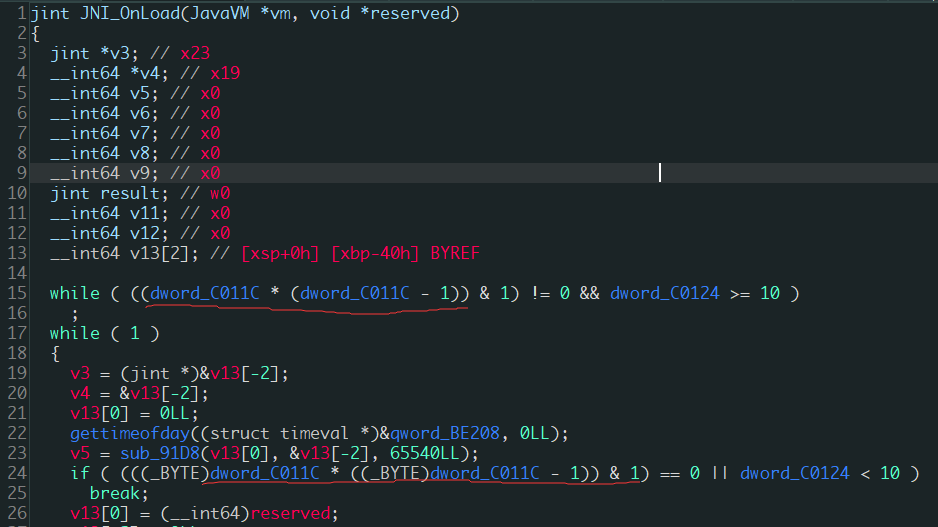
# 百度加固

<https://bbs.pediy.com/thread-271388.htm>

https://bbs.pediy.com/thread-257926-1.htm

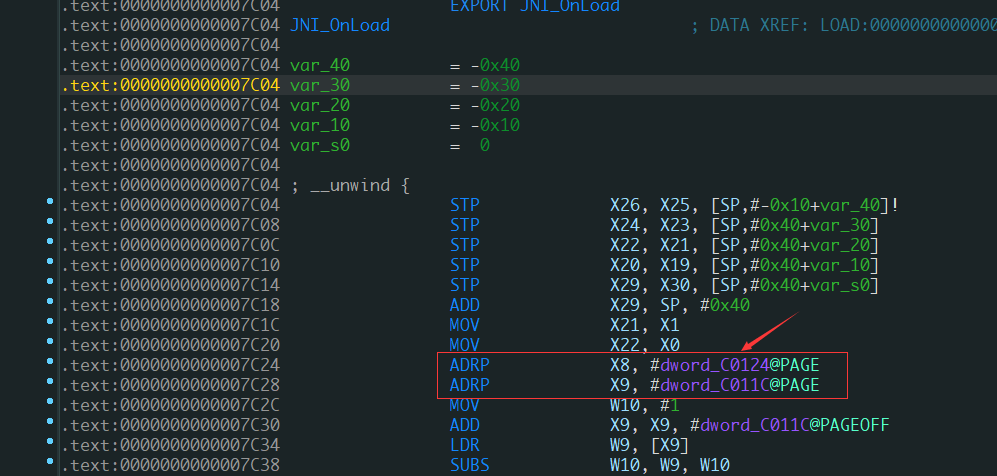
## 判断条件永真或者永假

第一个例子是百度的加固



上图中可以明显的看出垃圾代码，x\*(x-1)恒为偶数。其实这种运算是可以利用IDA自己优化的，主要是通过把计算的x从bss的未初始化变量，指向一个具体值。

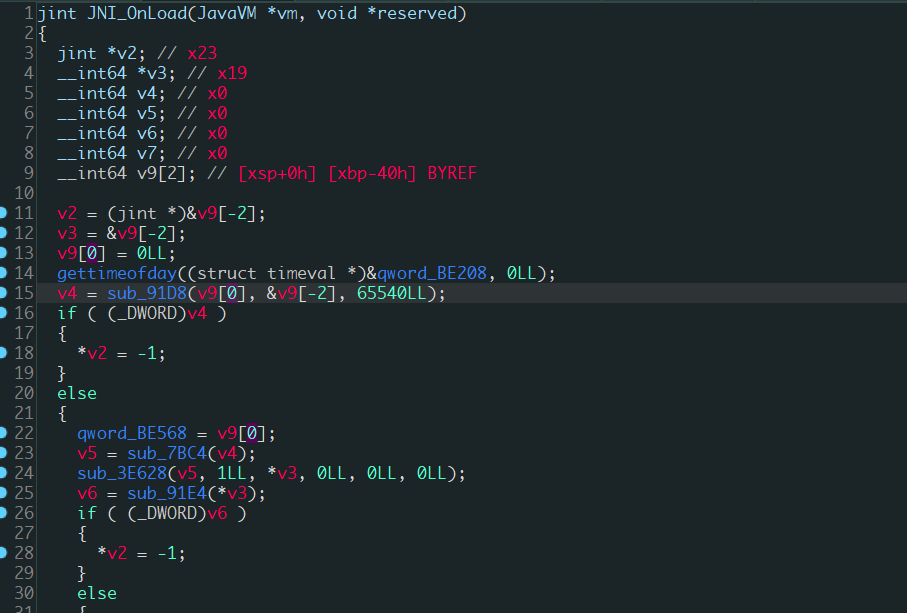
我们这里把如下代码随意指向一个地址或者直接给相应的寄存器赋值



下面是patch脚本



Patch之后重新F5，可以看到垃圾代码已经被消除

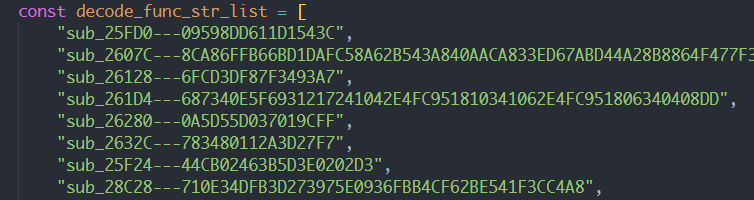


## 字符串加密

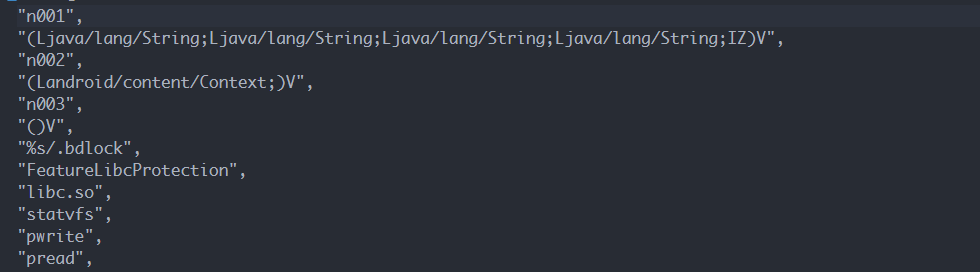
