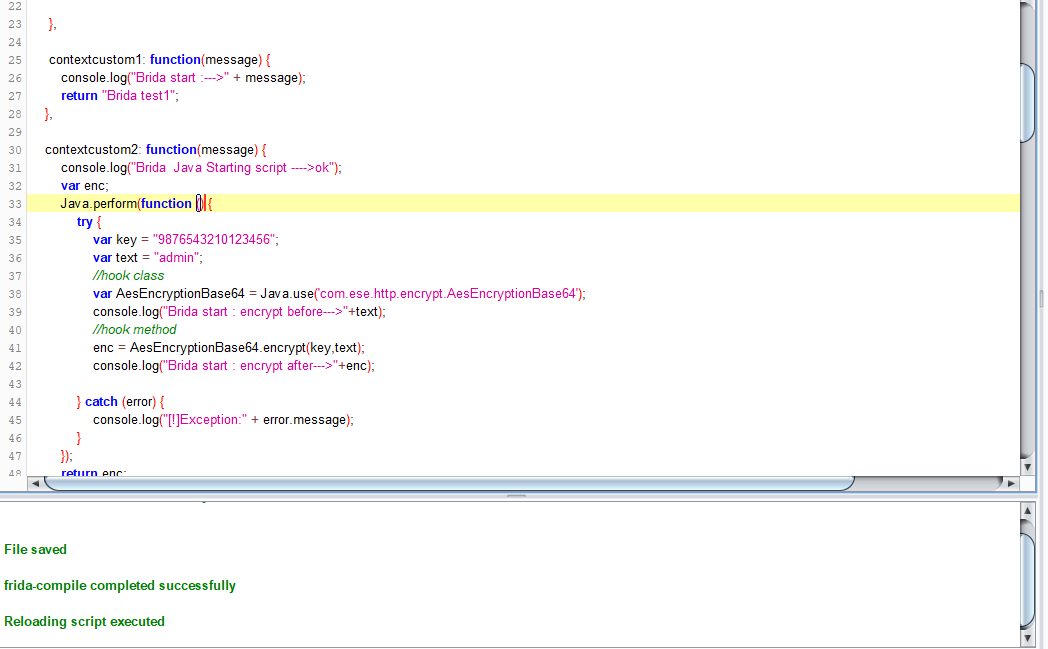
加载默认的brida.js：



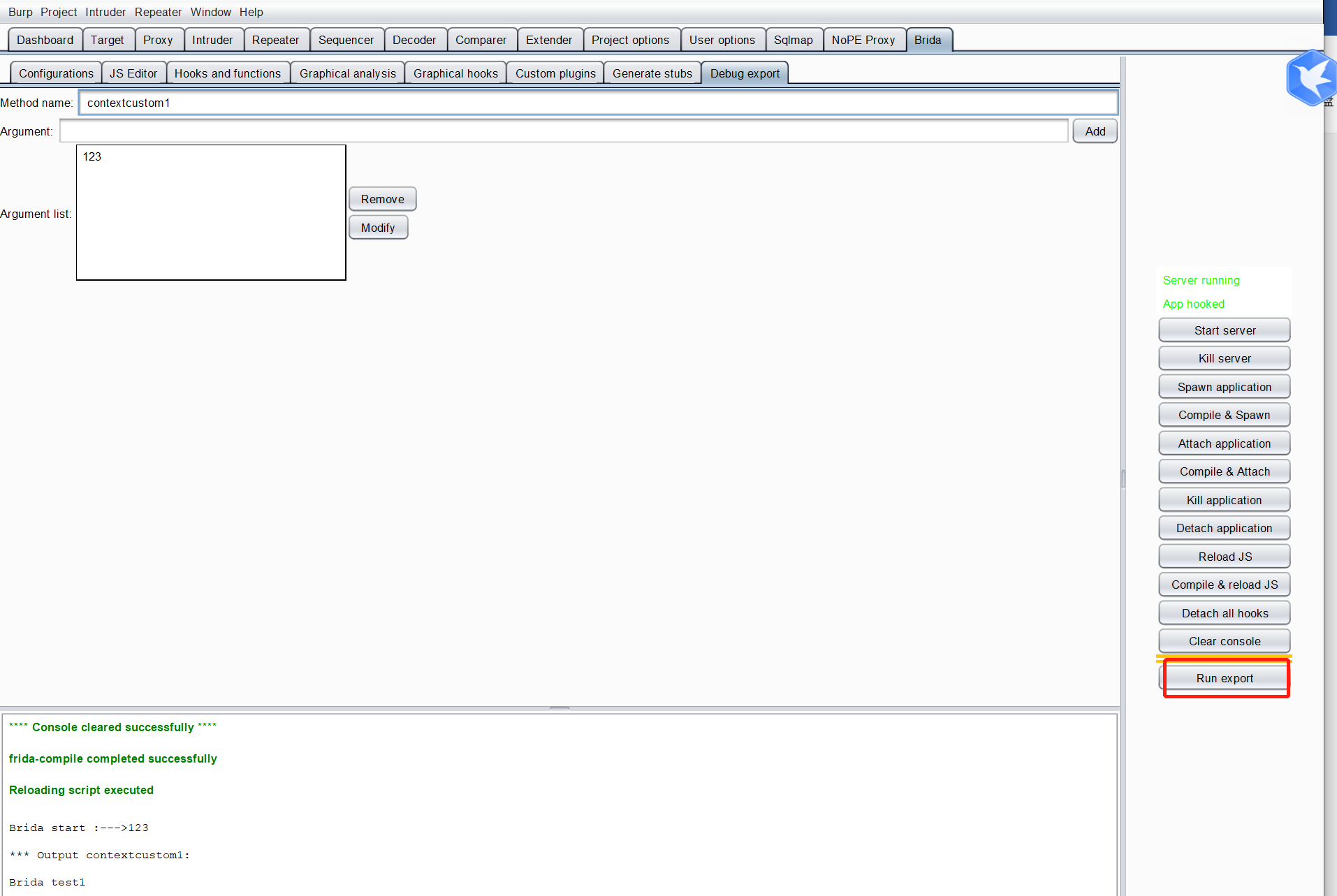
写入下图种的js代码：



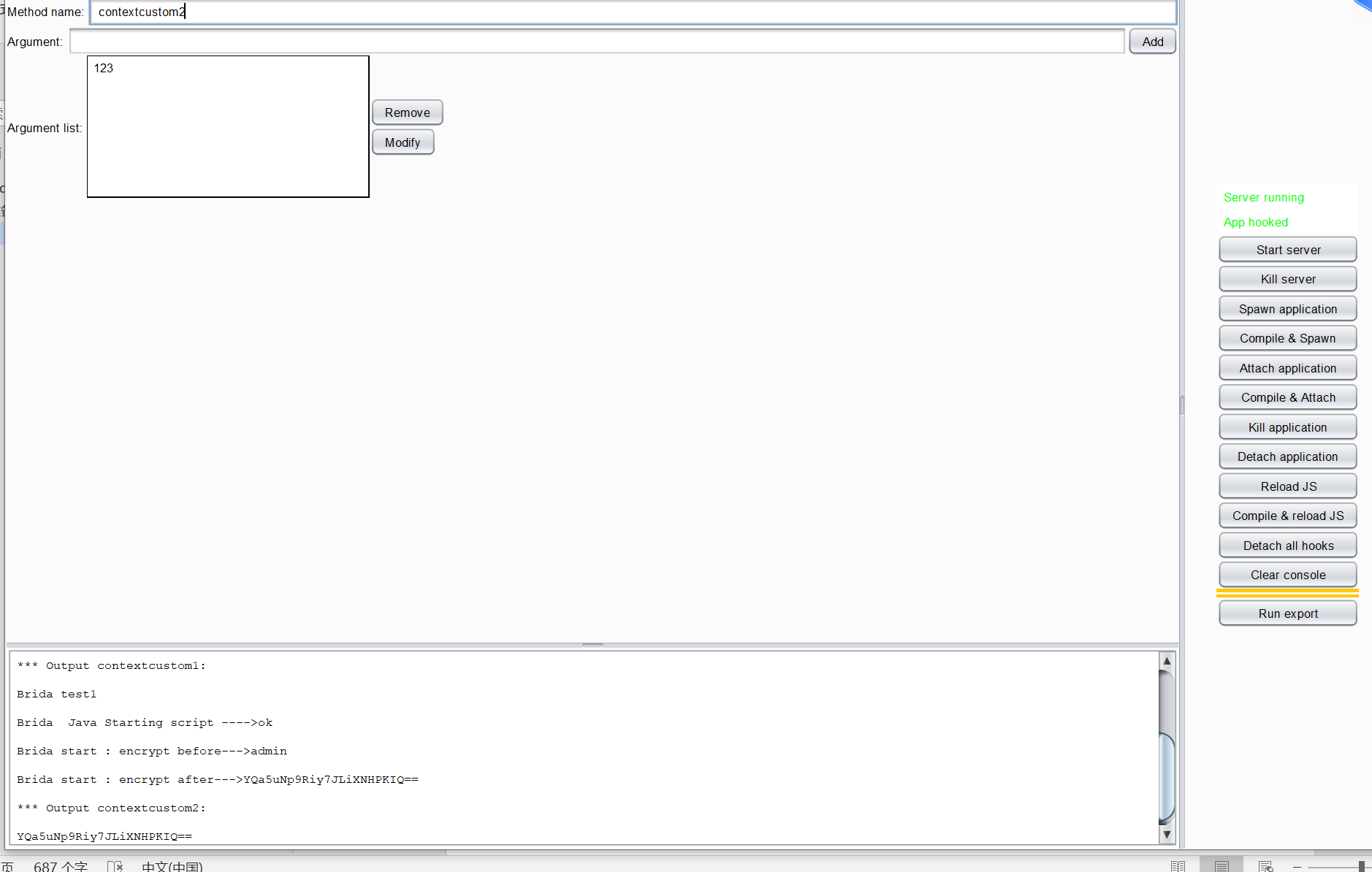
