Web服务器安全方案[原创＋总结]

Web服务器安全方案
对于web服务器的安全，大的方面主要存在于两个方面：

1：系统本身的安全
2：web程序的安全

下面我具体的说说这两个方面的问题：
首先是系统本身的安全，我详细的说一下。
由于使用windows系统的人比较多，所以相对来说，该系统的漏洞也就会比较多，同时懂得一些简单攻击的用户也比较多。我们需要做到如下方面：
1：给系统打上最新的所有补丁，这个可以去microsoft
网站去更新系统，注意一定要打上所有的补丁。
2：尽量给系统安装一个防火墙，这里我推荐一款防火墙，是俄国人写的，获得了国际大奖，具体她如何好我就不介绍了，名称为blackice（黑冰）.她有两种版本，server的和普通pc的。如果对于服务器，推荐我们使用server的版本。
3：尽量给系统安装一个好的杀毒软件，避免一些简单的病毒的入侵，当然说实话，厉害的病毒杀毒软件是没有用处的，只能避免简单的而已，我们真正依赖的其实还是一个好的防火墙。这里瑞星正版的软件相对来说还是比较好的，至于国外的卡巴斯基还是非常出色的。
4：停止server服务，该服务提供 RPC
支持、文件、打印以及命名管道共享。（net stop server,在"开始"-"管理工具"-"服务"中 把server服务停止并改为手动或禁止）,单独删除共享的方法：net share ipc$ /delete
5：用net
share命令确认默认的共享（ipc$,c$,d$...admin$,system$）已经删除，查看手工设置的共享，删除不需要的共享，最好全部删除
6：停止TCP/IP Services服务，该服务支持以下 TCP/IP 服务：Character Generator, Daytime, Discard, Echo, 以及 Quote of the
Day。，它对应多个端口，如7，9，17等，可能导致DOS（DDOS）攻击。
7：用net
start命令查看启动的服务，确认停止所有不必要的服务，特别是：停止telnet，ftp服务等。
8：用net user和net
localgroup命令查看异常的用户和本地组，删除或禁用不必要的帐号,guest ,iusr_hostname等
9：用扫描工具检查开放的可疑端口，查找木马。（这里我们可以使用一些软件比如开放端口查看软件来查看本服务器所有开放的端口。软件名称为：Active ports。还有可以使用一些扫描工具，比如superscan,或者更加强大的GFI Languard scan 软件来查找本服务器的一些弱的漏洞还有端口。）
10：禁止一般用户从网络访问计算机，在管理工具-本地安全策略-
用户权利指派-拒绝从网络登陆计算机中设置。

11：帐号尽可能少，且尽可能少用来登录；
说明：网站帐号一般只用来做系统维护，多余的帐号一个也不要，因为多一个帐号就会多一份被攻破的危险。
12：除过Administrator外，有必要再增加一个属于管理员组的帐号；
说明：两个管理员组的帐号，一方面防止管理员一旦忘记一个帐号的口令还有一个备用帐号；另方面，一旦黑客攻破一个帐号并更改口令，我们还有机会重新在短期内取得控制权。
13：将Administrator重命名，改为一个不易猜的名字。其他一般帐号也应尊循着一原则。（可以在安全策略中进行设置修改）

