Wp

1. 前言

阅读参考文章可以知道这是一个flash的类型混淆的漏洞，根据题目的要求可以知道是复现这个漏洞，也就是做到类型混淆这一步。我们可以选择复现这个漏洞，也可以不用。允许多解的原因就是降低一下难度，有精力的朋友可以按照题目要求去做，做完相信你能收获不少；不过只为拿到flag的话，这题是可以不用复现，另外解出来的。

当然，本题设计的初衷是为了让大家体验一下分析真实漏洞的情形，但怕大家误会说我们是来骗大家的wp 和 poc的，所以设计题目的时候，不用复现漏洞也能拿到flag，而且最后提交的时候只用提交自己的大概解法和思路，也不用交Poc。

由于特殊原因，就不提供分析的WP了。

这里提供最快取得flag的方法，不用复现这个漏洞。

1. 审题

查看题目要求与flag的组成，关键部分就是class\_6 这个类的字节码，

与 ExternalInterface.call("alert","Congratulations to you !"); 这句话对应的字节码。

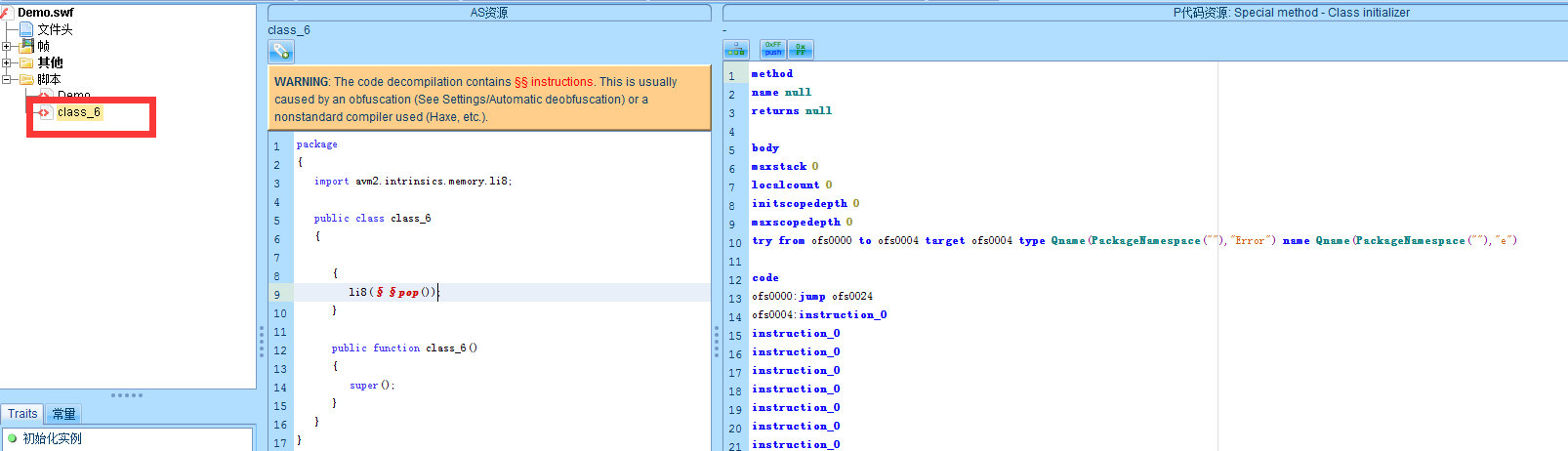
只要还原出这2个部分，就可以拿到flag，不需要去复现。

1. 分析Demo.swf

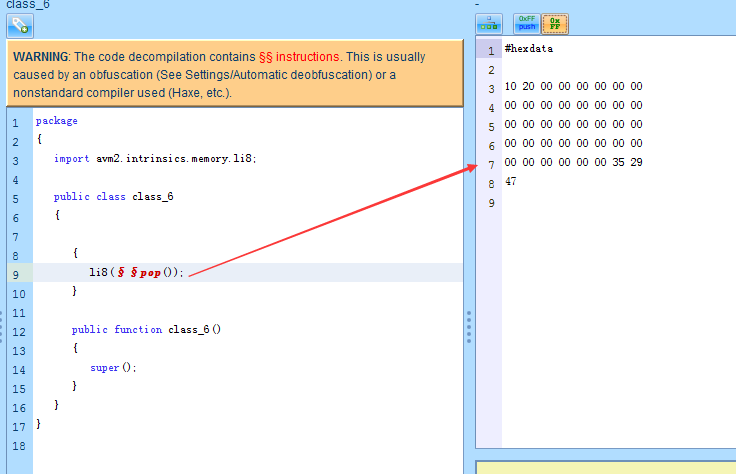
根据题目链接提供的文件可知有个 Demo.swf文件。首先我们去百度搜索反编译swf的工具，找到一个名叫 JPEXS 的工具，安装一下。



安装好之后，反编译提供的demo，可以看到有个class\_6的类



我们去看flag组成部分的字节码：

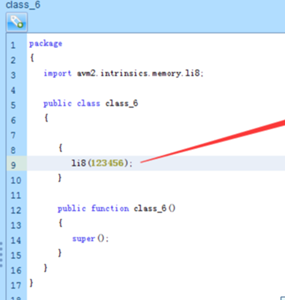


最后9个字节是：

00 00 00 00 00 00 35 29 47

这与题目描述中提供的class\_6的图片存在不同点。也就是li8() 这个地方。

题目描述中这里是 li8(123456)