保存并重新编译js，然后进行调试（每次修改js代码后都需要进行保存编译操作）：

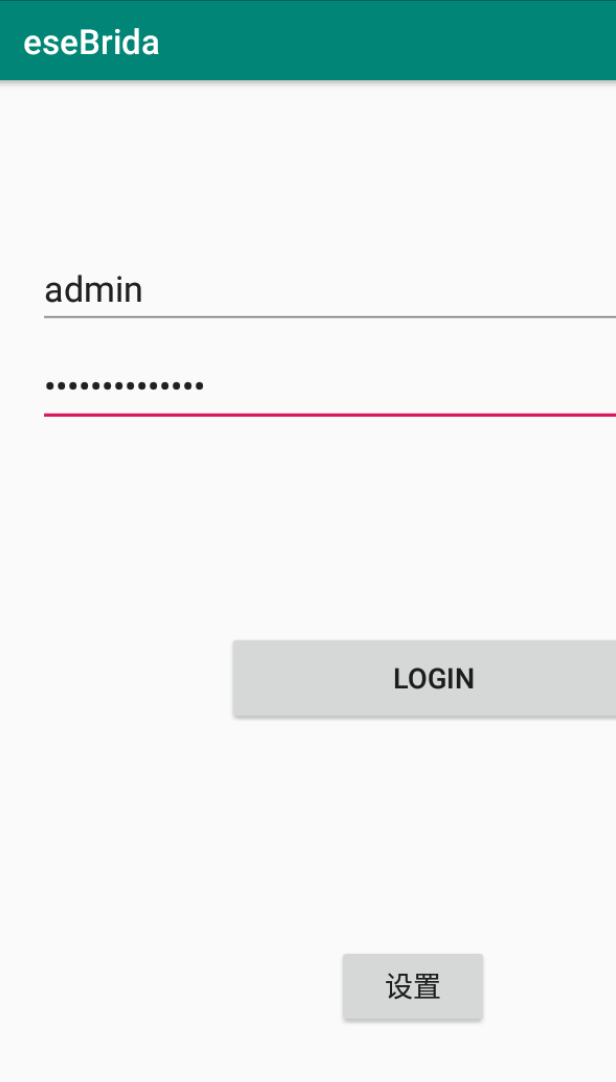


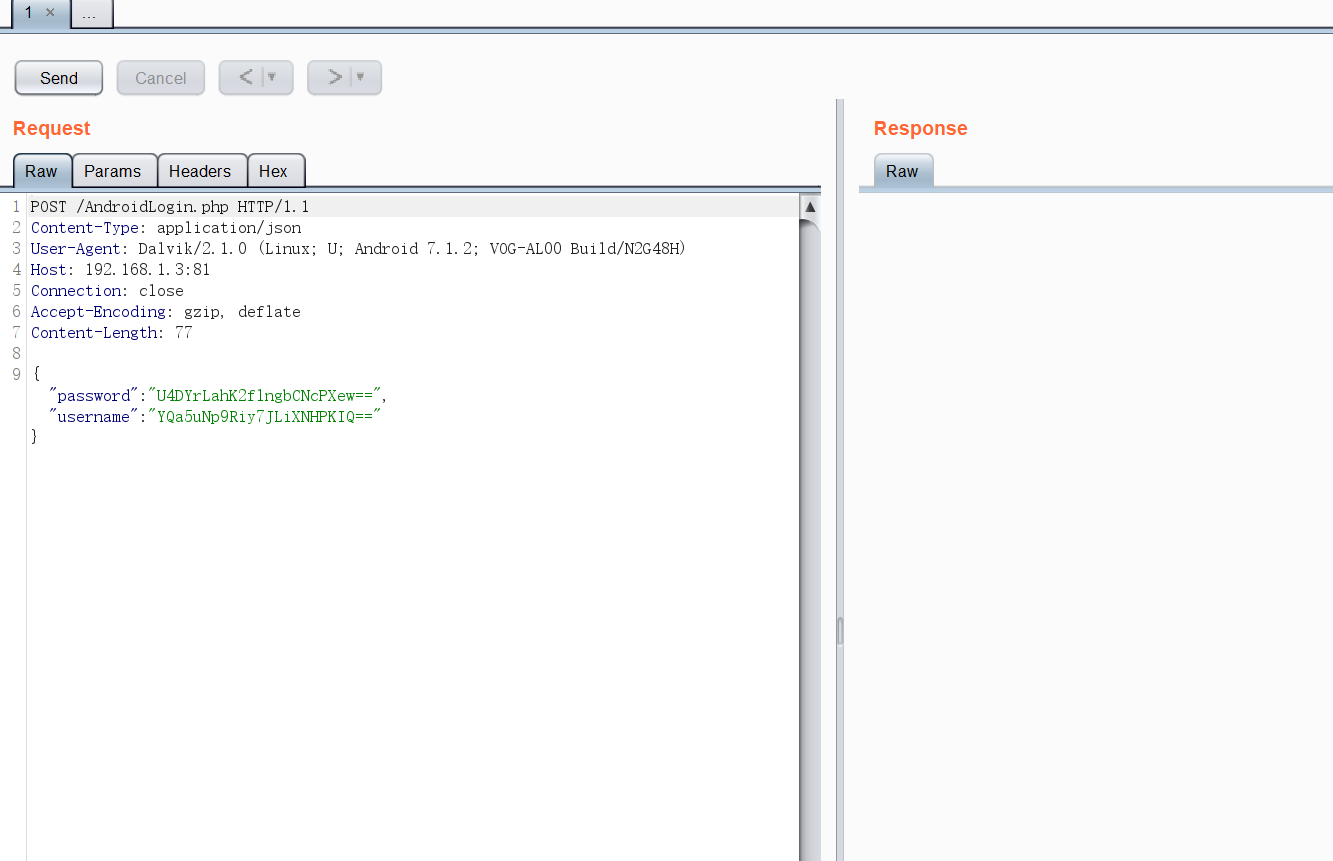
添加数据运行contextcustom1方法，确认brida有无问题：