14：将Guest帐号禁用，同时重命名为一个复杂的名字，增加口令，并将它从Guest组删掉；
说明：有的黑客工具正是利用了guest 的弱点，可以将帐号从一般用户提升到管理员组。
15：给所有用户帐号一个复杂的口令（系统帐号出外），长度最少在8位以上，
且必须同时包含字母、数字、特殊字符。同时不要使用大家熟悉的单词（如microsoft）、熟悉的键盘顺序（如qwert）、熟悉的数字（如2000）等。
说明：口令是黑客攻击的重点，口令一旦被突破也就无任何系统安全可言了，而这往往是不少网管所忽视的地方，据我们的测试，仅字母加数字的5位口令在几分钟内就会被攻破，而所推荐的方案则要安全的多。
16：在帐号属性中设立锁定次数，比如改帐号失败登录次数超过5次即锁定改帐号。这样可以防止某些大规模的登录尝试，同时也使管理员对该帐号提高警惕。（也是在安全策略中修改，具体的自己查资料，这篇文章就不介绍了）
17：加强日志审核；
说明：日志任何包括事件查看器中的应用、系统、安全日志，IIS中的WWW、SMTP、FTP日志、SQL
SERVER日志等，从中可以看出某些攻击迹象，因此每天查看日志是保证系统安全的必不可少的环节。安全日志缺省是不记录，帐号审核可以从域用户管理器——规则——
审核中选择指标；NTFS中对文件的审核从资源管理器中选取。要注意的一点是，只需选取你真正关心的指标就可以了，如果全选，则记录数目太大，反而不利于分析；另外太多对系统资源也是一种浪费。
18：加强数据备份；
说明：这一点非常重要，站点的核心是数据，数据一旦遭到破坏后果不堪设想，而这往往是黑客们真正关心的东西；遗憾的是，不少网管在这一点上作的并不好，不是备份不完全，就是备份不及时。数据备份需要仔细计划，制定出一个策略并作了测试以后才实施，而且随着网站的更新，备份计划也需要不断地调整。
19：只保留TCP/IP协议，删除NETBEUI、IPX/SPX协议；
说明：网站需要的通讯协议只有TCP/IP，而NETBEUI是一个只能用于局域网的协议，IPX/SPX是面临淘汰的协议，放在网站上没有任何用处，反而会被某些黑客工具利用。
20：停掉没有用的服务，只保留与网站有关的服务和服务器某些必须的服务。

说明：有些服务比如RAS服务、Spooler服务等会给黑客带来可乘之机，如果确实没有用处建议禁止掉，同时也能节约一些系统资源。但要注意有些服务是操作系统必须的服务，建议在停掉前查阅帮助文档并首先在测试服务器上作一下测试。
21：隐藏上次登录用户名,修改注册表Winnt4.0：
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\Current Version\Winlogon
中增加DontDisplayLastUserName，将其值设为1。Windows2000中该项已经存在，只需将其值改为1。
说明：缺省情况下，上次登录的用户名会出现在登录框中，这就为黑客猜测口令提供了线索，最好的方式就是隐藏上次登录用户名。（同样可以在安全策略中进行配置）
22：关闭并删除默认站点：
默认FTP站点
默认Web站点
管理Web站点
上面的部分资料来自互联网，其余的为本人所写。当然当我们发现机器可能遭到攻击的时候，我们的处理方案我以后写出，这里就不介绍了。

然后就是web程序的安全了。架设在windows下的服务器，这里所讨论的范畴只是IIS，所以不讨论Apache。IIS支持的为asp or asp.net.对于asp.net我不是非常了解，方案我就少说一点。对于asp程序，我具体的说一下。
目前因为很多开源的web网站的代码，如果细心的程序员就会发现那些代码里面存在很多的漏洞，比如简单的注入漏洞，还有一些非常简单的上传功能可以上传脚本文件，然后攻击者可以直接执行这些上传的脚本，达到一些他们的目的。比如添加系统用户，或者删除服务器文件，还有修改主页等等。这些都是非常简单的漏洞，但是确不被一些粗心的coder注意。我们作为网络的管理员，在配置完了一个web以后我们应该做一次入侵检测的，这个有专门的软件，他可以检测到一些简单的注入，或者代码漏洞，还有系统的一些缺陷。具体的软件名称自己去查询一下。
下面我具体的说说一些需要注意的问题了（对于一些比较早的漏洞我就不说了，现在的系统大部分都不存在这些漏洞了）：
1：ACCESS mdb 数据库有可能被下载的漏洞
问题描述：

在用ACCESS做后台数据库时，如果有人通过各种方法知道或者猜到了服务器的ACCESS数据库的路径和数据库名称，那么他能够下载这个ACCESS数据库文件，这是非常危险的。比如：如果你的ACCESS数据库book.mdb放在虚拟目录下的database目录下，那么有人在浏览器中打入：
http://someurl/database/book.mdb

如果你的book.mdb数据库没有事先加密的话，那book.mdb中所有重要的数据都掌握在别人的手中。

解决方法：

(1)

