原文地址:http://drops.wooyun.org/tools/738

0x00 闲扯

好吧继上一篇文章之后，就没发文章了！(其实是一直在写但是写的很少还凑不起一篇文章而已)

但是这几天对插件进行了一定的改良了 因为在自己在实际的XSS过程中也发现了自己的插件 还不够强大！

不能够百分之百的满足自己的需求！所以就根据自己平常的需求给加了上去！

我想做到玩XSS一个工具即可解决需求！所以感觉即使是现在的插件也还有很大的不足！

所以很希望得到你们的意见 东凑一块 西凑一块写成一个真正的一个插件解决需求！

(现在想给hook生成功能块加一个 是否自动把hook转换成短链接！但是。。。在技术上有点问题)

另外还有什么其他的功能 或者一些比较猥琐的小技巧 希望大家提出来 我加上。。。

比如在chrome里隐藏payload 。。。我一直在想如何才能构造一段被chrome认为是无效的字符。。并且还能够执行！

0x01 界面以及功能介绍

兼容性没做所以在不同的分辨率下会乱码。。。这个不会做

分辨率：1280 X 800 (求教)

编码

请把需要编码的字符放在左边的输入框内 Encode / Decode

然后点击中间上方的Encode按钮 选择你要编码的类型 然后点击相应的按钮即可！

经过相应编码后的内容便会在右边的Output框中！

新功能介绍

进制编码常规变异：

进制编码包括：

```
html编码的十进制编码
html编码的十六进制编码
javascript的十六进制编码
javascript的八进制编码
```

进制编码常规变异的功能：

会给编码前面的数字多加7个0，因为IE对进制编码加0，只识别到八个0，多了的话就认为这不是个有效的值了！

也有很多程序过滤规则也是这样写的！他们会把你变异了的值给解析回来，然后再判断是不是危险字符！

适用场景：

当进制编码被解析回来，再次过滤了的时候，比如 &#60 在过滤程序中被还原回来再次过滤了！

但是&#0060没有在过滤程序中被还原回来，但是在页面中被浏览器被解析还原了，那么就可以用进制编码的常规变异！

进制编码非常规变异：

进制编码非常规变异功能：

会给编码的数字前面多加10个0！原因同上！

适用场景：

当进制编码被解析回来，再次过滤了的时候，比如 &#60 或者 &#000000060都被还原回来，再次过滤的话，那么便可以用非常规变异！

IE识别到8个0 可是chrome能识别到更多的0！很多过滤程序都是根据IE的8个0来写的！所以更多的0 也是一种绕过方式！

使用心得：以上的功能都是自己亲身经历到的 当时是某GOV的站 在这里贴出payload吧 以及笔记

```
search?str=xxxx%3Ca%20href=%22data:text/html;%26%23000000000000000000098ase64%26%230000000000000044%20PGltZyBzcmM9eCBvbmVycm9yPWFsZXJ0KDEpPg==%22%3Etest%3C/a%3E
```

缺陷参数：str

过滤规则够BT 但是同样能绕！

会把提交的编码 给解码 然后再插入到网页中！然后再对网页内的值进行检查 过滤！

base64 --> %26%23000000000000000000098ase64

首先%26%23会被还原成 &# 于是变成了： &#00000000098 于是被还原成：b b插入到页面 再检查 b+ase64 = base64 满足规则 于是又过滤成 base64 绕过失败！

但是如果b的html十进制编码 再多加几个000 便不会被他给解码 但是在浏览器中又会被解码 于是便可以这样绕过！

html编码去分号：

(此选项可配合其他选项一起使用 比如进行html编码时 勾选常规变异 + 去html编码分号)

ps:小伙伴们勿淘气别选了 常规变异 又勾选非常规变异。。。 那你到底是要闹哪样！

还有勾选编码时 也勾选了 hook生成的话 那么我不知道你要闹那样。。

如果要对生成的钩子进行编码的话 那么就弄两次吧 一次：生成 复制 ，二次： 粘贴 编码！

使用场景：

html实体编码的分号在大部分情况下都是可以去掉的 能减少输入字符！我有强迫症。。

我一般选择html实体编码的时候 都会必勾选！

&#URL编码：

这个对我来说真的是经常用到！比如在测试反射型XSS 以及 DOM XSS时！

因为&#在url中都有特殊的含义 我们很多时候都是把他们当做一个html实体编码表示的方式而已！

可是浏览器不会这样认为，& 会被认为是参数的分隔符 比如一个url:

```
http://xssec.net/?x=1&c=2&d=4
```

如果我们在url上写&号是会被当成参数分隔符的 进行一下url编码就号了！

#号呢，就是location.hash获取的值以及什么的 所以这种字符还是url编码的好~

比如以下这个payload:

```
search.php?searchfield=xsser%c0\%22%20onfocus=%26%2397%26%23108%26%23101%26%23114%26%23116%26%2340%26%2347%26%2374%26%23105%26%23110%26%2347%26%2341%20autofocus//&imag
```

大家还原下编码就能看到原本的字符了！

Hook生成

基本介绍 ：把你的钩子(hook)放入到左边的Encode里 然后勾选你要进行生成的hook类型！然后点后面的生成就好了！

每次请选择一种，别淘气 我没写太多的判断js。。。 精力有限！

下面贴几个演示吧：

在钩子生成方面做的不是特别好，也是自己经验不足的原因 大家有更猥琐的 加载钩子的payload 求pm加上去！

下载链接：

xss-encode_20131115121612.crx_.zip

联系方式：

mail:root@xssec.net

http://t.qq.com/Ox_Jin