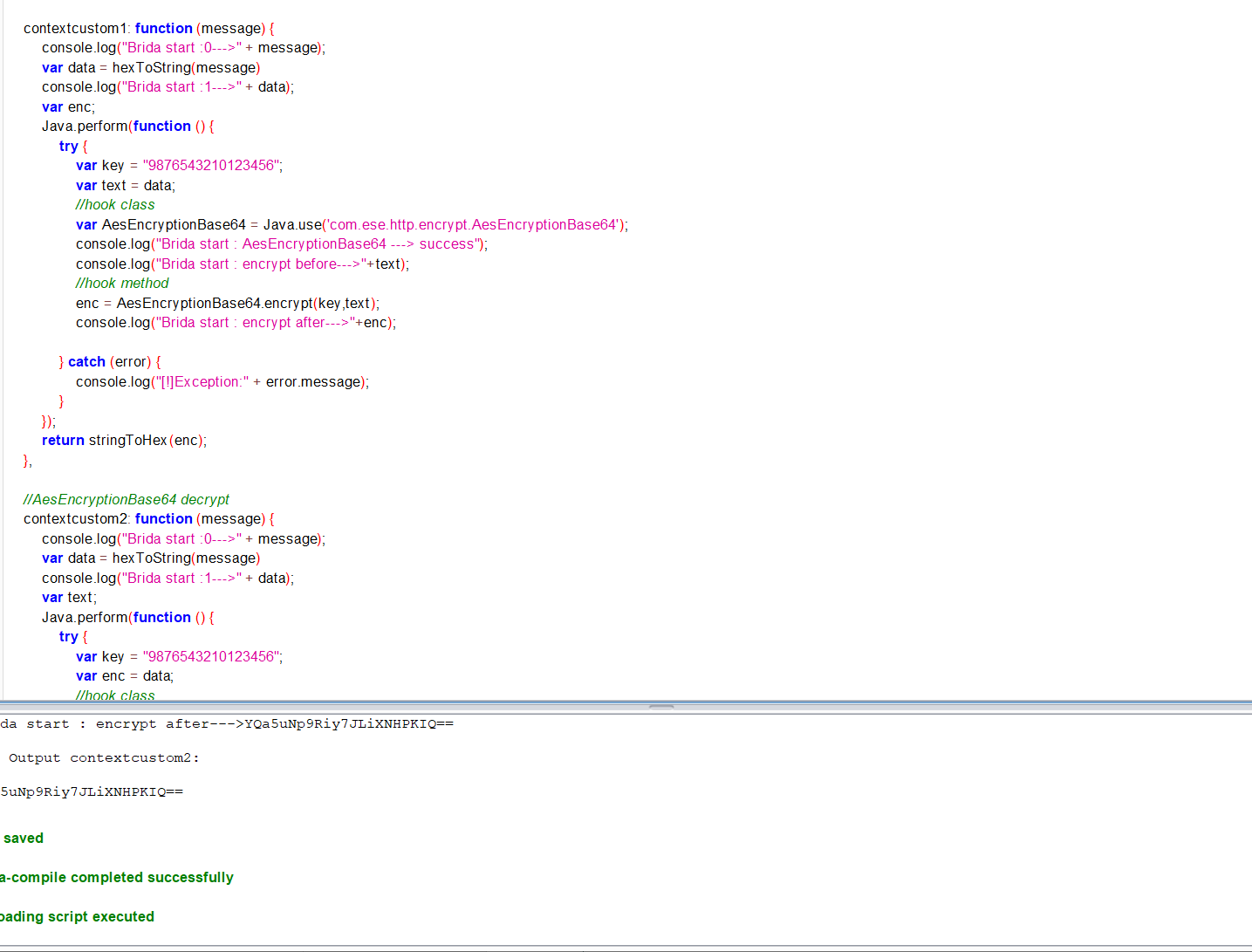
运行contextcustom2方法，对比加解密函数是否正确：

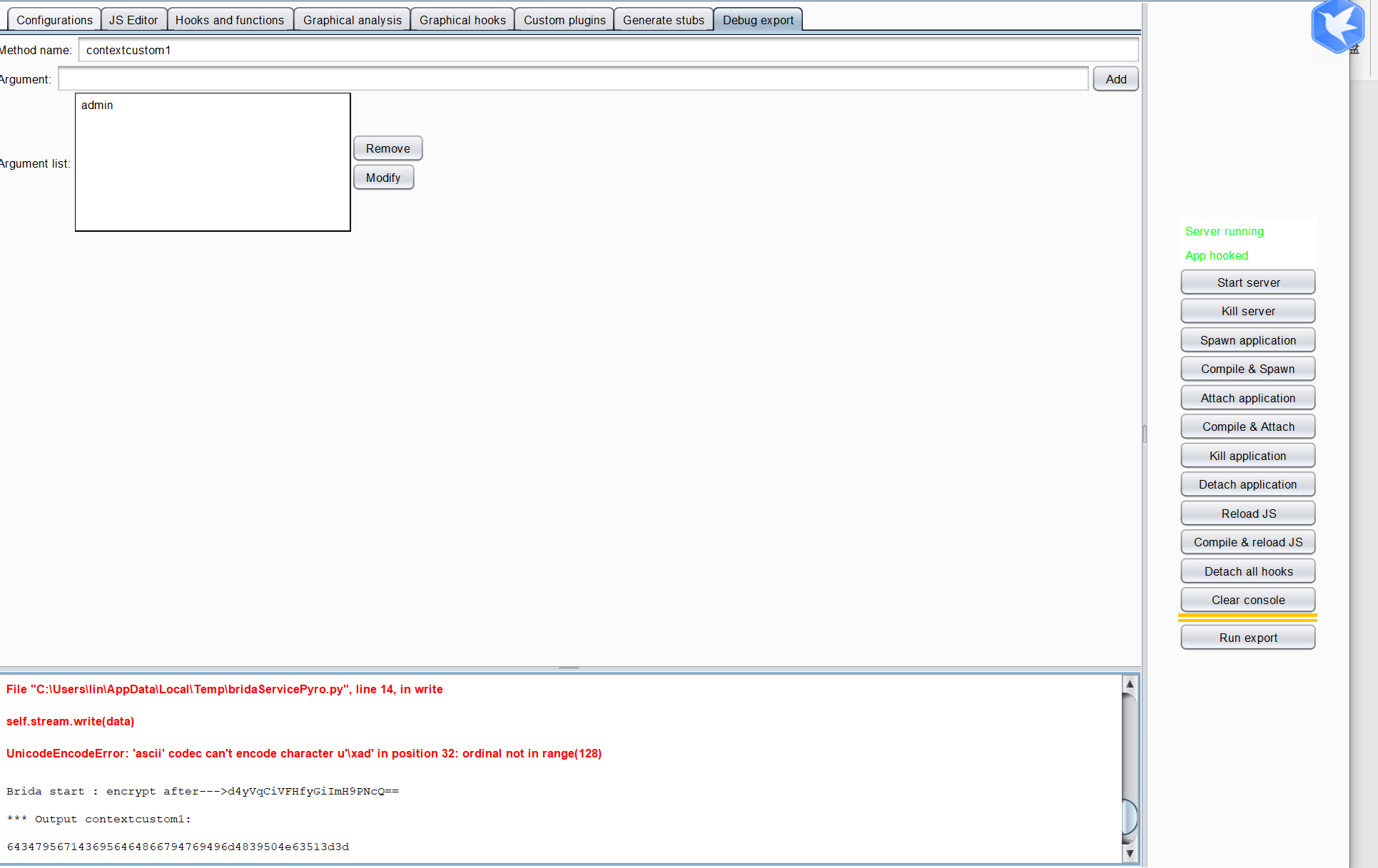


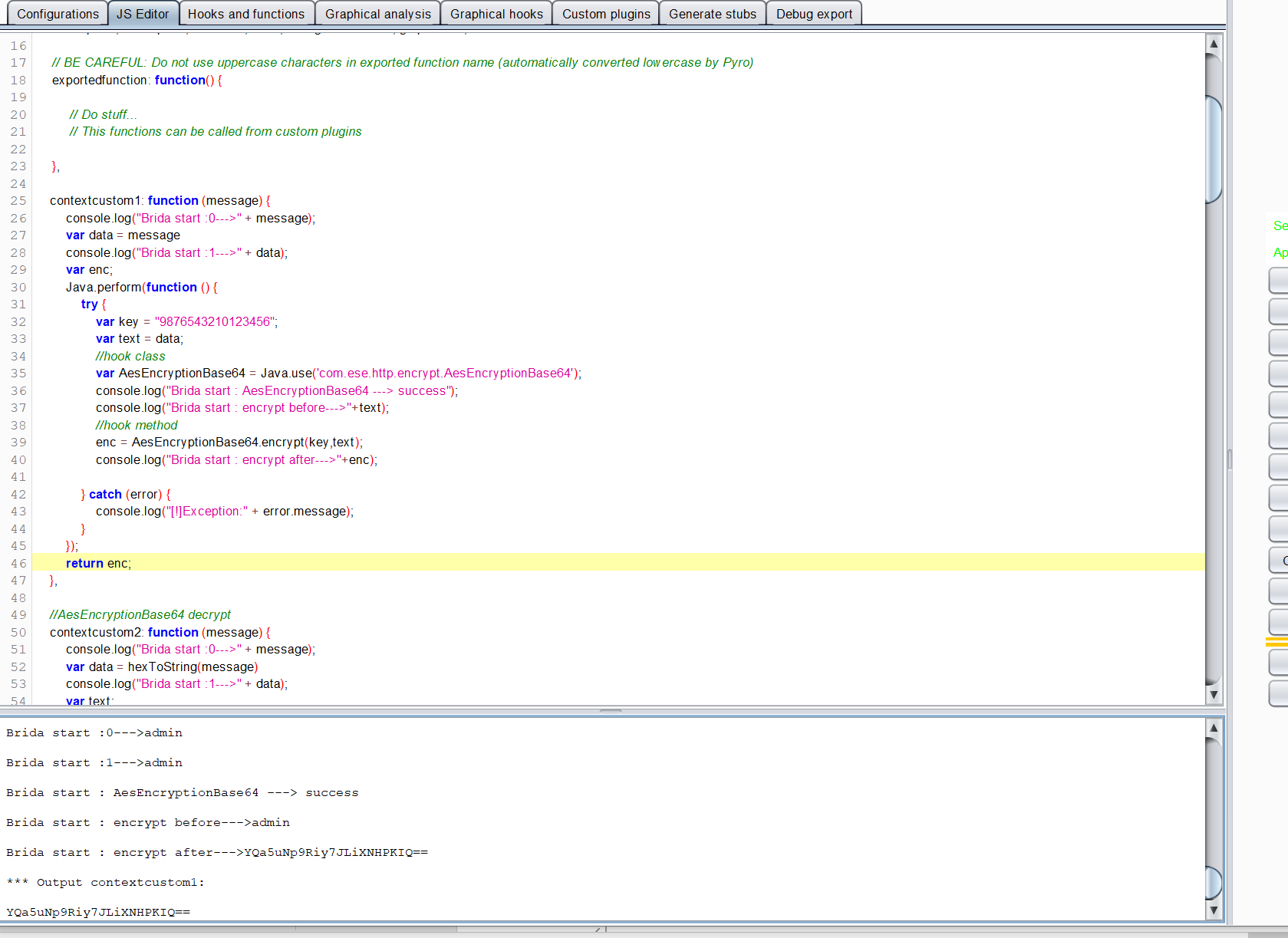


