常见Web风险分析及防护实践

2016.12

目录

第一章小节1：web应用被攻击原因分析

- **甲方视角**
- 现在绝大多数的应用系统都已经或准备迁移到B/S环境下；
- B/S系统的数量少点的也有几十套，多点的可达到几百套；
- 不同的B/S系统多数都分散于独立的虚拟机中，但存在一定数量的多B/S系统（二级站点）运行在一台物理服务器；
- 基于需求进行B/S系统的变更时，不会考虑由此而带来的安全隐患；

- **乙方视角**
- 开发B/S系统相对复杂，开发周期长，往往是整体开发团队所有人员参与，以满足甲方需求功能实现、快速业务交付为目标，而这往往会对应用安全的埋下隐患：
  - 开发人员具备的安全规范编程能力不一，漏洞在此产生；
  - 代码测试集缺少代码安全测试集；

- **攻击者视角**
- B/S系统为少有的对外提供服务的展示窗口，
- B/S系统存在很大的攻击价值，例如政治影响、经济利益、个人隐私信息等；
- 重要的一点是发起针对B/S系统攻击的门槛很低，因为互联网上大量的黑客培训、攻击工具泛滥，几分钟的时间就可以掌握进行攻击的相关知识；
- 75%+的攻击事件都是针对B/S系统的，这是我们知道的，实际上这些攻击事件能够产生的安全事故远大于75%；

我们看一下下面关于国内“学校教育”行业网站安全的两张图表：

左图：按存在高危漏洞的网站占比，学校教育类占23%

右图：按被植入后门的网站占比，学校教育类则高达46%

通过这两组数据的对比，我们也可以发现似乎恶意攻击者更青睐于攻击学校教育。

数据于来源2013年360互联网安全中心

第一章小节2：web应用危险漏洞介绍

序号	web漏洞名称	漏洞被利用后的威胁	漏洞严重等级
1.1	SQL注入漏洞	导致数据库被拖库、撞库，敏感信息被泄露、篡改或删除；由此而引发的衍生威胁有账号信息盗取、上传webshell木马到服务器，达到远程控制整个web网站服务，以此为跳板进一步攻击内部网络其他主机	▼▼▼▼▼
1.2	XSS跨站脚本漏洞	可以盗取用户帐户cookie信息，修改用户设置，重定向到黑客的钓鱼站点，可控制用户执行DDOS攻击，利用客户端浏览器的漏洞传播蠕虫等	▼▼▼▼▼
1.3	Struts2框架命令执行漏洞	可以利用此漏洞在系统上执行任意命令，达到远程控制服务器	▼▼▼▼▼
1.4	命令注入漏洞	可以利用此漏洞在系统上执行任意命令，达到远程控制服务器	▼▼▼▼▽
1.5	文件上传漏洞	直接或间接上传webshell木马文件，达到远程控制整个web网站服务，以此为跳板进一步攻击内部网络其他主机。	▼▼▼▼▽

web漏洞原理说明

- **SQL注入漏洞**

攻击者通过构建特殊的SQL语句，将其拼接到Web表单或请求域名URL中，用于欺骗web服务器去执行，进而执行攻击者所要的操作。

- **XSS跨站脚本漏洞**

它指的是恶意攻击者往Web页面里插入恶意html代码，当用户浏览该页或点击网页链接时时，嵌入其中Web里面的html代码会被执行，从而达到恶意用户的特殊目的。

- **Struts2框架命令执行漏洞**

Apache Struts2中WebWork框架使用XWork中的ParametersInterceptor不允许参数名中出现“#”字符，但如果使用了Java的unicode字符串表示\u0023，攻击者就可以绕过保护，修改保护Java方式执行的值。进一步可调用java语句来执行任意命令，甚至控制操作系统。

- **命令注入漏洞**

由于Web应用程序对用户提交的数据过滤不严格，导致黑客可以通过构造特殊命令字符串的方式，将数据提交至Web应用程序中，并利用该方式执行外部程序或系统命令实施攻击。

- **文件上传漏洞**

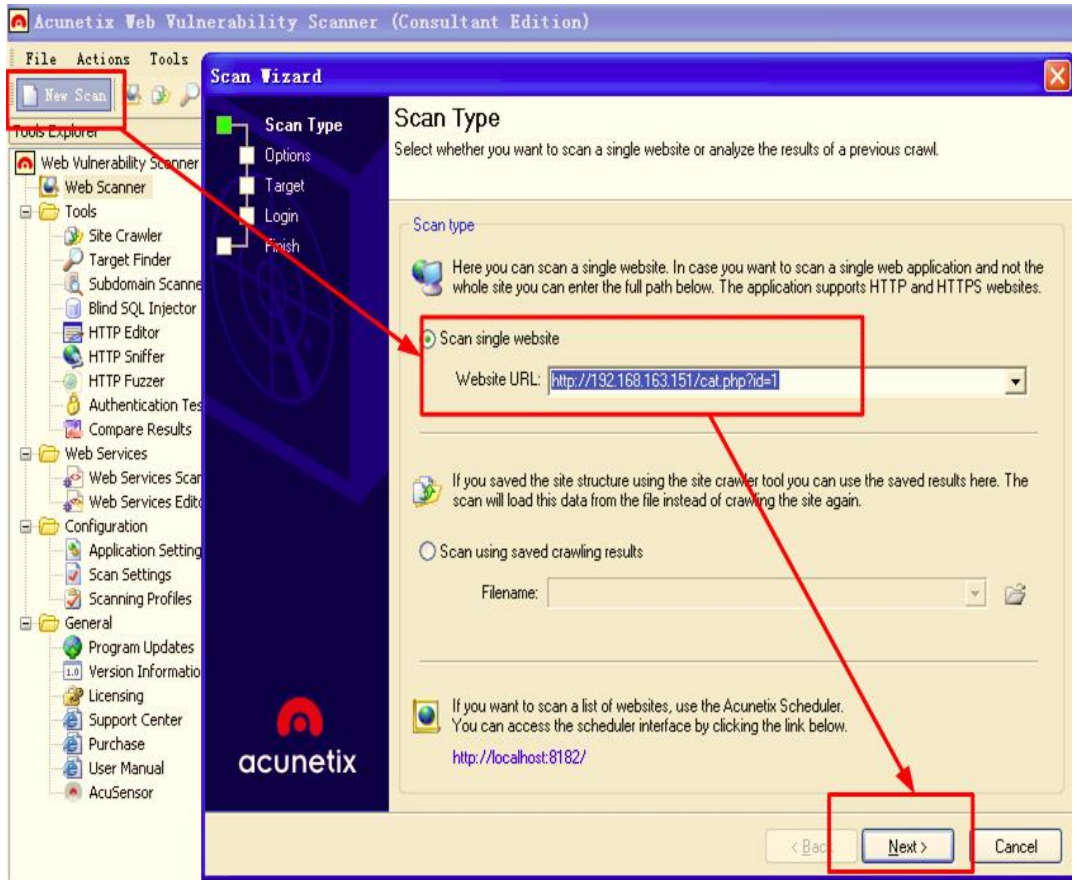
文件上传漏洞指攻击者利用程序缺陷绕过系统对文件的验证与处理策略将恶意程序上传到服务器并获得执行服务器端命令。

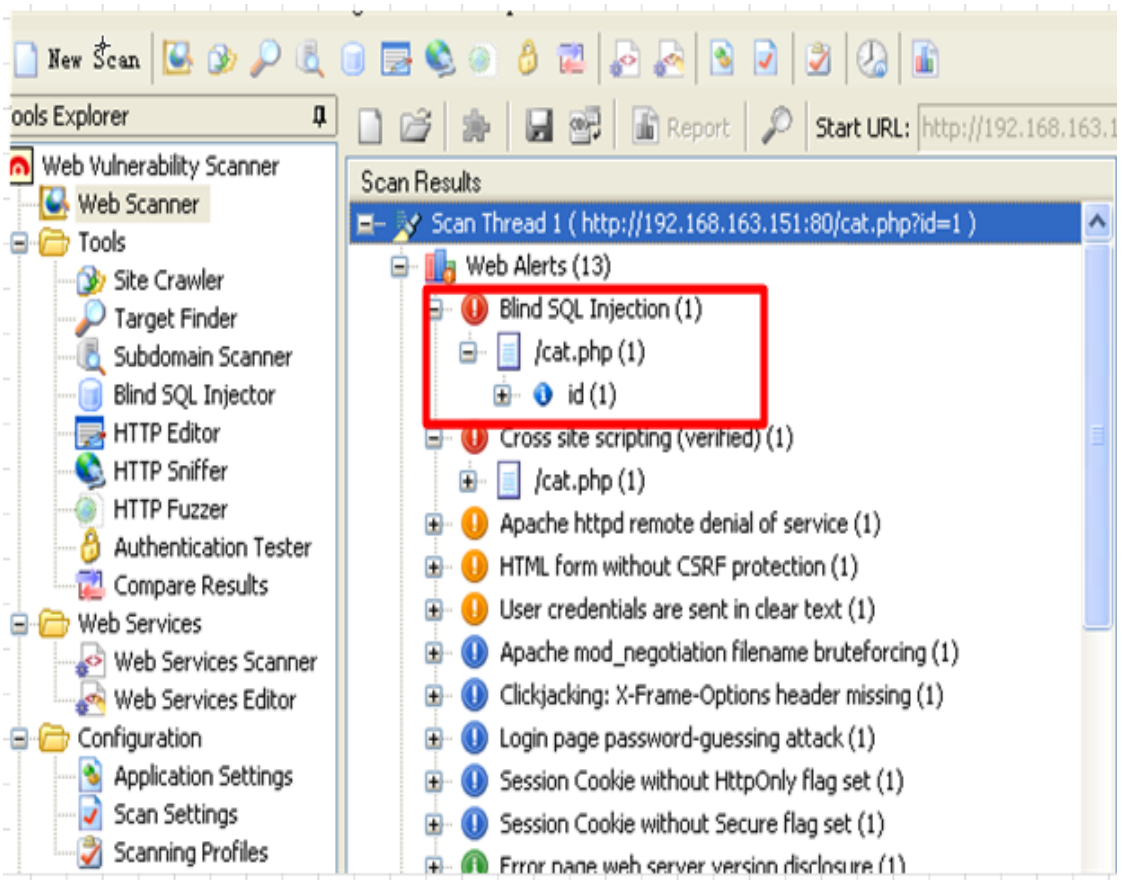
第一章小节3：攻击web应用演示示例1：SQL注入攻击演示

演示URL示例：http://192.168.163.151/cat.php?id=1’