为你的数据库文件名称起个复杂的非常规的名字，并把他放在非常规目录下。所谓"非常规"。打个比方：比如有个数据库要保存的是有关书籍的信息，可不要把他起个"book.mdb"的名字，起个怪怪的名称，比如d34ksfslf.mdb，再把他放在如./kdslf/i44/studi/

的几层目录下，这样黑客要想通过猜的方式得到你的ACCESS数据库文件就难上加难了

(2)不要把数据库名写在程序中。有些人喜欢把DSN写在程序中，比如：

```
DBPath = Server.MapPath("cmddb.mdb")
conn.Open "driver={Microsoft Access Driver (*.mdb)};dbq=" & DBPath
```

假如万一给人拿到了源程序，你的ACCESS数据库的名字就一览无余。因此建议你在ODBC里设置数据源，再在程序中这样写：

```
conn.open "shujiyuan"
```

(3)使用ACCESS来为数据库文件编码及加密。首先在选取"工具->安 全->加密/解密数据库，选取数据库（如：employer.mdb），然后接确定，接着会出现"数据库加密后另存为"的窗口，存为：employer1.mdb。接着employer.mdb就会被编码，然后存为employer1.mdb.

要注意的是，以上的动作并不是对数据库设置密码，而只是对数据库文件加以编码，目的是为了防止他人使用别的工具来查看数据库文件的内容。

接下来我们为数据库加密，首先以打开经过编码了的employer1.mdb，在打开时，选择"独占"方式。然后选取功能表的"工具->安全->设置数据库密码"，接着输入密码即可。

为employer1.mdb设置密码之后，接下来如果再使用ACCEES数据库文件时，则ACCESS会先要求输入密码，验证正确后才能够启动数据库。

不过要在ASP程序中的connection对象的open方法中增加PWD的参数即可，例如：

```
param="driver={Microsoft Access Driver (*.mdb)};Pwd=yfdsfs"
param=param&";dbq="&server.mappath("employer1.mdb")
conn.open param
```

这样即使他人得到了employer1.mdb文件，没有密码他是无法看到employer1.mdb的。

2：ASP程序密码验证漏洞

漏洞描述：

很多网站把密码放到数据库中，在登陆验证中用以下sql,(以asp为例）

```
sql="select * from user where username='"&username&"'and pass='"& pass &'"
```

此时，您只要根据sql构造一个特殊的用户名和密码，如：ben' or '1'='1

，就可以进入本来你没有特权的页面。再来看看上面那个语句吧：
sql="select * from user where username='"&username&"'and pass='"&
pass&"'
　　此时，您只要根据sql构造一个特殊的用户名和密码，如：ben' or '1'='1
这样,程序将会变成这样: sql="select*from username where
username="&ben'or'1'='1&"and pass="&pass&"
or
是一个逻辑运算符,作用是在判断两个条件的时候,只要其中一个条件成立,那么等式将会成立.而在语言中,是以1来代表真的(成立).那么在这行语句中,原语句的"and"验证将不再继续,而因为"1=1"和"or"令语句返回为真值.。
　　另外我们也可以构造以下的用户名：
username='aa' or username<>'aa'
pass='aa' or pass<>'aa'
　　相应的在浏览器端的用户名框内写入：aa' or username<>'aa
口令框内写入：aa' or
pass<>'aa,注意这两个字符串两头是没有'的。这样就可以成功的骗过系统而进入。

后一种方法理论虽然如此，但要实践是非常困难的，下面两个条件都必须具备。
　　1.
你首先要能够准确的知道系统在表中是用哪两个字段存储用户名和口令的，只有这样你才能准确的构造出这个进攻性的字符串。实际上这是很难猜中的。
　　2.系统对你输入的字符串不进行有效性检查。
　　问题解决和建议：
　　对输入的内容验证和"'"号的处理。
这个有点类似sql注入了。
3：IIS4或者IIS5中安装有INDEX SERVER服务会漏洞ASP源程序
　　问题描述：
　　在运行IIS4或者IIS5的Index
Server，输入特殊的字符格式可以看到ASP源程序或者其它页面的程序。甚至以及添打了最近关于参看源代码的补丁程序的系统，或者没有.htw文件的系统，一样存在该问题。获得asp程序，甚至global.asa文件的源代码，无疑对系统是一个非常重大的安全隐患。往往这些代码中包含了用户密码和ID，以及数据库的源路径和名称等等。这对于攻击者收集系统信息，进行下一步的入侵都是非常重要的。
通过构建下面的特殊程序可以参看该程序源代码：
http://sourceurl/null.htw?CiWebHitsFile=/default.asp&CiRestriction=none&CiHiliteType=Full