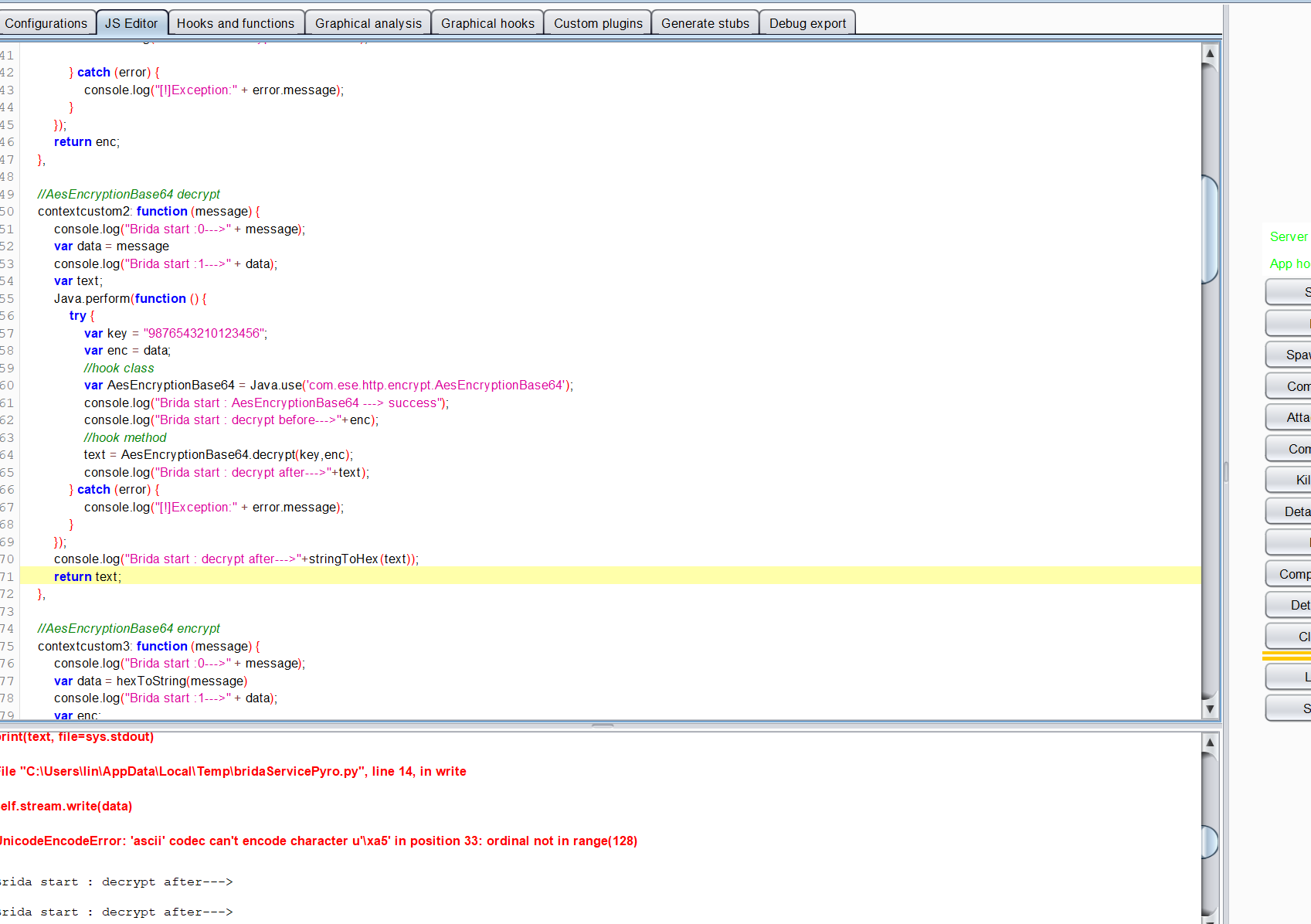


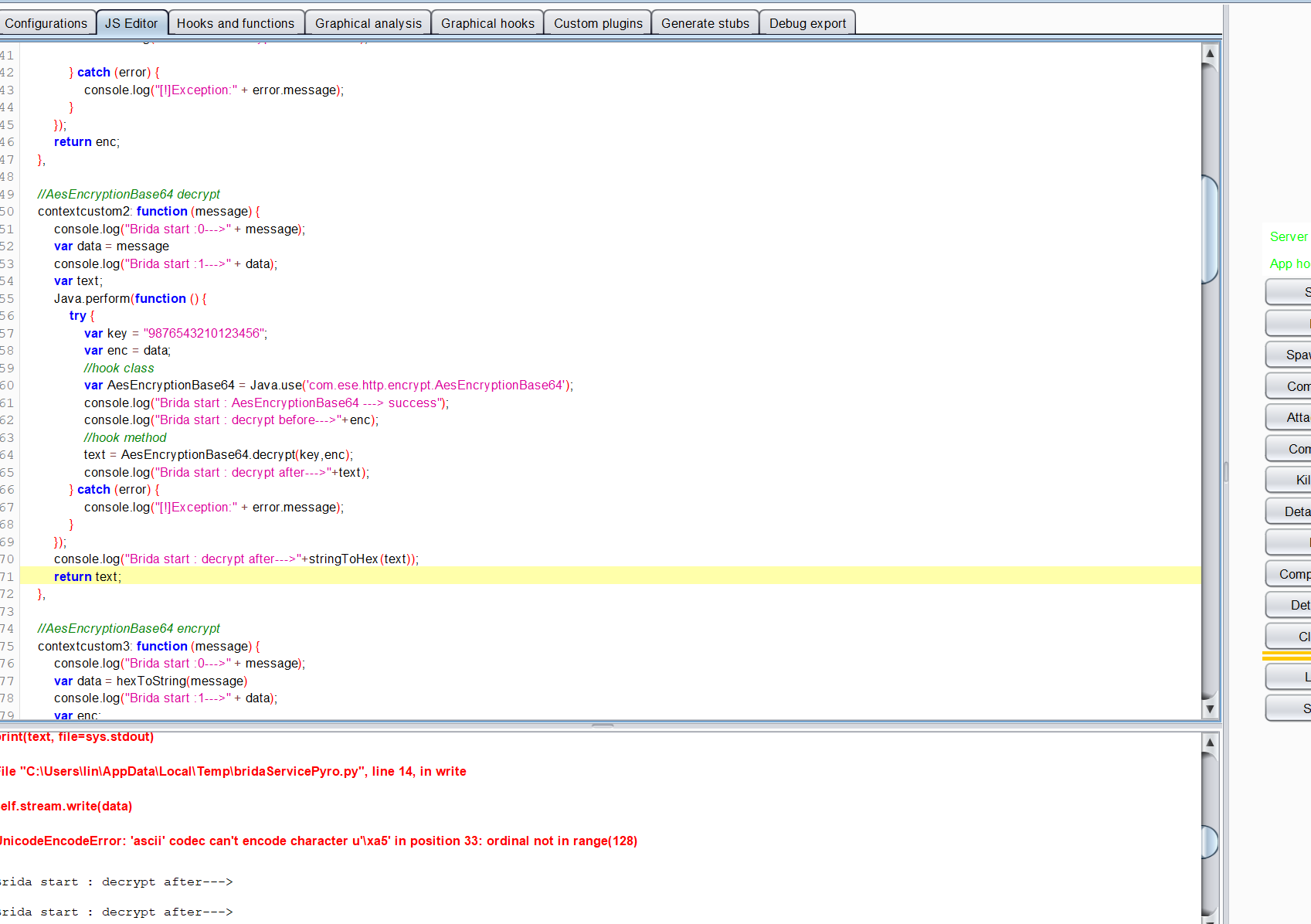
使用大佬的js代码进行调试：

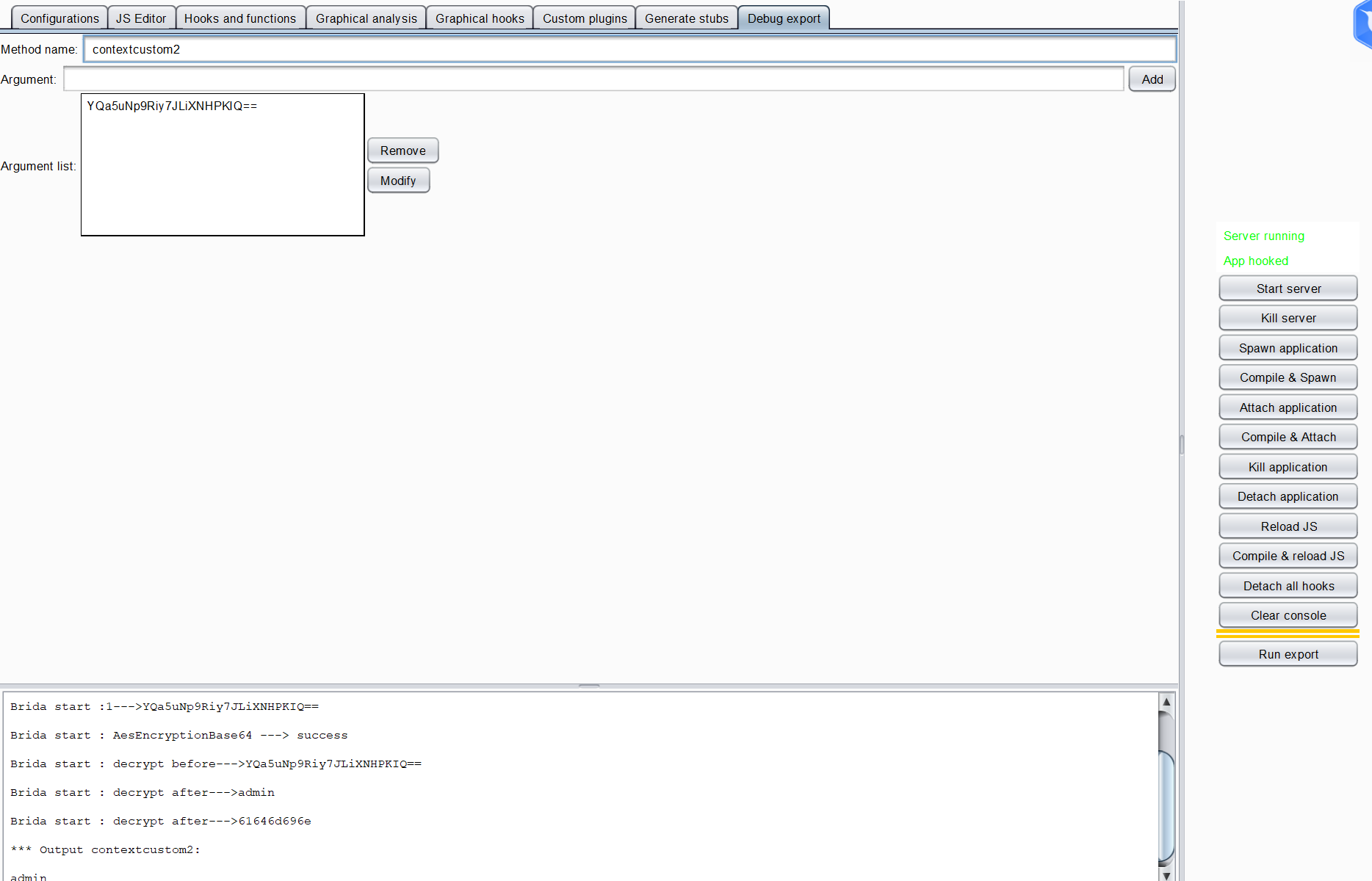


