

## 信息安全实验二 XSS

### 1. 获得 admin 的密码

根据 SQL 注入实验得到的散列值：

5f4dcc3b5aa765d61d8327deb882cf99，进行 MD5 解密后即可得到

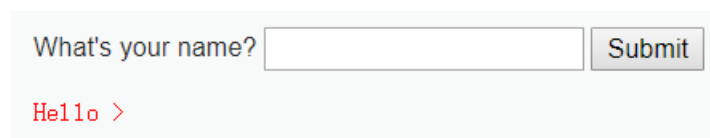
密码为：password，登陆即可。

### 2. High 难度下的漏洞利用进行 alert 弹窗

根据助教演示的 easy 和 medium 难度的 js 弹窗，首先进行尝试输入 js

语句看是否能够弹窗：<script>alert(\xss\)<\script>

结果为：



然后再看看 php 的源代码：

```
$name = preg_replace( '/<(.*s(.*)c(.*)r(.*)i(.*)p(.*)t/i', '', $_GET[ 'name' ] );
```

其对 script 语句进行了完善的正则表达式匹配，无论是按照之前的大小写

还是中间插一个 script 都会被正则表达式匹配并且替换为空。

所以显然用 script 标签无法绕过这个障碍，考虑到最终目的是执行 alert

能够弹窗，而其他标签也可以触发 alert，所以可以用其他标签引起一个错

误从而触发调用 alert 方法，选用 img 标签，然后用一个非法的 src 这样

在有错误的情况下即可实现弹窗：<img src=null onerror=alert(/xss/)>


执行结果如下：

ilities/xss\_r/?name=<img+src%3Dnull+onerror%3Dalert%28%2F%29>#

github uestc ML reference 来自 202.38.79.49 /xss/ 确定

## Vulnerability: Reflected Cross Site Scripting

What's your name?

Hello 

### More Information

- [https://www.owasp.org/index.php/Cross-site\\_Scripting\\_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- [https://www.owasp.org/index.php/XSS\\_Filter\\_Evasion\\_Cheat\\_Sheet](https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet)
- [https://en.wikipedia.org/wiki/Cross-site\\_scripting](https://en.wikipedia.org/wiki/Cross-site_scripting)
- <http://www.cgisecurity.com/xss-faq.html>
- <http://www.scriptalert1.com/>

Home  
Instructions  
Setup / Reset DB  
Brute Force  
Command Injection  
CSRF  
File Inclusion  
File Upload  
Insecure CAPTCHA  
SQL Injection  
SQL Injection (Blind)  
Weak Session IDs  
XSS (DOM)  
**XSS (Reflected)**  
XSS (Stored)  
DVWA Security  
PHP Info  
About  
Logout

Username: admin  
Security Level: high

### 3. HW2——button

#### a) Low

利用 html 语言输入以下内容：

```
<input type="button" onclick="location.href='http://baidu.com';" value="Baidu"/>
```

可以直接显示结果：并且按下 baidu 的按钮会跳转到 baidu 的首页。

What's your name?

Hello

## b) Medium

由于只涉及对 script 的过滤处理，所以与 low 难度相同，也是输入

```
<input type="button" onclick="location.href='http://baidu.com';" value="Baidu"/>
```

可以实现输出跳转按钮的目的。

## c) High

也是只涉及到对 script 的正则匹配，无影响，同上。

The screenshot shows the DVWA web application interface. On the left is a sidebar with a list of security challenges: Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected) (highlighted in green), XSS (Stored), DVWA Security, PHP Info, About, and Logout. The main content area displays a form titled 'What's your name?' with an input field and a 'Submit' button. Below the form, the output shows 'Hello' followed by a button labeled 'Baidu'. Underneath the form is a section titled 'More Information' containing a list of links: [https://www.owasp.org/index.php/Cross-site\\_Scriptin](https://www.owasp.org/index.php/Cross-site_Scriptin), [https://www.owasp.org/index.php/XSS\\_Filter\\_Evasior](https://www.owasp.org/index.php/XSS_Filter_Evasior), [https://en.wikipedia.org/wiki/Cross-site\\_scripting](https://en.wikipedia.org/wiki/Cross-site_scripting), <http://www.cgisecurity.com/xss-faq.html>, and <http://www.scriptalert1.com/>. At the bottom left, the user status is shown as 'Username: admin' and 'Security Level: high'.