这样只是返回一些html格式的文件代码，但是当你添加%20到CiWebHitsFile的参数后面，如下:
http://someurl/null.htw?CiWebHitsFile=/default.asp%20&CiRestriction=none&CiHiliteType=Full

这将获得该程序的源代码。

（注意：/default.asp是以web的根开始计算。如某站点的http:///welcome/welcome.asp 那么对应就是：
http://someurl/null.htw?CiWebHitsFile=/welcome/welcome.asp%20&CiRestriction=none&CiHiliteType=Full
）

由于'null.htw'文件并非真正的系统映射文件，所以只是一个储存在系统内存中的虚拟文件。哪怕你已经从你的系统中删除了所有的真实的.htw文件，但是由于对null.htw文件的请求默认是由webhits.dll来处理。所以，IIS仍然收到该漏洞的威胁。

问题解决或者建议：

如果该webhits提供的功能是系统必须的，请下载相应的补丁程序。如果没必要，请用IIS的MMC管理工具简单移除.htw的映象文件。
补丁程序如下：
Index Server 2.0:
Intel:
http://www.microsoft.com/downloads/release.asp?ReleaseID=17727
Alpha:
http://www.microsoft.com/downloads/release.asp?ReleaseID=17728
Indexing Services for Windows 2000:
Intel:
http://www.microsoft.com/downloads/release.asp?ReleaseID=17726

4：具体说说两个如何防止sql注入的方法（代码来自互联网）：

```
"============================
"过滤提交表单中的SQL
"============================
function ForSqlForm()
dim fqys,errc,i,items
dim nothis(18)
nothis(0)="net user"

nothis(1)="xp_cmdshell"

nothis(2)="/add"

nothis(3)="exec%20master.dbo.xp_cmdshell"

nothis(4)="net localgroup administrators"

nothis(5)="select"
```

```
nothis(6)="count"

nothis(7)="asc"

nothis(8)="char"

nothis(9)="mid"

nothis(10)="'"

nothis(11)=":"

nothis(12)="""

nothis(13)="insert"

nothis(14)="delete"

nothis(15)="drop"

nothis(16)="truncate"

nothis(17)="from"

nothis(18)="%"

"nothis(19)="@"

errc=false

for i= 0 to ubound(nothis)
for each items in request.Form
if instr(request.Form(items),nothis(i))<>0 then
response.write("
")
response.write("你所填写的信息:" &
server.HTMLEncode(request.Form(items)) & "
含非法字符:" & nothis(i))
response.write("
")
response.write("对不起,你所填写的信息含非法字符！ 返回")
response.End()
end if
```

```
next
next
end function
"==========================
"过滤查询中的SQL
"==========================
function ForSqlInjection()
dim fqys,errc,i
dim nothis(19)
fqys = request.ServerVariables("QUERY_STRING")
nothis(0)="net user"

nothis(1)="xp_cmdshell"

nothis(2)="/add"

nothis(3)="exec%20master.dbo.xp_cmdshell"

nothis(4)="net localgroup administrators"

nothis(5)="select"

nothis(6)="count"

nothis(7)="asc"

nothis(8)="char"

nothis(9)="mid"

nothis(10)="'"

nothis(11)=":"

nothis(12)="""

nothis(13)="insert"

nothis(14)="delete"

nothis(15)="drop"

nothis(16)="truncate"
```

```
nothis(17)="from"

nothis(18)="%"

nothis(19)="@"

errc=false

for i= 0 to ubound(nothis)

if instr(FQYs,nothis(i))<>0 then

errc=true

end if

next

if errc then
response.write "查询信息含非法字符！返回"
response.end

end if

end function
```