查看输出，发现有问题，查看js代码发现该js传输的为16进制byte流数据，更改代码使之传输字符串数据查看结果：



同理修改解密脚本：





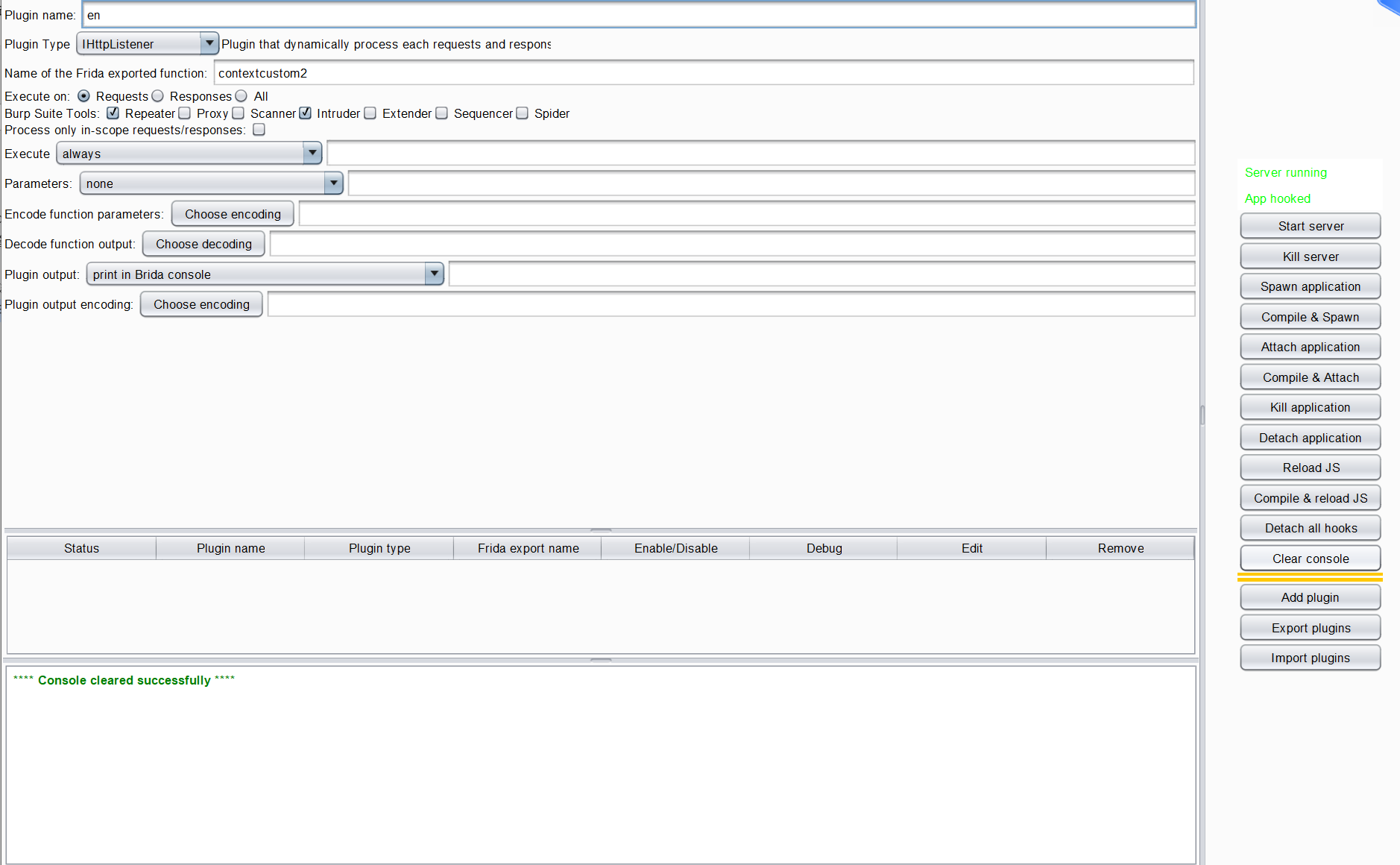
按理说大佬的js应该是没有问题的，不清楚是调用还是brida更新后机制改变了，有兴趣的可以研究下，然后希望有知道原因的大佬指点下本萌新。