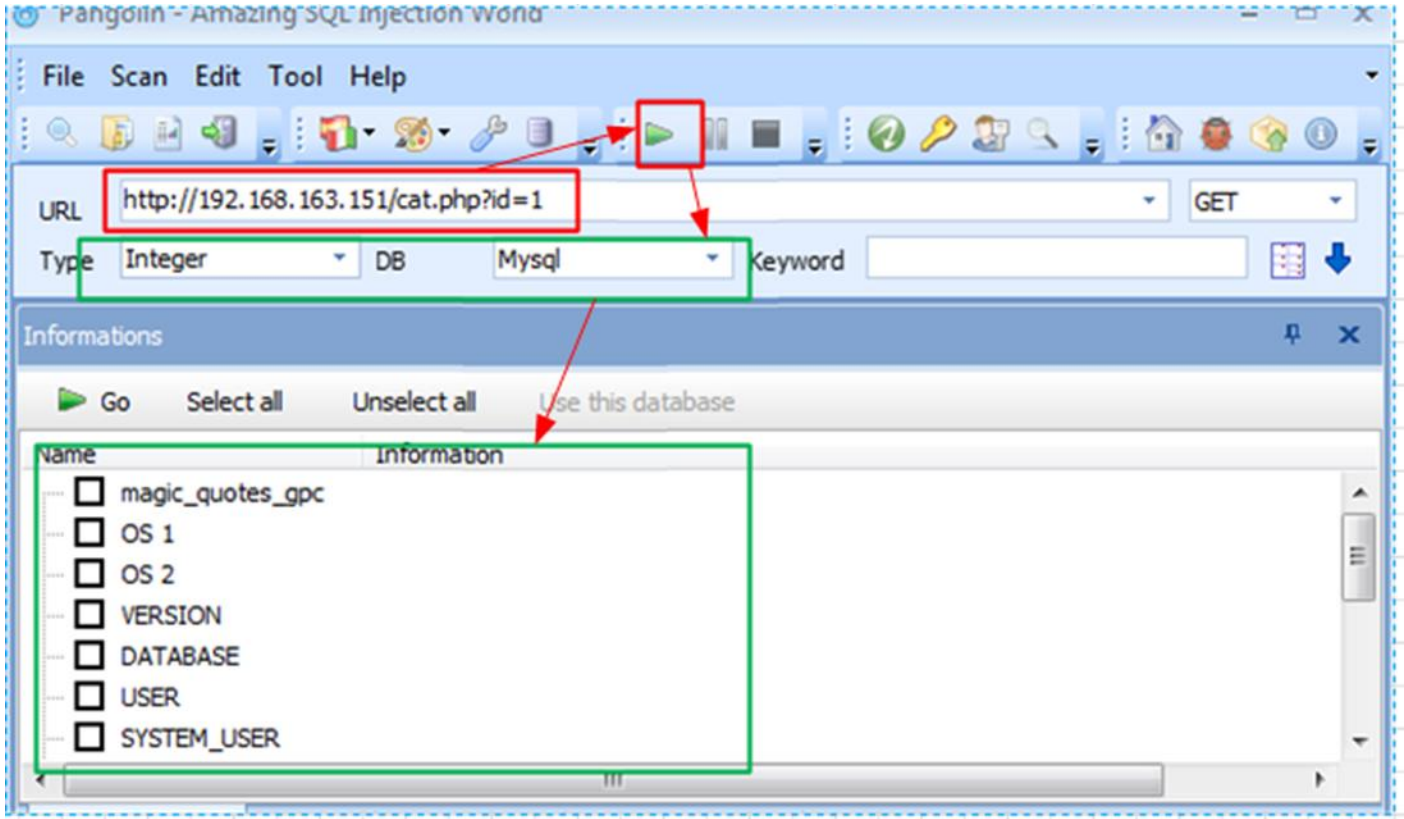
第一步：可以在参数id=1后面加个单引号试试，判断此URL是否存在SQL注入漏洞，结果报错。

第二步：将URL填入web应用漏洞扫描器进行漏洞确认，我们看到的结果是次URL确实存在SQL注入漏洞。



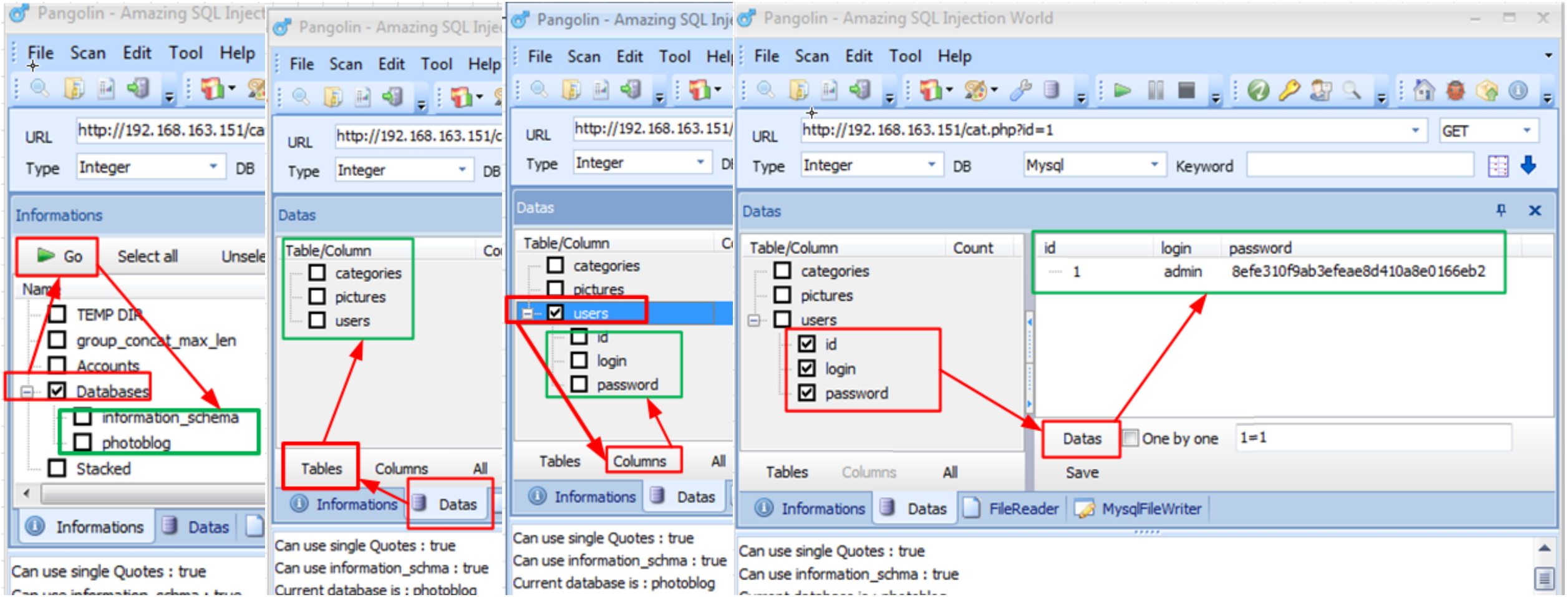


第三步：判断出此URL存在着SQL注入漏洞后，使用注入工具来对注入点进行利用，先爆出后台数据库信息。截图我们可以看到，网站后台数据库是使用的Mysql。



第四步：对数据库进行拖库。

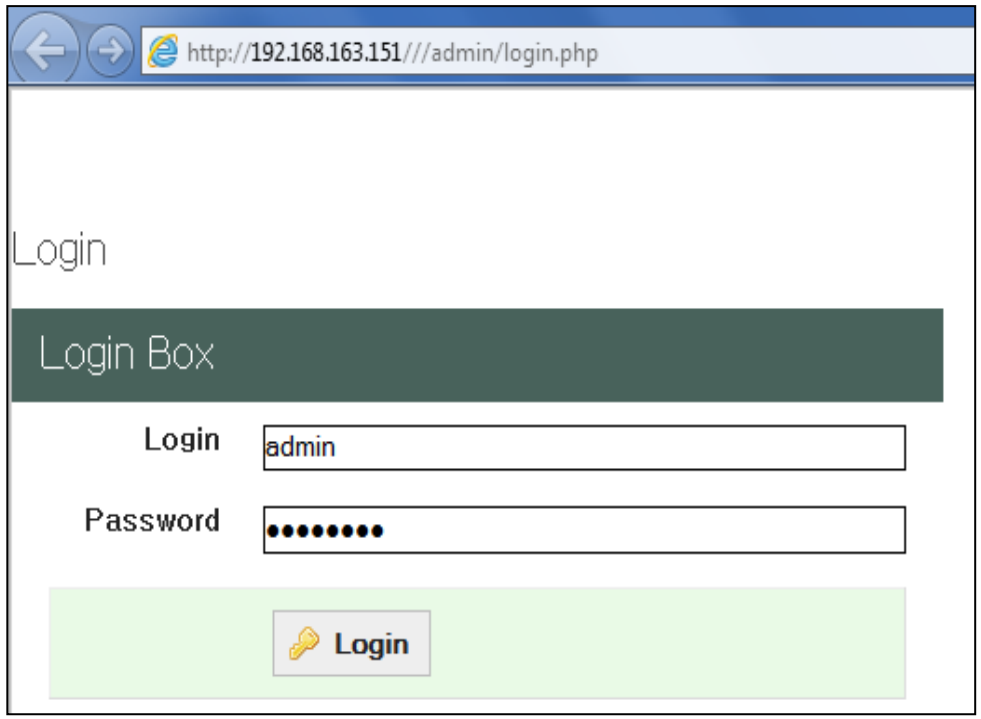
“数据库名”▸“表”▸“字段”▸“字段内容”，发现一个admin及长串密码字符串。