所以很显然，这最后9个字节：00 00 00 00 00 00 35 29 47 应该不是真正的flag组成部分。

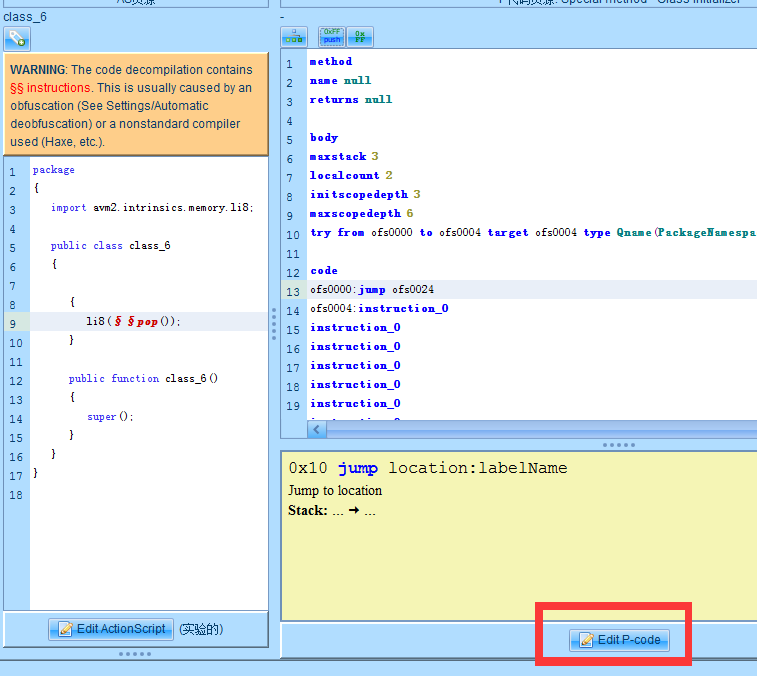
我们需要还原这个Demo中的class\_6类。

根据题目提示的参考文章：<https://www.freebuf.com/vuls/174206.html>

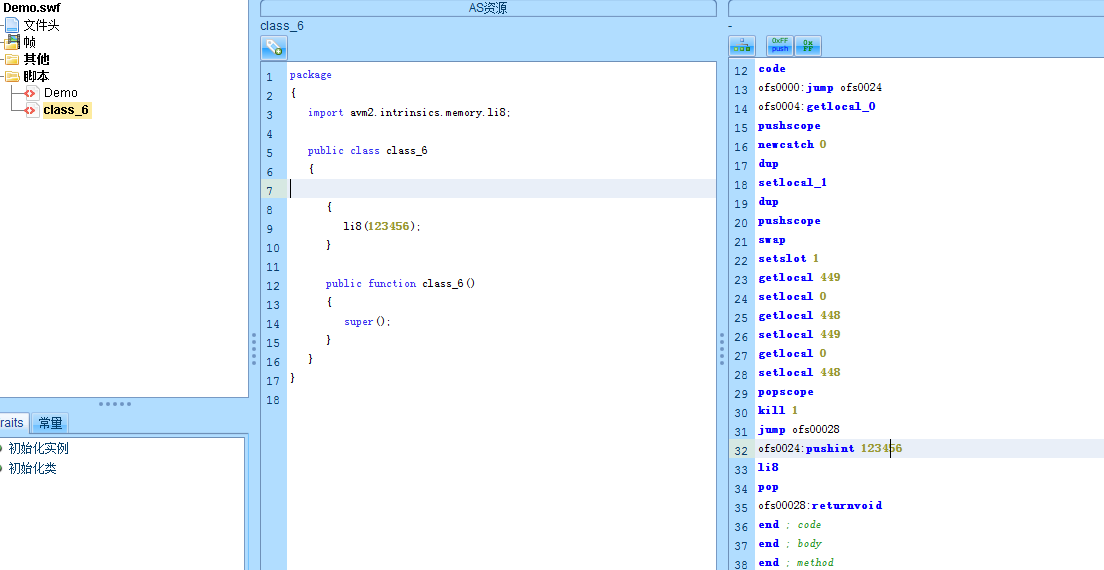
该文章中有一幅图说明了class\_6 出应该填写什么数据，即下图：



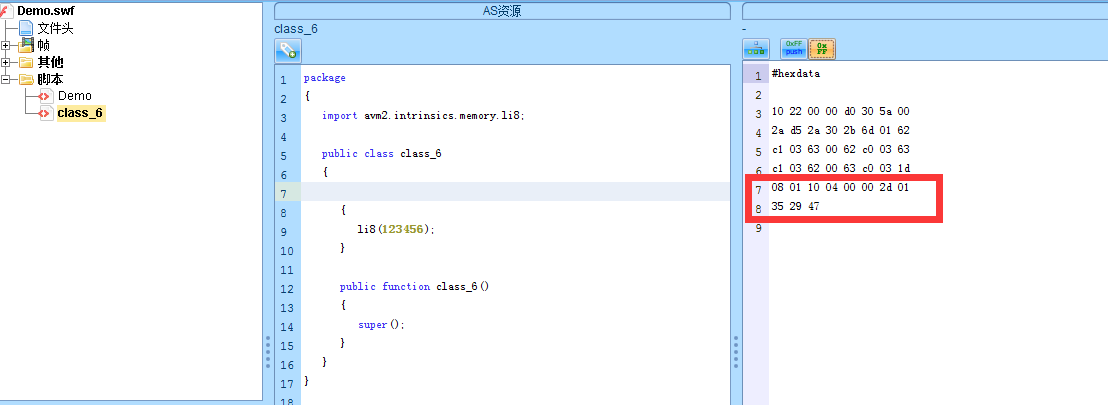
我们依照上图来修改Demo.swf中的class\_6类：



JPEXS提供了修改代码的功能，点击上图红框圈起来的 Edit P-code 进行修改，修改完成之后如下图：



那么去查看Li8(123456)出的字节码：



取到最后9个字节：10 04 00 00 2d 01 35 29 47 ，这就是flag的其中一部分。

1. 写ActionScript代码

那么，flag就剩下另一部分了：

ExternalInterface.call("alert","Congratulations to you !");

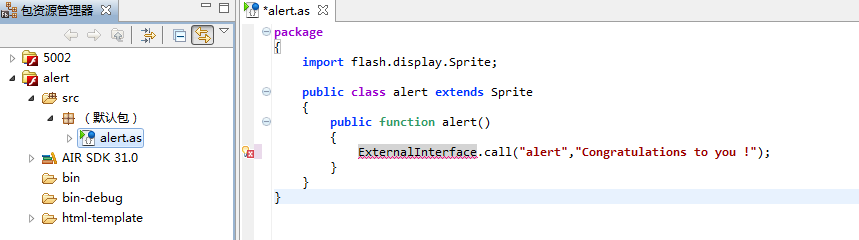
这句话的字节码。

这里存在2种方法：

* 找一个有使用ExternalInterface.call 函数的swf，用JPEXS 反编译去取得字节码。
* 手动编写 actionScript的代码，再用JPEXS 反编译去取得字节码。。