## 插件使用

Bridav0.4直接附带有几个利用脚本的插件，挺方便的，也简单的用下插件，入个门。

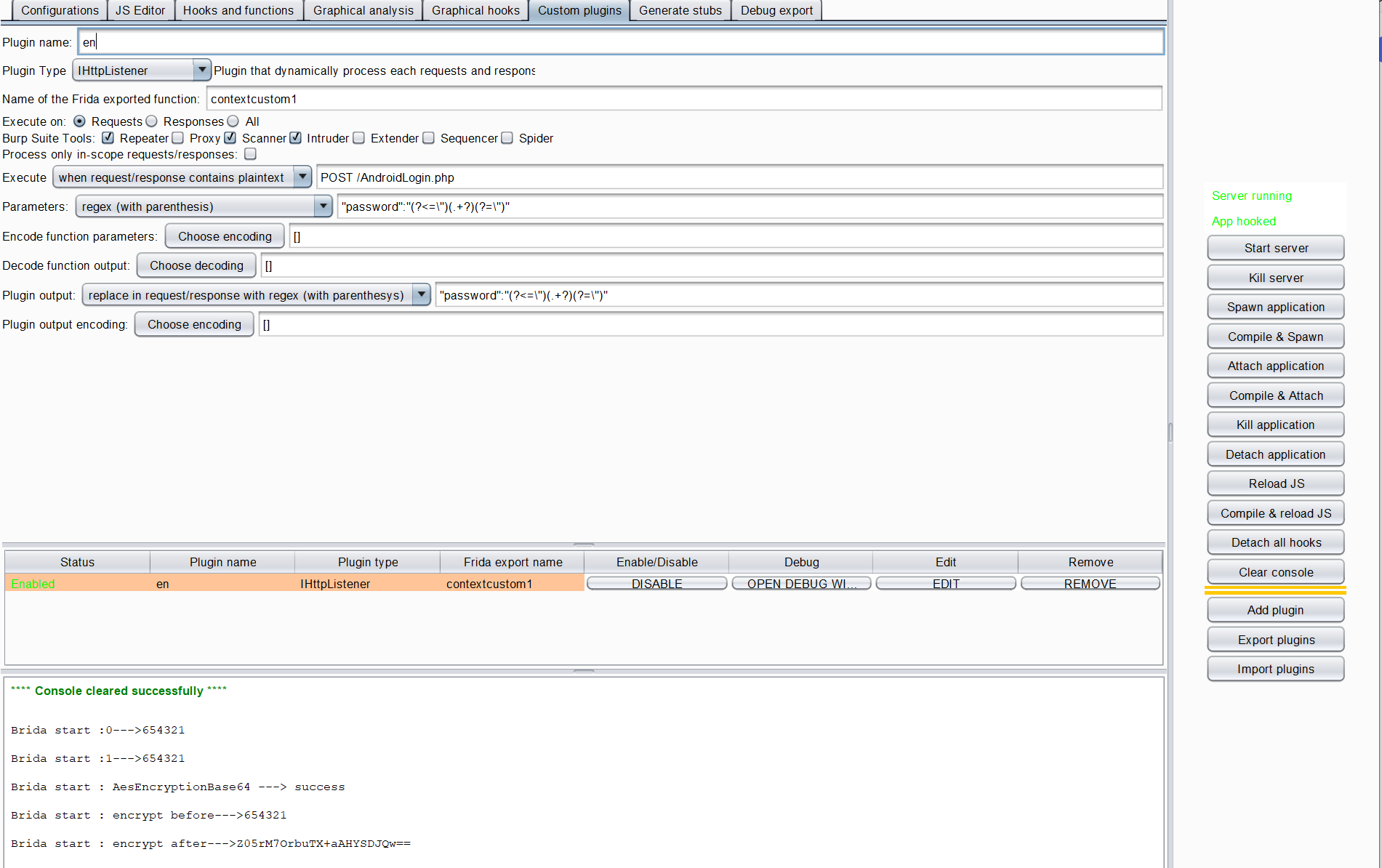
### IHttpListener

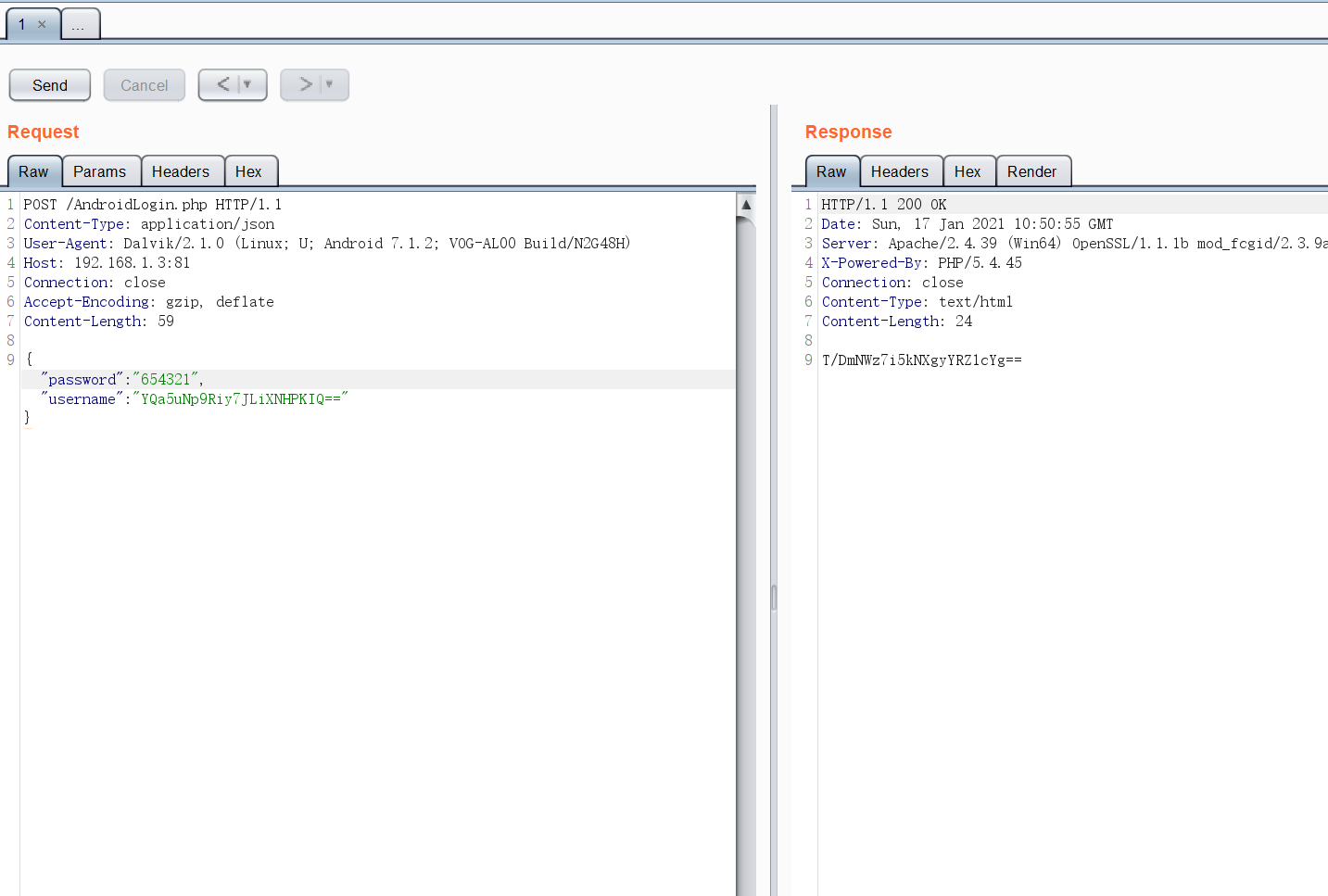
IhttpListener插件是根据条件自动调用js加解的插件：

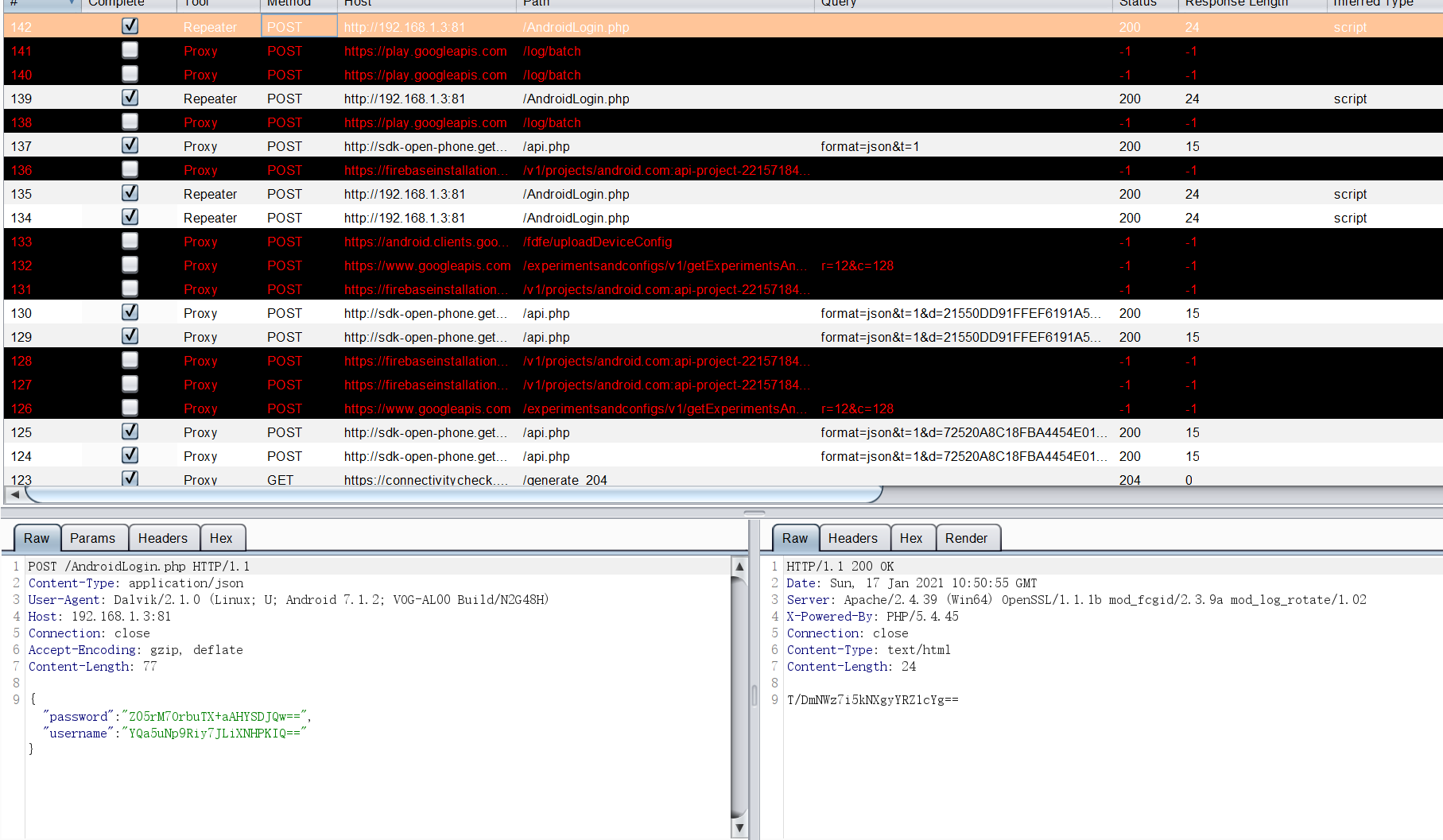


下列是该插件各项代表的意义：

* 插件名称：BridaEncryptPassword
* 插件类型：IHttpListener
* Frida导出函数的名称：（我们在前面的步骤中定义的JS函数的名称。**请勿**在导出的函数名称中**使用大写字符**）
* 执行于：**请求**/**响应**
* Burp Suite工具：**Repeater**，**Intruder**和**Scanner**（为了能够手动测试*密码*参数，使用Intruder执行蛮力并使用Scanner进行扫描）
* 仅处理范围内的请求/响应：**是**（为了避免对意外的请求进行加密）
* 执行：（我们可以给插件一个正则表达式或检查的所有请求/响应选择是否插件应该启用纯文本字符串，为了只能在指定的请求/响应上运行我们的插件。在我们的情况下，我们仅对登录请求感兴趣）