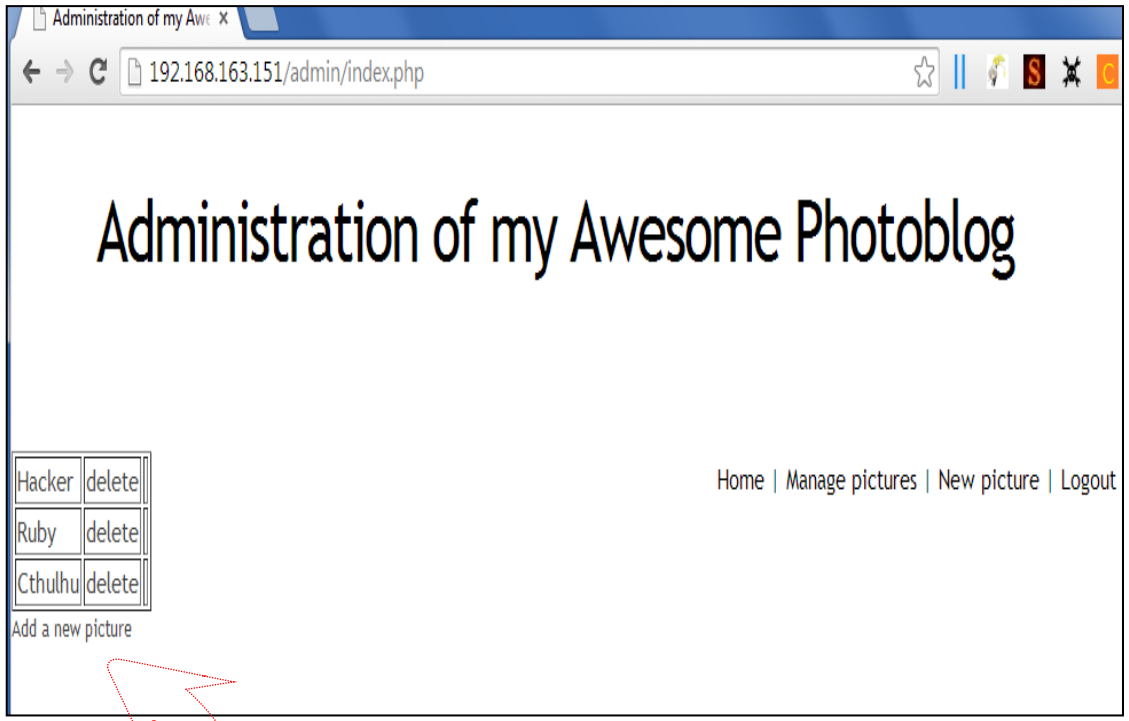


第五步：通过访问http://www.md5.com/密码解密网站进行破解，发现密码是P4ssw0rd。

第六步：找到网站后台，使用前面破解出来的用户密码admin:P4sswOrd登陆。

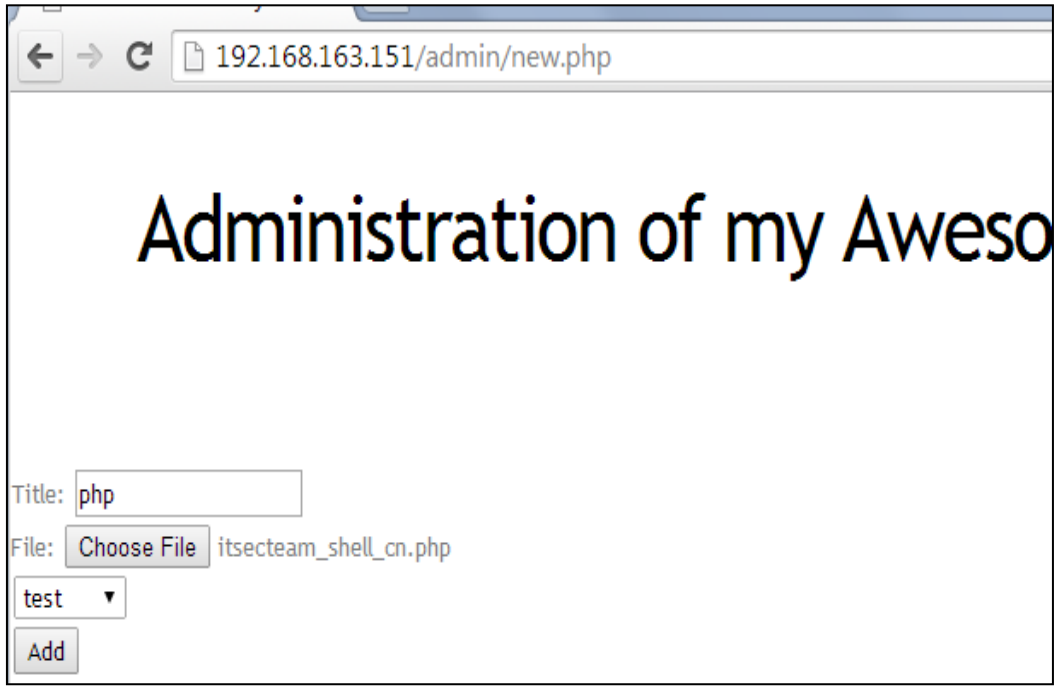
一旦成功登陆后台，可以查找文件上传组件。



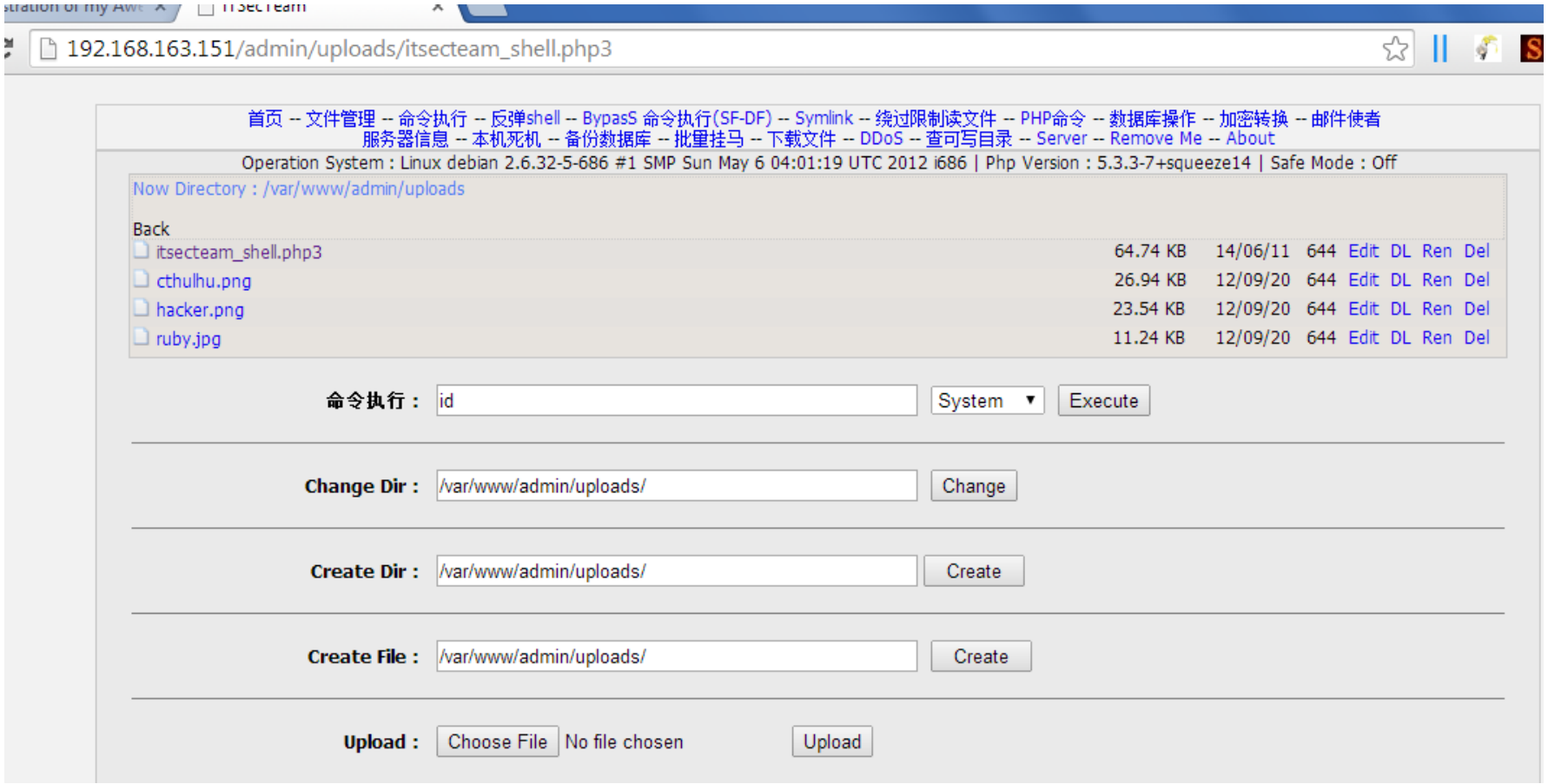


找到上传文件使用的菜单

第七步：我们先直接上传一个PHP webshell木马文件进行测试，提交后发现报错，服务器不允许此类型格式的文件，明显做了白名单，限定特定文件类型格式才可以上传。



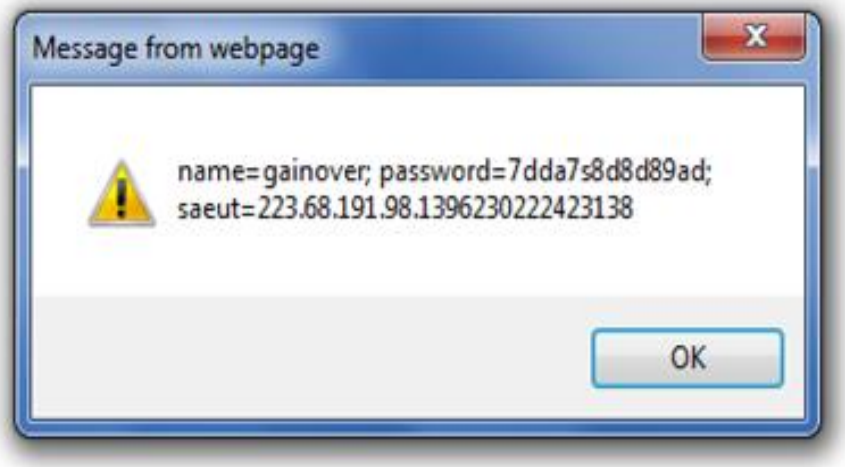
第八步：Apache有个解析漏洞：从右到左开始判断解析，如果右边不可识别解析，就再往左边判断。那我们将后缀修改为.php3。上传可以成功，我们试着去访问一下上传的这个脚本文件，OKay，webshell木马可以使用了。



第一章小节3：攻击web应用演示2：反射型XSS跨站脚本攻击演示

演示URL示例：http://xsst.sinaapp.com/example/1-2.php?page=2;alert(document.cookie)

第一步：在page=2后面添加 ;alert(document.cookie) 看一下效果，直接成功。



web应用开发者在看了上面的示例后，查看了原因，原来是参数被load时，根本没有进行任何的安全检查，好吧，那我抓紧把危险的字符、函数过滤掉。

第二步：代码既然修改好了，那我们再来看看。

演示URL示例：http://xsst.sinaapp.com/example/test1-2.php?page=3;alert(document.cookies);

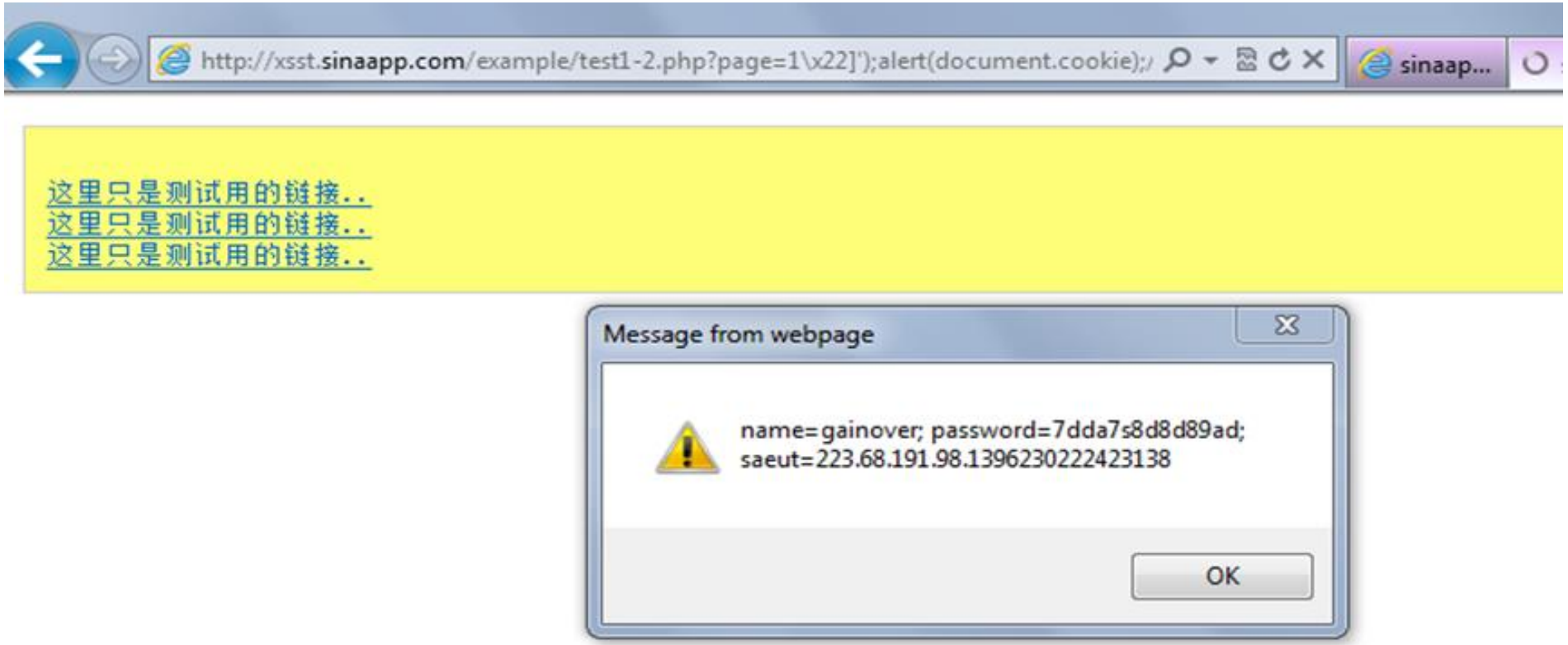
oH... 很不幸...你好像失败了！！

但我们知道Javascript支持转义的字符，我们可以将双引号变为转义字符 \x22。

第三步：我们调整为\x22]’);alert(document.cookie);再次提交到服务端，查看已经成功。

演示URL示例：

http://xsst.sinaapp.com/example/test1-2.php?page=1\x22]’);alert(document.cookie);