百度加固中大部分字符串都通过加密存储，我们可以通过脚本的方式将这些字符串和相应的解密函数给对应起来，再通过frida主动调用去拿到解密后的字符串，再通过脚本写成备注的形式方便静态分析。



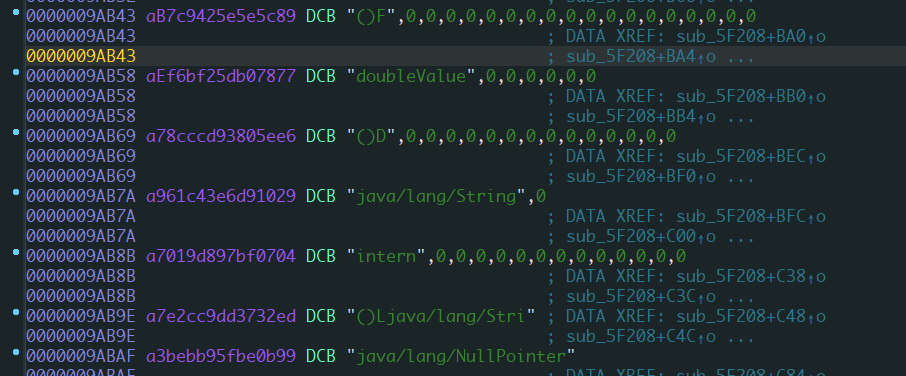
首先写idapython脚本，dump对应关系

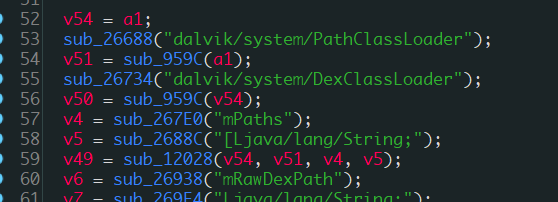


再通过frida主动调用，拿到对应的解密字符串



Patch到so之后

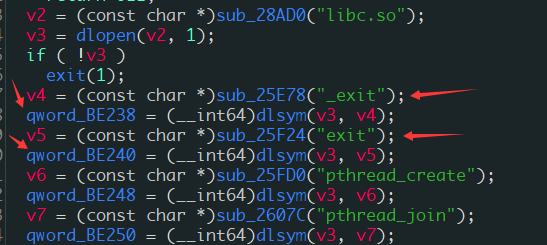






## Libc函数动态获取

在patch完上述的字符串之后，发现sub\_B3B4，动态加载了一系列的libc函数，存在变量中。



我们将相关的变量patch成libc函数的名称，方便我们的分析

先将F5过后的伪代码复制出来，之后通过解析相应的函数名，使用set\_name来将变量地址patch成相应的libc函数名



完成后效果