* 参数：**正则表达式**。（我们可以定义一个正则表达式，可以包括使用正则表达式组参数的任意数量的在我们的插件，我们想要传递的参数传递给弗里达导出函数只密码字段并且我们的正则表达式可以完成此任务； Brida提供了许多不同的选项来传递参数，例如完整的请求/响应，正文，标头，带有弹出窗口的动态Feed，...）
* 编码函数参数：**无**（在将提取的参数发送到Frida导出函数之前，我们可以对其进行编码，但是在这种情况下没有必要。这对于二进制输入非常重要：在这种情况下，最好使用ASCII-HEX编码参数或Base64并在Frida导出函数中对其进行解码。单击时，自定义插件的所有编码/解码选项都会打开一个弹出式窗口，您可以在其中选择一种或多种编码/压缩算法，例如Base64，ASCII-HEX，URL， GZIP，...）
* 解码函数输出：**无**（我们可以解码Frida返回的输出。至于参数，如果输出为二进制形式，则最好使用Frida导出函数对其进行编码，并使用此选项在插件中对其进行解码。例如，我们用于插件的iOS函数*+ cryptoPassword：*返回Base64输出。后端等待使用相同函数编码的加密数据，因此我们不需要对其进行解码）
* 插件输出：**用正则表达式（带括号）**（我们将Frida返回的加密值替换为他的未加密形式，由我们在Repeater或Burp在“入侵者/扫描仪”中插入。至于参数，我们使用REGEX组来选择插入点）
* 插件输出编码：（此菜单允许我们在将Frida输出插入请求/响应之前对其进行编码。）