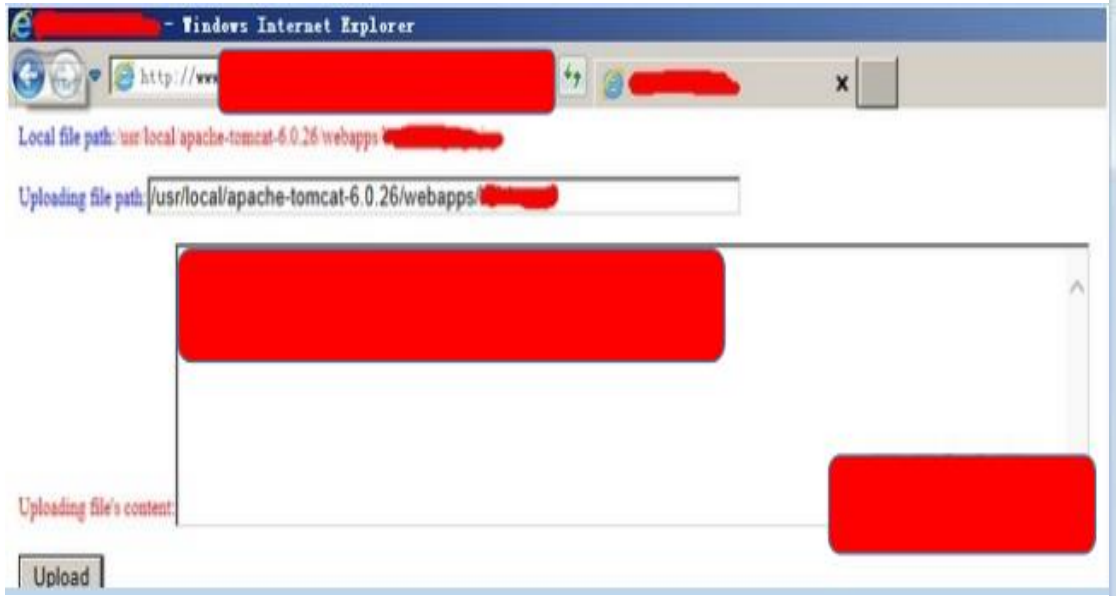


第一章小节3：攻击web应用演示3：Struts2框架命令执行漏洞演示

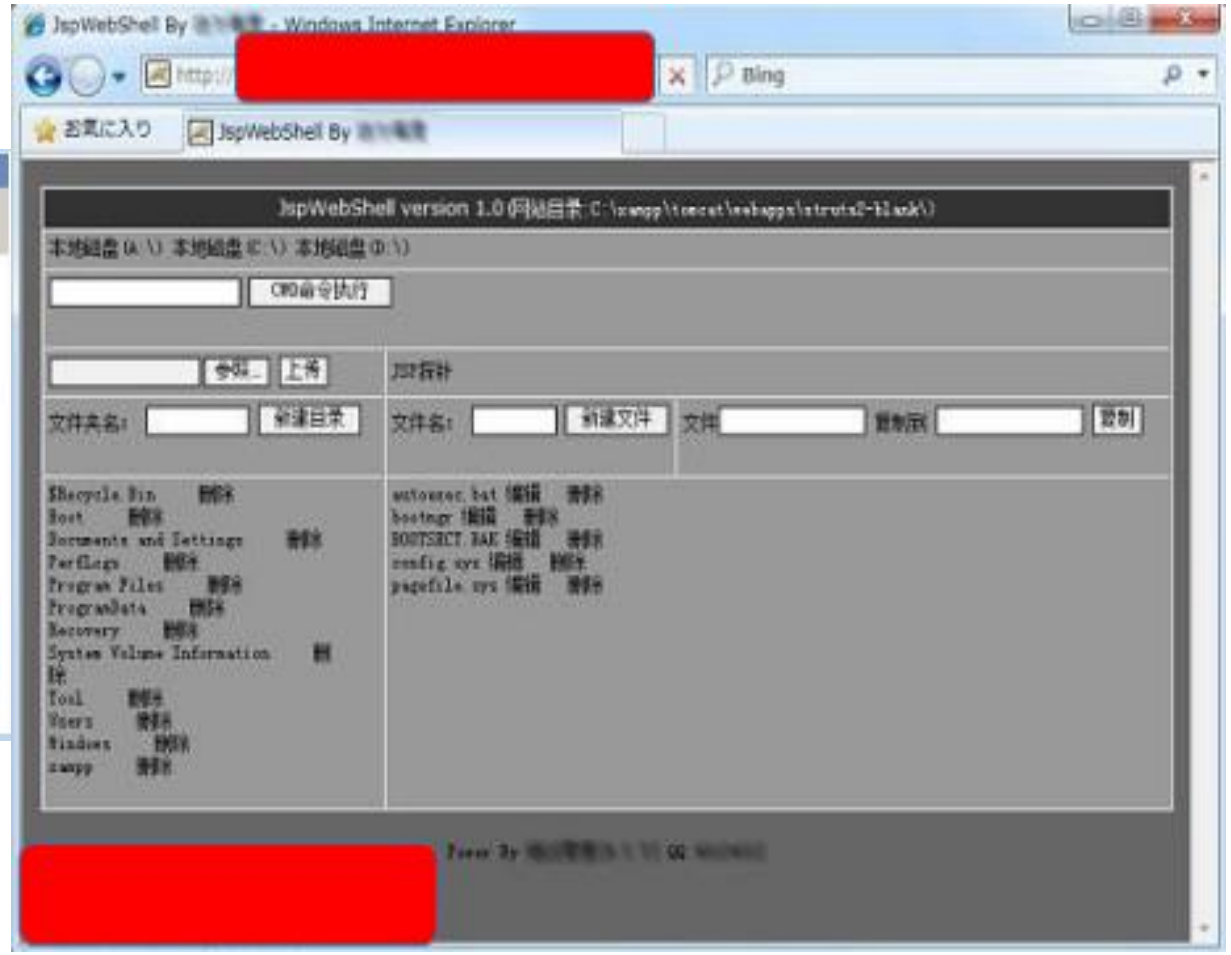
演示URL示例：http://xxx.xx.cn

第一步：用一个python脚本，测试后发现存在Struts2框架命令执行漏洞。

第二步：利用这个漏洞执行写入操作，写入一段具有jsp上传功能页面的代码，成功后发现可以访问该页面。



第三步：在输入内容框里提交jsp webshell的代码，从而获得了可以控制网站的webshell。



第一章小节3：攻击web应用演示4：命令注入漏洞攻击演示

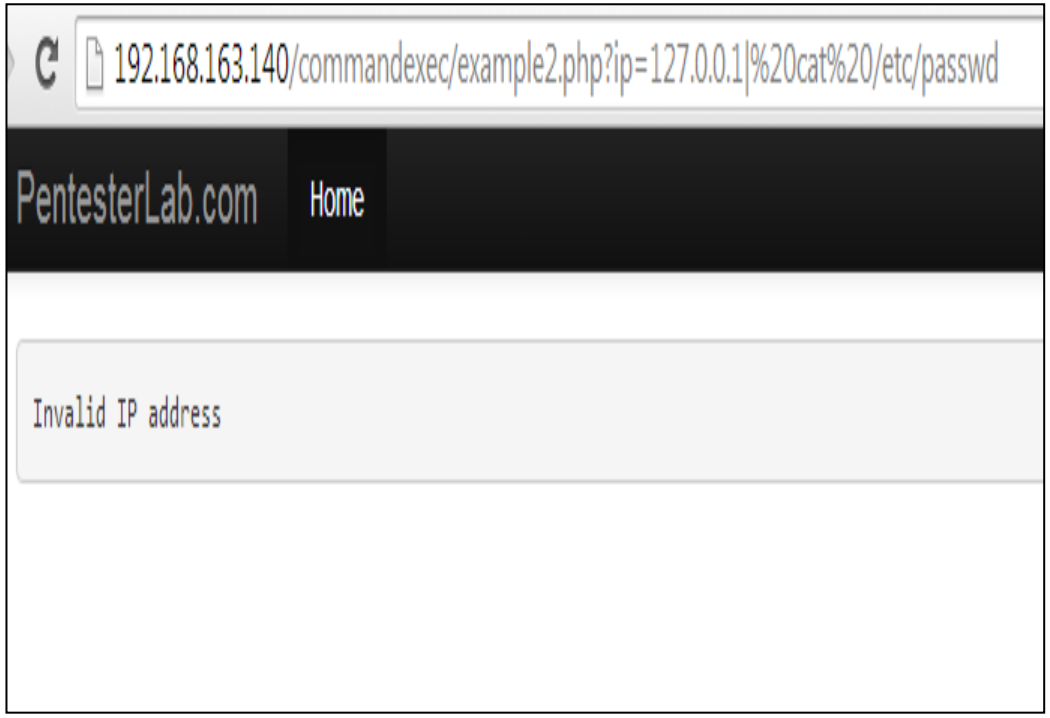
演示URL示例：

http://192.168.163.140/commandexec/example1.php?ip=127.0.0.1| cat /etc/passwd

这个演示URL中变量ip已经对输入进行了正则匹配，只获取ip作为参数。

第一步：直接提交非匹配的数据，结果会报错。

第二步：这里我们使用换行符(URL编码为%0a)来进行绕过。结果为命令已经成功被执行了。



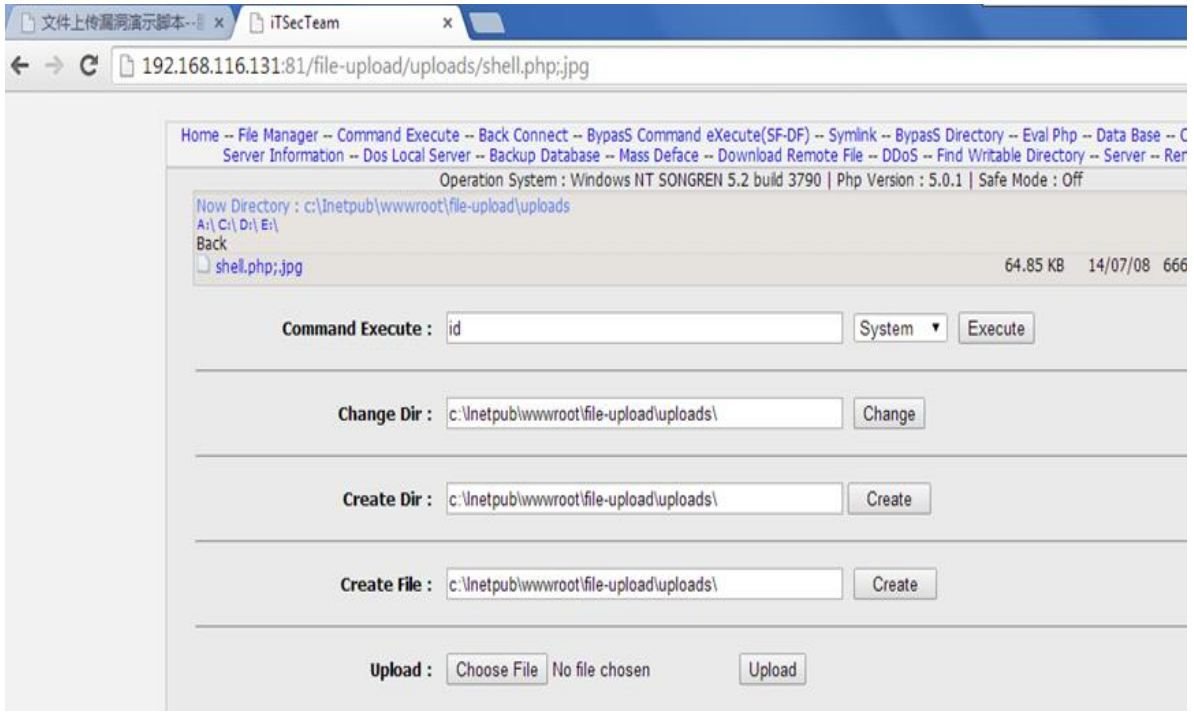
第一章小节3：攻击web应用演示5：文件上传漏洞攻击演示

演示URL示例：http://192.168.116.131:81/file-upload/uploads/

服务器端对用户上传的文件后缀进行检测，如果发现是不允许的后缀类型，就会禁止文件上传。

第一步：直接提交非匹配的数据会报错

第二步：结合IIS6.0解析漏洞，上传类似于xxx.php;.jpg，将被IIS6.0服务器解析为php文件，我们访问一下，webshell已经可以使用了。



认识下webshell木马

webshell是web入侵的脚本攻击工具。简单的说来，webshell就是一个asp或php脚本文件，黑客在入侵了一个网站后，常常在将这些asp或php木马后门文件放置在网站服务器的web目录中，与正常的网页文件混在一起。然后黑客就可以用web的方式，通过asp或php木马后门控制网站服务器，包括上传下载文件、查看数据库、执行任意程序命令等。