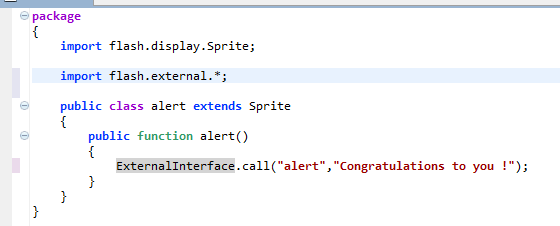
为了确保我们取得的字节码确实 alert弹出出Congratulations to you ! 的情况，这里选择第二种方法—写代码。

首先，搭建ActionScript的编译环境，可安装Flash builder 4.7。安装好之后，创建工程，写入代码：

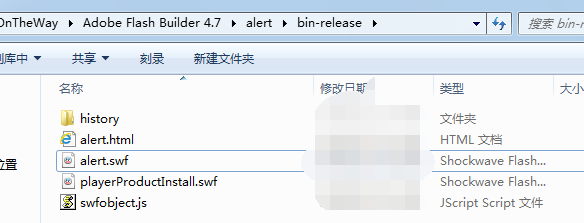


可以看到有个错误，这个错误是缺少包，使用谷歌或者百度即可解决这个问题。

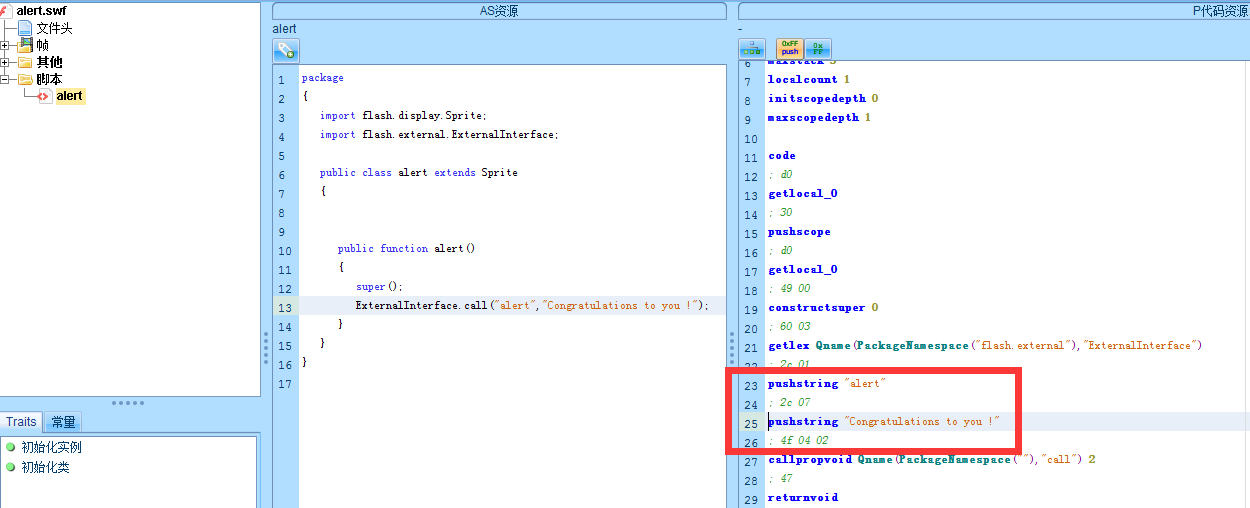
如下：



编译成功后，会在bin-release 文件夹下生成 swf 文件。



使用 JPEXS 打开 alert.swf ，取得字节码：



取最后3个字节： 4f 04 02 。

那么最终拿到flag ： D0g3{4f0402100400002d01352947}