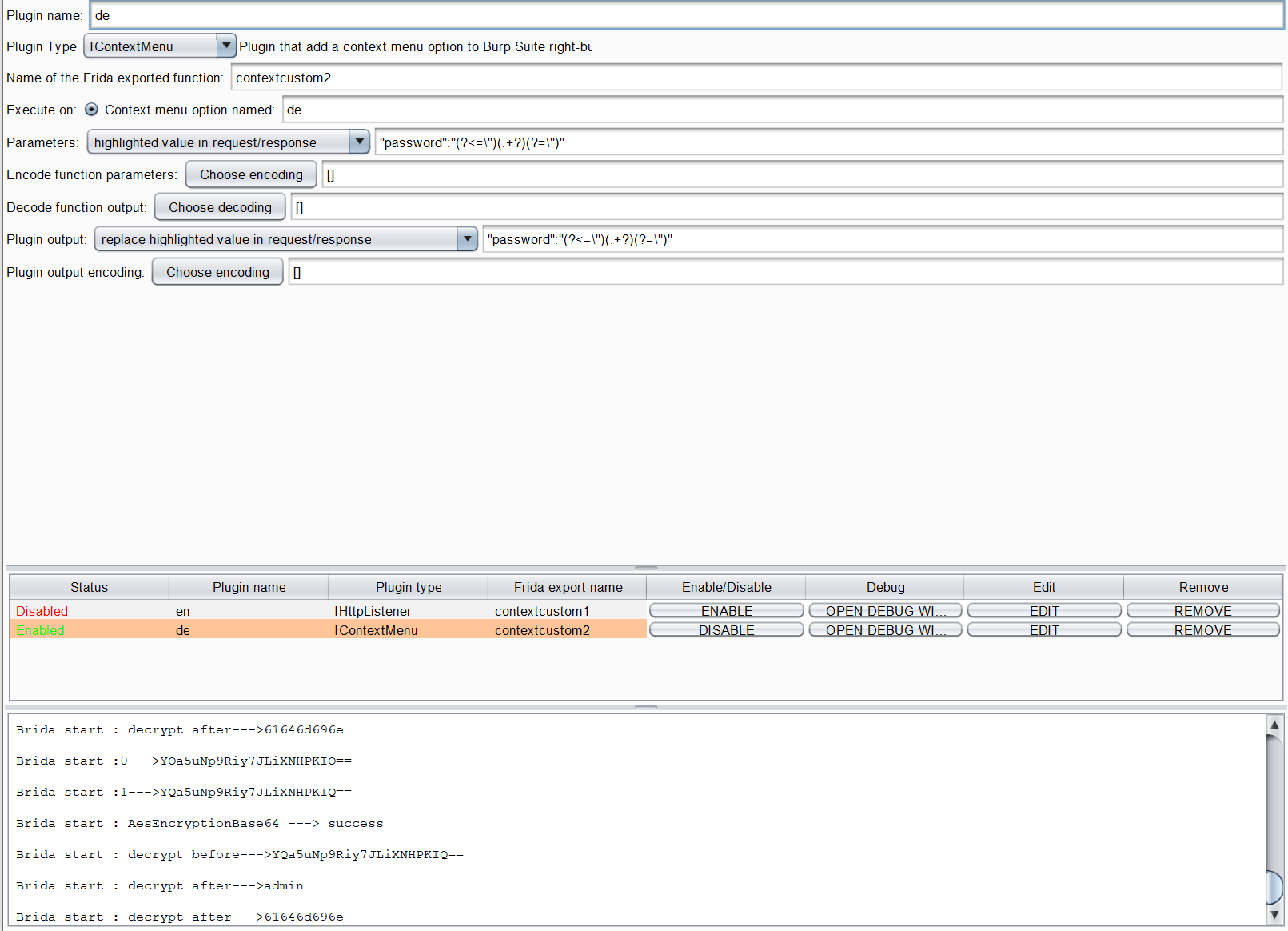




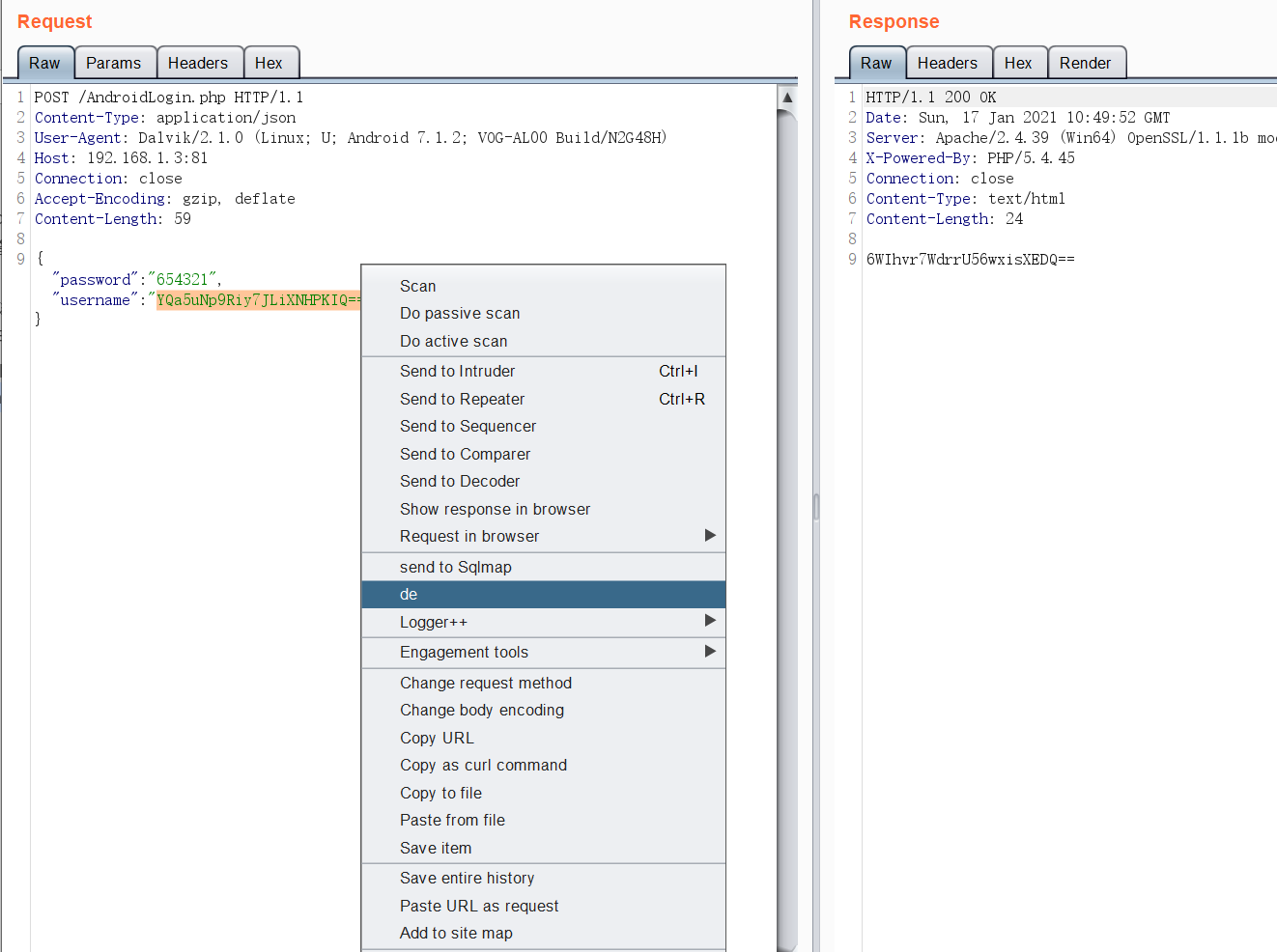
正则等规则在这个插件中尤其重要，匹配错误的话就会失误。

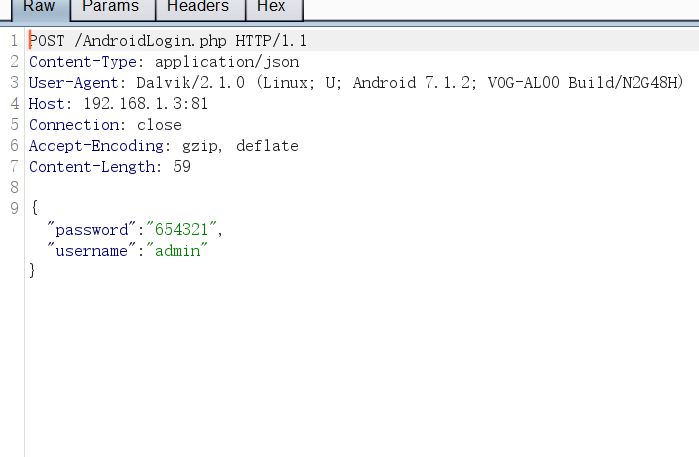
### IcontextMenu

IcontextMenu该插件是添加单击右键可以加解密选项功能的插件：



* 插件名称：
* 插件类型：IContextMenu
* Frida导出函数的名称：（我们在前面的步骤中定义的JS函数的名称。**请勿在导出的功能名称中使用大写字符**）
* 执行上：右键解密功能的名称
* 参数：**请求/响应中突出显示的值**（通过单击并拖动鼠标左键突出显示的请求/响应部分，将作为参数提供给Frida导出函数。Brida提供了许多其他不同的选项来传递参数，例如完整的请求/响应，带组的正则表达式，正文，标题，带弹出窗口的动态Feed，...）
* 编码函数参数：**无**（我们可以在将参数发送到移动应用程序之前对其进行编码，但是在这种情况下没有必要。这对于二进制输入非常重要：在这种情况下，最好使用ASCII-HEX或Base64对参数进行编码并在Frida导出的函数中对其进行解码。单击时，“自定义插件”的所有编码/解码选项均会打开一个弹出窗口，您可以在其中选择一种或多种编码/压缩算法，例如Base64，ASCII-HEX，URL，GZIP ...）
* 解码函数输出：**无**（我们可以解码Frida返回的输出。至于参数，如果输出为二进制形式，则最好使用Frida导出函数对其进行编码，并使用此选项在插件中对其进行解码。在我们的插件中，我们用于插件的iOS函数*+ cryptoResponse：，*以ASCII字符串返回搜索操作的结果，因此我们不需要对其进行解码）
* 插件输出：**在请求/响应中替换突出显示的值**（Frida导出功能的输出将代替作为参数提供的突出显示的值）
* 插件输出编码：**无**（此菜单允许我们在将Frida输出插入所选位置之前对其进行编码。但是，在我们当前的插件中，这不是必需的，我们将其设置为*none*）





Emm，还有两个插件类型，有兴趣的可以去了解下，在brida的GitHub上有详细的教程我也是参考其wiki学习的：<https://github.com/federicodotta/Brida/wiki>

## 参考

<https://github.com/federicodotta/Brida/wiki----->《birda的wiki》

<https://blog.csdn.net/weixin_39190897/article/details/102691898--->《移动安全-Brida加解密数据》