- webshell木马的危害

通过URL直接访问木马的方式，去控制网站服务器：包括上传下载文件、查看数据库、漏洞扫描、执行任意程序命令等。

- webshell木马的分类

  a) 一句话木马

  b) 小马

  c) 大马

a）一句话木马

- 工作原理

将一句话木马插入到脚本文件或者数据库中，其接收入侵者通过客户端提交的数据（在服务器端安装了一个接收器），主要用于上传大马使用。

- 一句话木马的常见形式

asp一句话木马：

```
<%execute(request("value"))%>
```

php一句话木马：

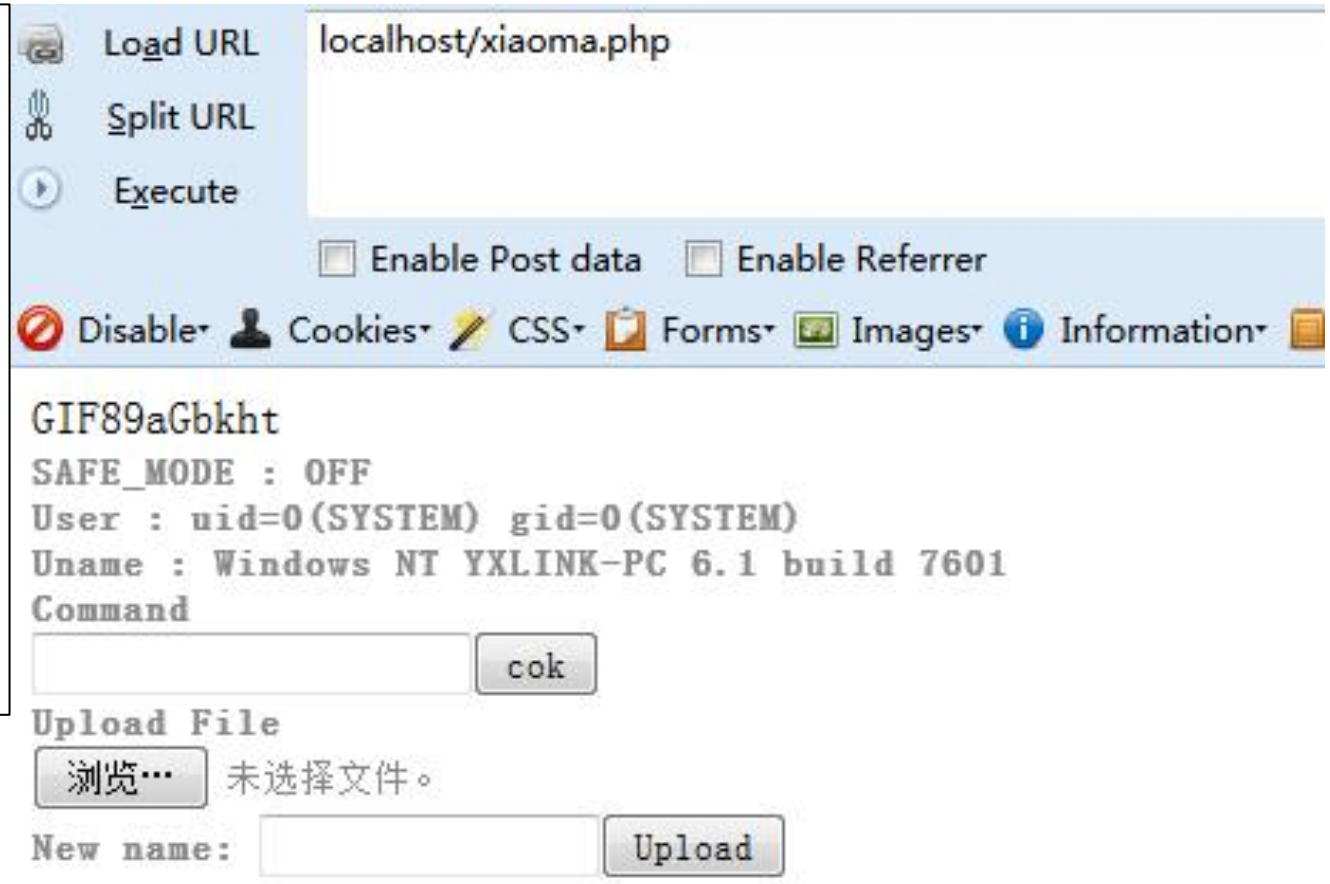
```
<?php @eval($_POST[value]);?>
```

aspx一句话木马：

```
<%@ Page Language="Jscript"%>
```

b）小马

小马与大马的区别是，小马的体积很小，因为有些网站为了防止被上传webshell木马，做了文件大小的限制，通常小马的体积只有几KB（隐藏性好），而大马的体积通常都要几百KB，另外，小马的功能简介，通常只有上传功能，为了然后上传大马而生。大马的功能很多，但是易被查杀。



c）大马常用的一些敏感函数

- 执行系统命令:

system()，passthru()，shell_exec()， exec()，popen()， proc_open()

- 代码执行与加密:

eval()，assert()，call_user_func()，base64_decode()， gzinflate()，gzuncompress()，gzdecode()，str_rot13()

- 文件包含与生成:

require()， require_once()，include()， include_once()，file_get_contents()， file_put_contents()， fputs()，fwrite()

第一章小节4：web漏洞的防范

漏洞类型	代码层面漏洞防范
SQL注入漏洞	● 查询语句都使用数据库提供的参数化查询接口 ● 检查变量数据类型和格式，并严格规定数据长度 ● 转译或过滤特殊符号 ● 使用普通用户权限连接数据库连接 ● 指定返回的错误页面 ● 设置web目录的许可权限
XSS跨站脚本漏洞	● 验证输入数据的格式、长度、范围和内容，对script、iframe等字样进行严格检查,允许有限的html，避免插入用户可控的数据 ● 输出确认：对输入的经常造成安全问题的字符进行编码
Struts2框架命令执行漏洞	● 升级struts2框架到最新版本
命令注入漏洞	● 在代码级调用shell时，对命令行中的特殊字符进行转义
文件上传漏洞	● 文件上传的目录设置为不可执行 ● 多种组合方式验证文件类型 ● 使用随机数改写上传文件的文件名和访问路径

第二章小节1：事前安全预警发现

WPDRRC信息安全模型是我国“八六三”信息安全专家组提出的适合中国国情的信息系统安全保障体系建设模型,信息安全模型在信息系统安全建设中起着重要的指导作用。其中W就是预警的环节，以此安全模型作为参考，我们要重视和建设web应用安全防护的预警，在应用系统的正式上线后，要第一时间发现安全问题，根据发现的安全问题危害严重程度，按步骤有效完成相应的防护工作，防止其被恶意攻击利用，造成严重的安全事故。推荐采用专业的漏洞扫描系统进行应用系统的安全弱点检查。

第二章小节2：事中安全加固实施

1、通过设置黑白名单、过滤特殊字符、验证数据输入输出等方式修复代码漏洞。

缺点是滞后性（平均周期为71天）、可能存在被绕过的风险以及可能造成业务系统的其他功能无法使用

2、安全加固web应用及其支撑组件（操作系统、数据库、脚本），防止攻击者攻击web应用支撑组件的方式获取到网站乃至服务器的管理权限。

3、采购专业的防护设备，如web应用防火墙、防篡改系统、入侵防御等产品。

（“漏报”、“误报”以及“防护被绕过”）

第二章小节2：事中安全加固实施：Linux系统加固

安全加固工作存在风险，加固前应做好应急准备措施，并做好数据备份。

以常见的web应用架构LAMP(Linux+Apache+Mysql+Php)加固作为示例。

序号	加固项目	加固目的	加固方法
1	账号	● 禁用/删除无用账号； ● 查找空/弱口令账号； ● 增强口令安全策略； ● 限制能su到root的用户；	• 使用命令“passwd -l <用户名>”锁定不必要的账号； • 使用命令“awk -F: '($2==""){' /etc/shadow”查看空口令账号； • 使用命令“vi /etc/login.defs”修改配置文件； 例如设置连续输错3次密码，账号锁定5分钟 auth required pam_tally.so onerr=fail deny=3 unlock_time=300 • 使用命令“vi /etc/pam.d/su”修改配置文件； 例如：只允许test组用户su到root auth required pam_wheel.so group=test
2	服务	● 关闭不必要的服务； ● 对SSH服务进行安全检查；	• 使用命令“chkconfig --level <init级别> <服务名> on|off|reset”设置服务在个init级别下开机是否启动； • 使用命令“vi /etc/ssh/sshd_config”编辑配置文件 （1）不允许root直接登录 设置“PermitRootLogin ”的值为no （2）修改SSH使用的协议版本 设置“Protocol”的版本为2 （3）修改允许密码错误次数（默认6次） 设置“MaxAuthTries”的值为3

序号	加固项目	加固目的	加固防范
3	文件	● 修改默认的UMASK值； ● 修改Bash保留历史命令的条数； ● 修改登录超时； ● 限制Ctrl+Alt+Del命令； ● 修改NFS共享； ● 使用TCP Wrapper配置访问控制；	• 使用命令“vi /etc/profile”，添加行“UMASK 027”； • 使用命令“vi /etc/profile”，修改HISTSIZE=5和HISTFILESIZE=5即保留最新执行的5条命令； • 使用命令“vi /etc/profile”，添加“TMOUT=”行开头的注释，可设置为“TMOUT=180”，即超时时间为3分钟； • 使用命令“vi /etc/inittab”，在行开头添加注释符号“#”#ca::ctrlaltdel:/sbin/shutdown -t3 -r now，再使用命令“init q”应用设置 • 使用命令“vi /etc/exports”，删除不必要的共享； • 使用命令“vi /etc/hosts.allow”和“vi /etc/hosts.deny”修改配置；
4	日志	● 查看启用所有系统日志	• 使用命令“cat /etc/syslog.conf”查看syslogd的配置 系统日志（默认）/var/log/messages cron日志（默认）/var/log/cron 安全日志（默认）/var/log/secure

第二章小节2：事中安全加固实施：Apache加固

序号	加固项目	加固目的	加固方法
1	隐藏版本信息	防止恶意攻击者利用相关版本漏洞进行攻击	使用命令“vi /etc/httpd/conf/httpd.conf” ServerSignature Off　#关闭服务器生成的版本信息 ServerTokens Prod　#关闭服务器应答头中的版本信息
2	禁止目录遍历	防止直接访问目录时由于找不到默认主页而列出目录下文件	使用命令“vi /etc/httpd/conf/httpd.conf” Options -Indexes FollowSymLinks
3	禁用CGI	如果不需要运行CGI程序，建议禁用CGI	使用命令“vi /etc/httpd/conf/httpd.conf” #LoadModule cgi_module modules/mod_cgi.so
4	禁用SSI	禁用SSI（服务器端包含）	使用命令“vi /etc/httpd/conf/httpd.conf” Options Indexes FollowSymLinks -Includes

序号	加固项目	加固目的	加固方法
5	关闭TRACE	防止TRACE方法被访问者恶意利用	使用命令“vi /etc/httpd/conf/httpd.conf” 添加“TraceEnable Off”
6	上传目录设置	禁止动态脚本在上传目录的运行权限，防止攻击者绕过过滤系统上传webshell	使用命令“vi /etc/httpd/conf/httpd.conf” 以 "/var/www/html/upload"为上传目录示例 <FilesMatch "\.php$"> Order allow,deny Deny from all </FilesMatch>
7	mod_rewrite模块	此模块提供了一个基于正则表达式分析器的重写引擎来实时重写URL请求，防止盗链攻击	使用命令“vi /etc/httpd/conf/httpd.conf” 添加RewriteEngine on RewriteCond %{HTTP_REFERER} !^http://[0-9a-z]+\.test\.com/.*$
8	自定义错误信息	自定义Apache返回的错误信息，防止信息泄露	使用命令“vi /etc/httpd/conf/httpd.conf” ErrorDocument 500 "The server made a boo boo." ErrorDocument 404 /missing.html

第二章小节2：事中安全加固实施：Mysql加固

序号	加固项目	加固目的	加固方法
1	账户	● 以普通帐户安全运行mysqld; ● 确保系统不存在脆弱密码;	• 可以通过在/etc/my.cnf中设置 • 如要修改密码，执行如下命令： mysql> update user set password=password('test!p3') where user='root'; mysql> flush privileges;
2	网络连接	禁止网络连接，防止猜解密码攻击，溢出攻击和嗅探攻击	如果数据库不需远程访问，可以禁止远程tcp/ip连接 #cat /etc/my.cnf #ps -ef|grep -i mysql
3	文件安全	拒绝未授权用户访问数据库文件	# ls -al .mysql_history .bash_history 应为600权限 #ls –al /etc/my.cnf 应为644权限 #find / -name .MYD |xargs ls –al 应为600权限 #find / -name .MYI |xargs ls –al 应为600权限 #find / -name .frm |xargs ls –al 应为600权限
4	数据库授权	确保数据库没有不必要的或危险的授权	回收不必要的或危险的授权，可以执行revoke命令 mysql> select * from user; mysql>select * from db; mysql>select * from host; mysql>select * from tables_priv; mysql>select * from columns_priv;
5	日志审核	启用审核记录对数据库的操作，便于日后检查	打开/etc/my.cnf文件 [mysqld] log = filename