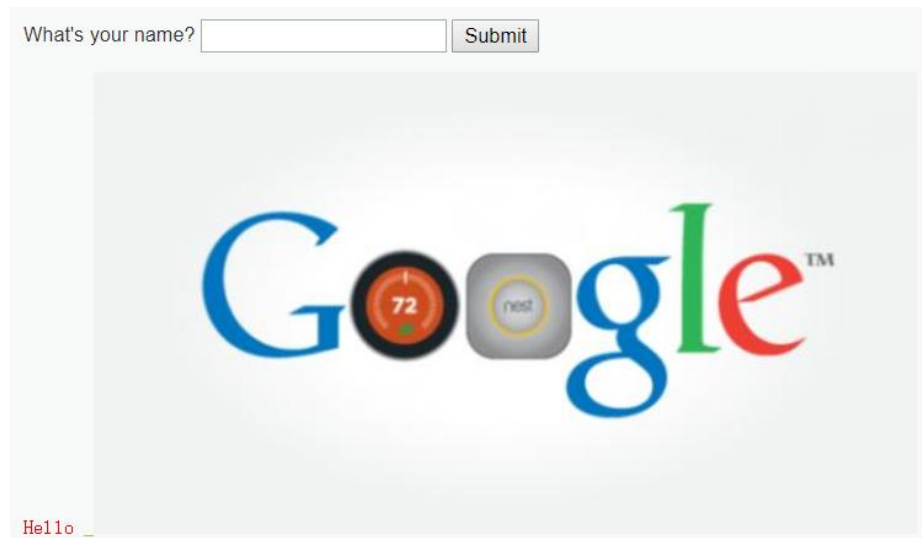
## 4. HW3——img

### a) Low

直接输入显示图片的标签语句：

```
<a href=http://google.com>  </a>
```

可以直接显示出图片，并且可以点击图片直接跳转到谷歌首页。



b) Medium

c) High

Medium 和 high 难度下方法也同上。结果为：

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)


DVWA Security

PHP Info

About

Logout

What's your name?



More Information

- [https://www.owasp.org/index.php/Cross-site\\_Scripting\\_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- [https://www.owasp.org/index.php/XSS\\_Filter\\_Evasion\\_Cheat\\_Sheet](https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet)
- [https://en.wikipedia.org/wiki/Cross-site\\_scripting](https://en.wikipedia.org/wiki/Cross-site_scripting)
- <http://www.cgisecurity.com/xss-faq.html>
- <http://www.scriptalert1.com/>

View Source View Help

Username: admin  
Security Level: high

## 6. XSS(Stored)

### a) Low

首先在两个输入框中尝试输入 script 标签尝试直接进行弹窗，发现并不能输进去，因为设置了输入长度的限制。所以审查网页元素查看其网页结构：

```
▼<tbody>
  ▼<tr>
    <td width="100">Name *</td>
    ▼<td>
      <input name="txtName" type="text" size="30"
        maxlength="10">
    </td>
  </tr>
  ▼<tr>
    <td width="100">Message *</td>
    ▼<td>
      <textarea name="mtxMessage" cols="40" rows="3"
        maxlength="10"></textarea>
    </td>
  </tr>
```

可以发现有个 maxlength 的属性对输入长度进行了设置，但是通过 php 源代码中并没有看到提交到服务器过程的长度限制，所以直接在网页上修改 maxlength，

```
▼<tbody>
  ▼<tr>
    <td width="100">Name *</td>
    ▼<td>
      <input name="txtName" type="text" size="30"
        maxlength="100">
    </td>
  </tr>
  ▼<tr>
    <td width="100">Message *</td>
    ▼<td>
      <textarea name="mtxMessage" cols="40" rows="3"
        maxlength="100"></textarea>
    </td>
  </tr>
```

都修改为 100 后，就可以突破 10 的长度显示输入 html 语句进行弹窗显示，查看 php 代码发现并未对 script 标签作出限制，所以输入 `<script>alert(\PB15111662\)<\script>` 即可。



确定

## Vulnerability: Stored Cross Site Scripting (XSS)

Name \*

Message \*

Sign Guestbook

Clear Guestbook