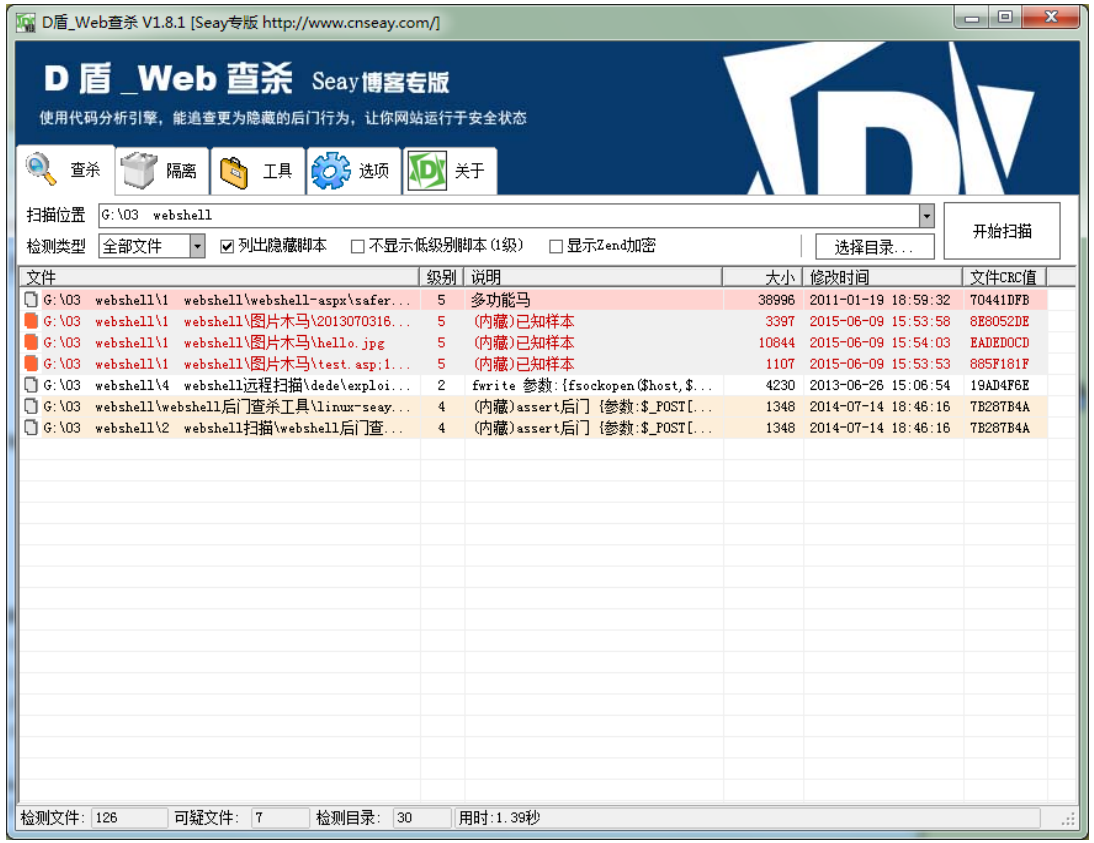
第二章小节2：事中安全加固实施：Php加固

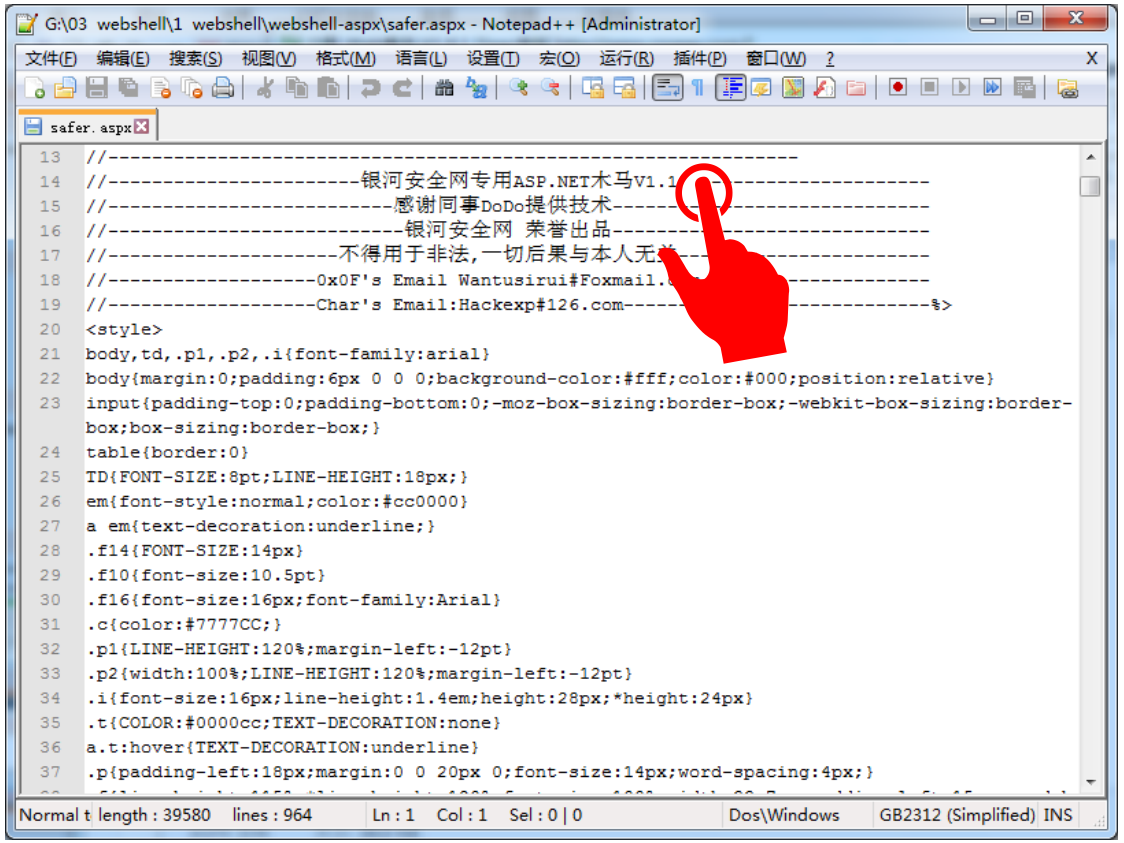
序号	加固项目	加固目的	加固方法
1	打开php安全模式	内嵌的非常重要安全机制，能够控制一些php中的函数使用	使用“vi /opt/lampp/etc/php.ini”， safe_mode = on
2	关闭PHP版本信息	防止黑客获取服务器中php版本的信息	修改php.ini， expose_php = Off
3	打开 magic_quotes_gpc	自动把用户提交对sql的查询进行转换，防止SQL注入	修改php.ini， magic_quotes_gpc = On
4	关闭远程文件打开	防止黑客远程远程包含漏洞	修改php.ini， allow_url_fopen = off
5	错误信息控制	防止错误信息暴露给恶意攻击者	修改php.ini， display_errors = Off

第二章小节3：事后安全应急处置：webshell木马查杀

专用木马查杀工具

- Windows系统可使用“铱迅webshell扫描器”、“D盾_web查杀”
- Linux系统可使用linux-seay查杀工具
- 如果网站有可信的数据备份，还可能使用文件夹比较工具，像Beyond_Compare。





Windows系统木马查杀结果

级别5显示的可疑文件是已经被确认的就是木马文件，可以直接删除。我们可以打开源文件看一下。

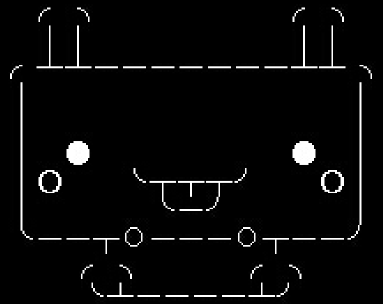
查出来的其他级别的文件，多是因为敏感函数的使用，而可能正常的脚本文件也使用，这就需要人工去确认的。

Linux系统查杀结果
Linux-seay查杀工具使用方法：
python main.py 查杀路径

```
文件:   F:\shell\phpddos\xin.php
后门描述:   PHP ddos_cc攻击脚本
后门代码:   fsockopen("udp:
最后修改时间: 2012-09-19 11:04:50

文件:   F:\shell\phpddos\xmvnnll.php
后门描述:   PHP ddos_cc攻击脚本
后门代码:   fsockopen("udp:
最后修改时间: 2010-10-10 10:10:10

文件:   F:\shell\phpddos\zlwh.php
后门描述:   PHP ddos_cc攻击脚本
后门代码:   fsockopen("udp:
最后修改时间: 2013-05-12 16:14:20

文件:   F:\shell\phpddos\连接攻击.php
后门描述:   PHP ddos_cc攻击脚本
后门代码:   phpddos
最后修改时间: 2012-03-22 17:11:25

【查杀完成】
        后门总数: 64

F:\killshell>
```

如果我们知道网站被攻击的一个时间段，
也根据文件最后修改时间查找后门

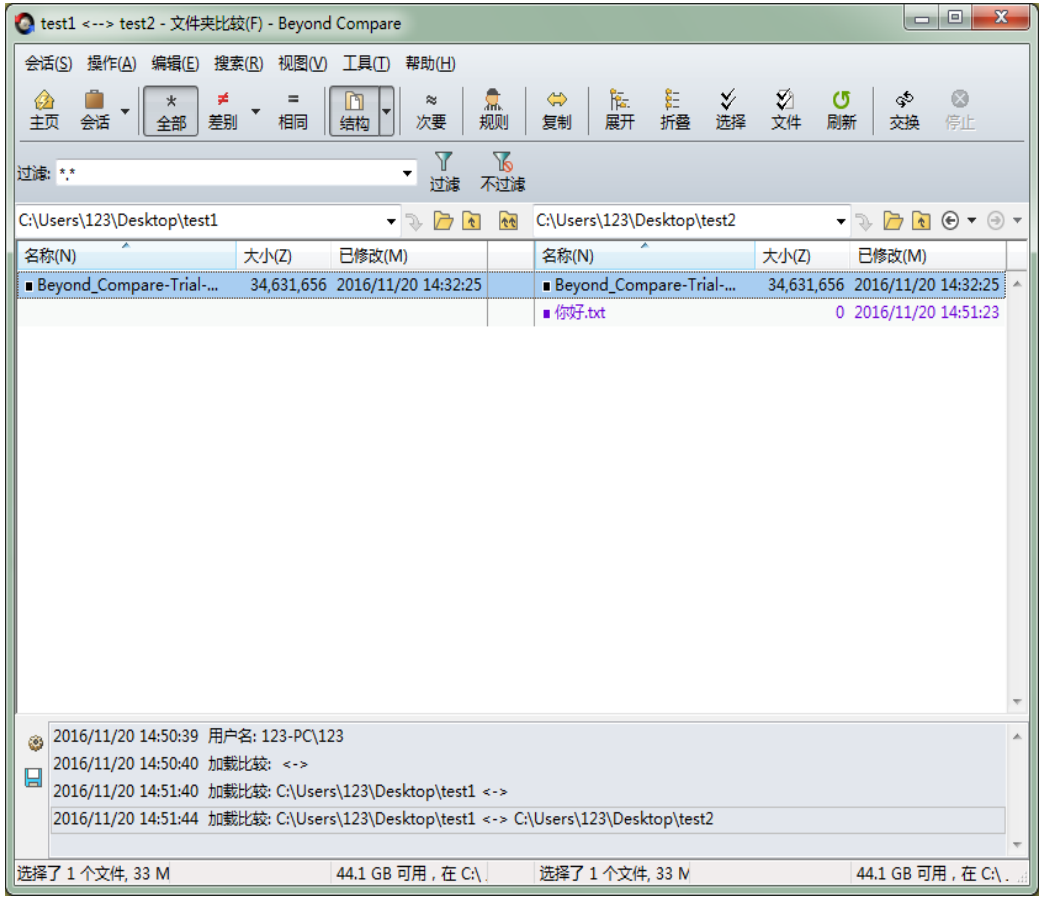
```
F:\killshell>main.py F:\shell\phpddos  "2014-07-13 00:00:00"
```

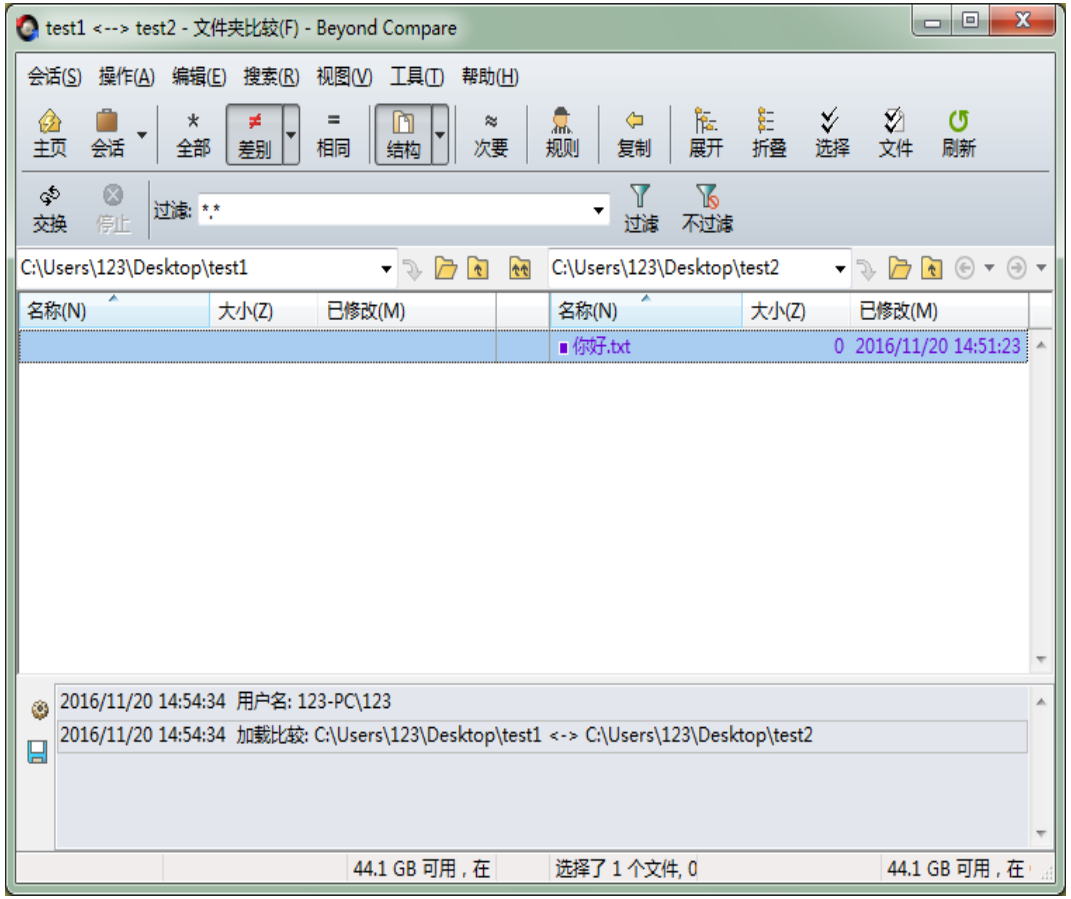


```
SeayFindShell 1.0
Author:Seay
SITE:www.cnseay.com

【开始查找】
F:\shell\phpddos

############################################
文件路径              最后修改时间

F:\shell\phpddos\adminy.php         2014-07-14 18:36:06
F:\shell\phpddos\CC攻击.php          2014-07-14 18:36:09

【查找完成】
        文件总数: 2
```

文件夹比较

测试文件test1和test2，选取要进行比较的文件夹进行比较，点击“差别”，右侧截图我们可以看到test2文件夹比test1文件夹多了一个你好.txt





第二章小节3：事后安全应急处置2：系统安全检查

一旦web被攻击成功后，大多数的黑客都会通过一些提权漏洞，创建具备管理员权限的系统账号，打开系统远程管理或者安装系统级别的后门木马，方便远程控制，进而控制整个服务器操作系统。我们必须要对服务器OS进行安全检查，以确保服务器操作系统是安全可信的，避免了网站漏洞修复、webshell木马查杀后，服务器还是被恶意攻击者远程控制着，为所欲为。

服务器OS的安全检查，涉及的内容比较多，主要检查的项目有：系统账号、启动服务、异常行为、关键日志等部分组成。

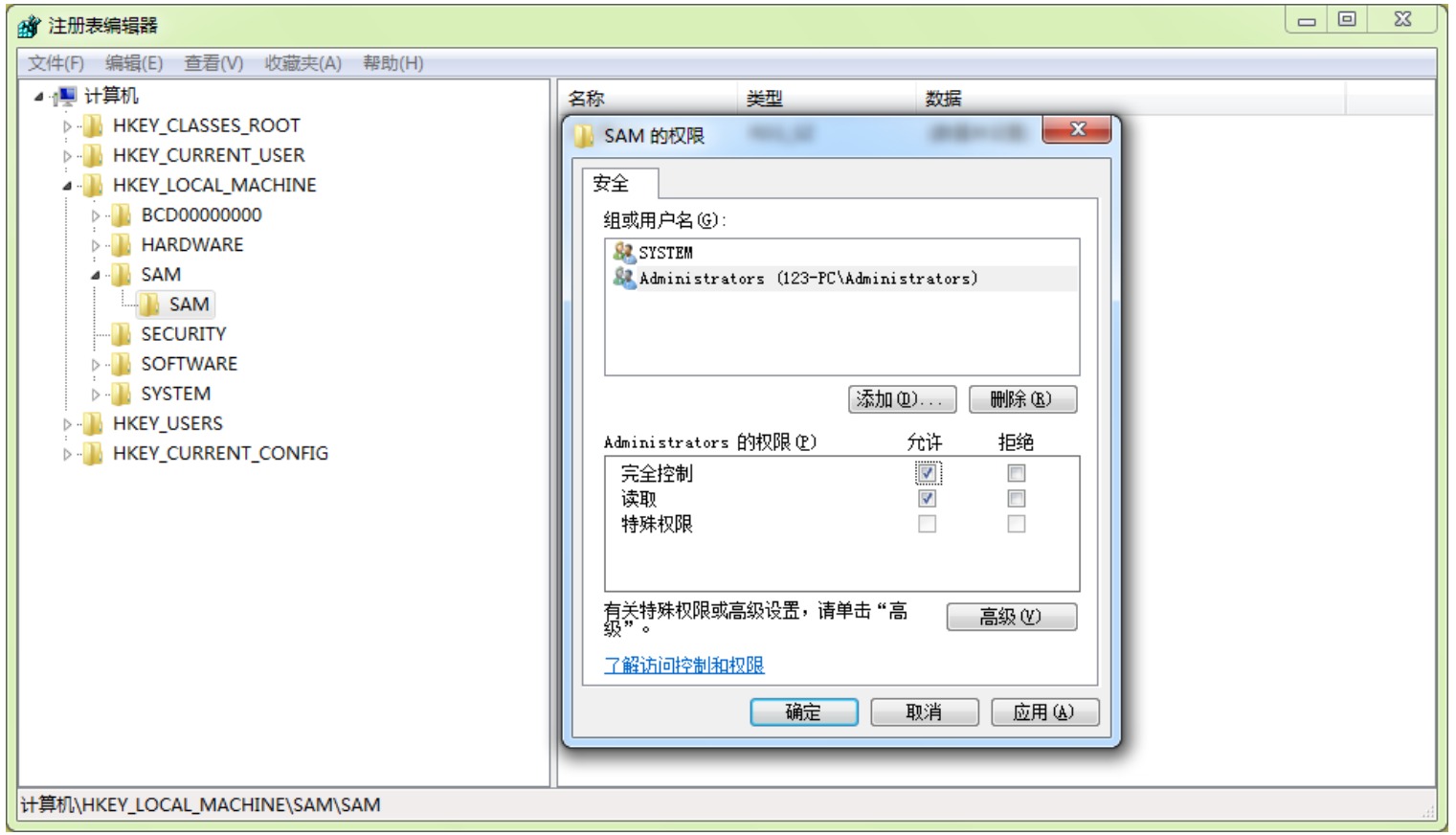
- Windows操作系统安全检查
- Linux操作系统安全检查

操作系统安全检查项目

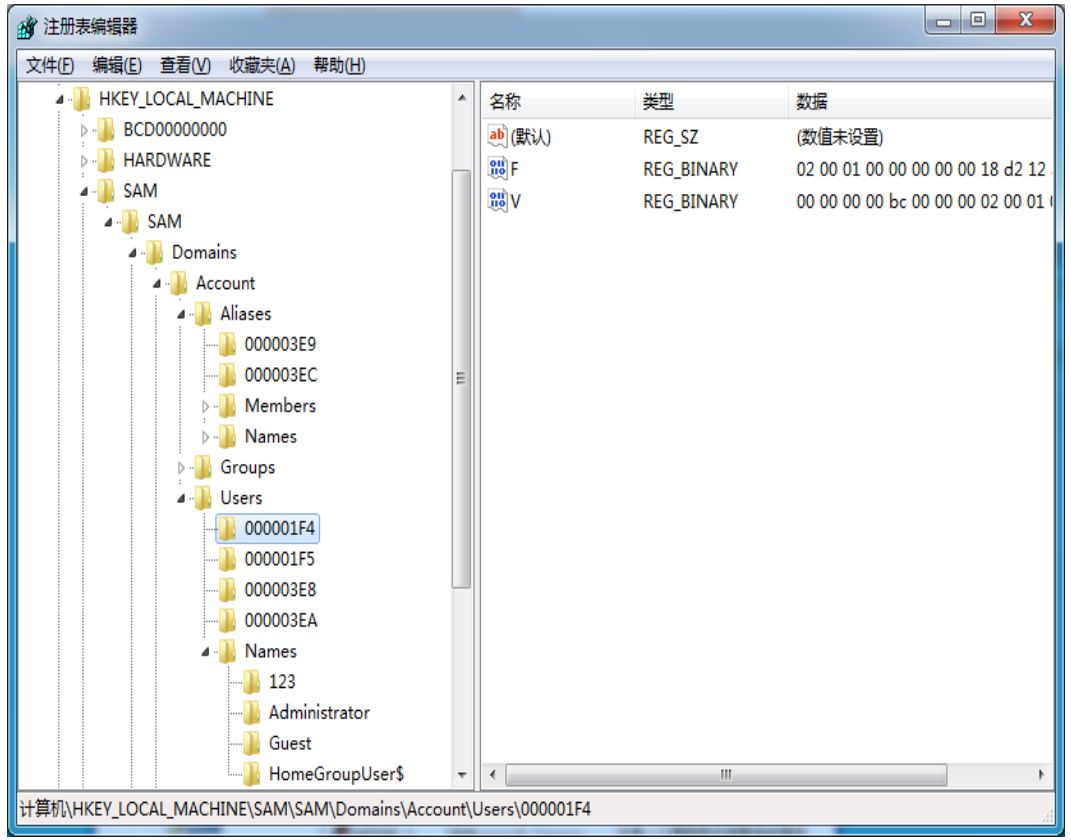
操作系统类型	安全检查项目	说明
Windows系统	系统账号及权限检查	检查是否存在隐藏、不明系统账号及其对应的权限信息
	服务与进程检查	检查是否存在隐藏、不明系统服务和进程
	其他异常行为检查	进行后门、木马、病毒检查
	日志安全分析	分析日志，找出攻击者的来源地址及其活动轨迹
Linux系统	敏感文件检查	检查系统敏感文件是否为系统默认值，不明的变化往往表示系统遭到入侵
	Rootkit后门检查	进行Rootkit检查
	进程与服务检查	检查是否存在隐藏、不明系统服务和进程
	日志安全分析	分析日志，找出攻击者的来源地址及其活动轨迹

Windows操作系统安全检查

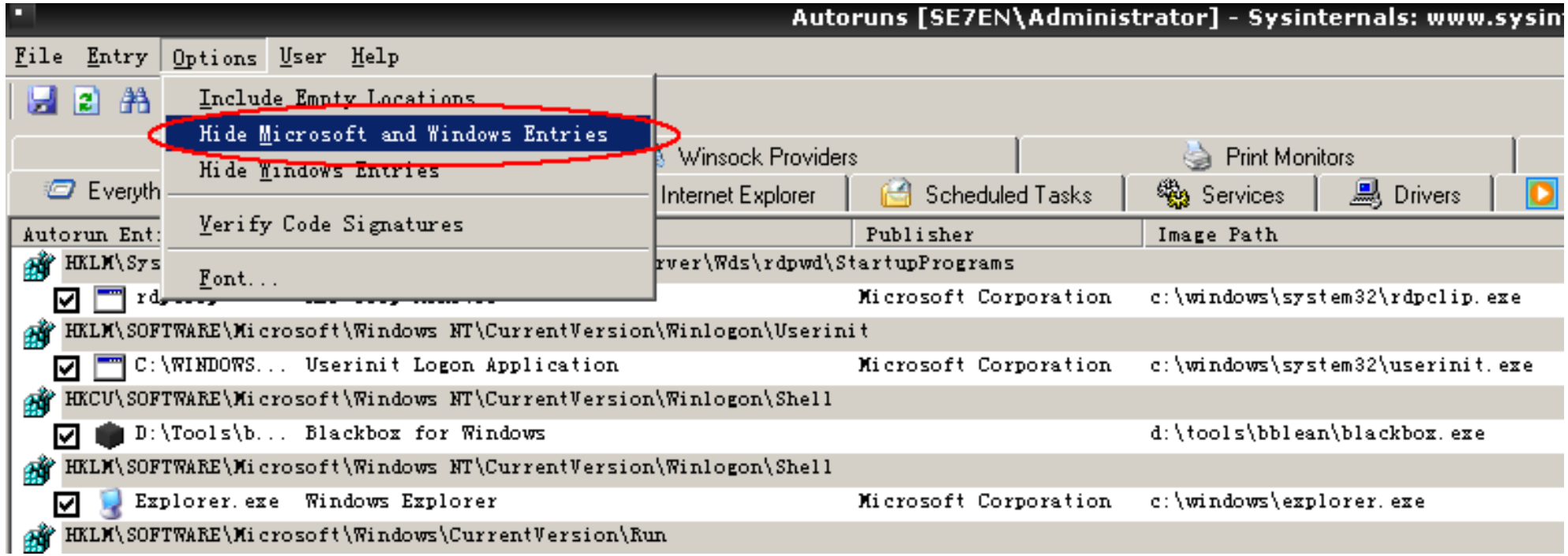
第一步：进入注册表，授权Administrators组用户对“SAM”表项的完全控制权限，如下图所示，确定后，关闭注册表，重新打开注册表。



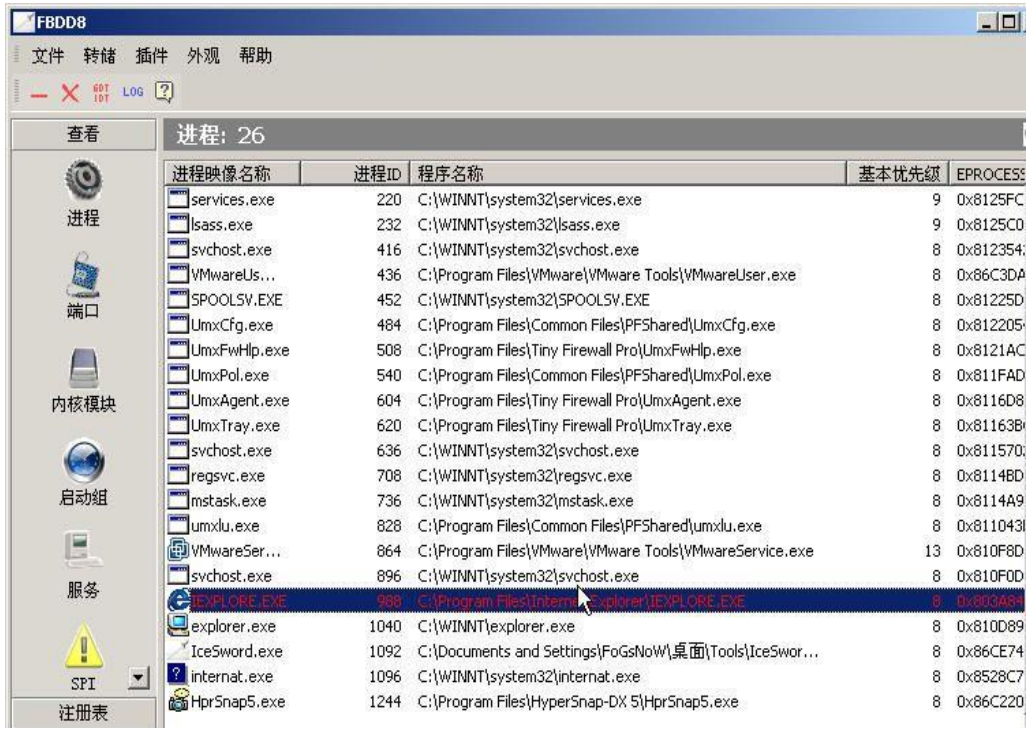
第二步：通过查看SAM下面的Users和Names两个键值及对应关系，可以找到隐藏的系统账号、克隆管理员账号等，一旦找到异常的系统账号，可以通过“net user 账号”，列出此账号相关的信息，如最后一次登录系统的时间、所属用户组等。

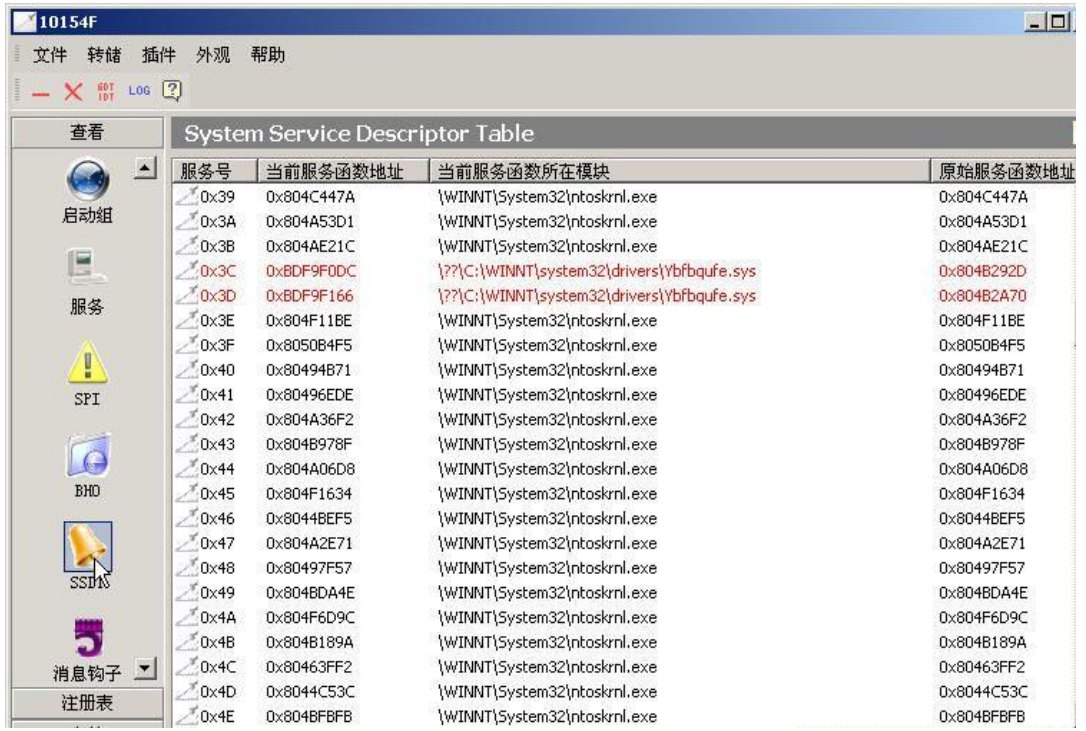


第三步：运行Autoruns列举出系统自启动服务的条目，通过选择“Options菜单”中的“Hide Microsoft and Windows Entries”，可由Autoruns自动筛选部分安全的启动项，由此找出异常的应用程序及路径。

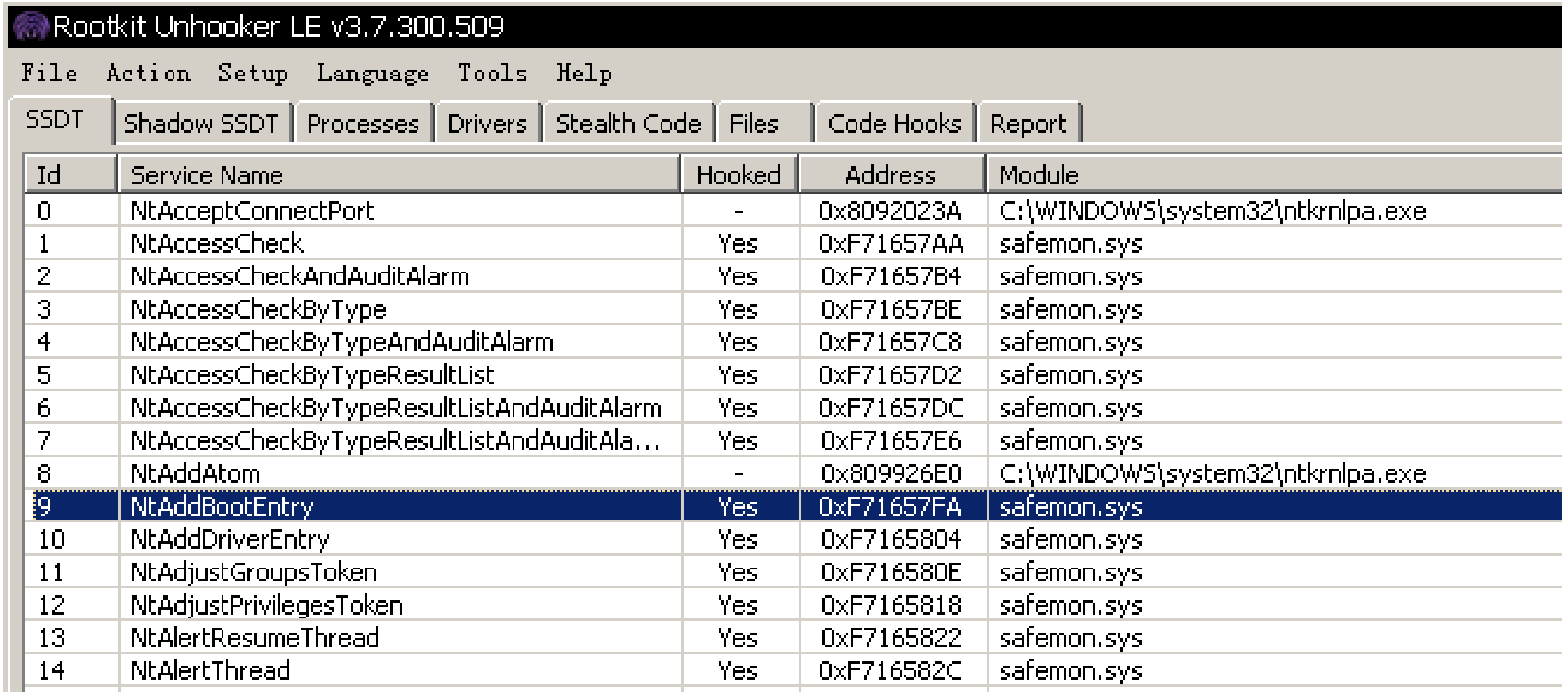


第四步：IceSword是一款集合了较多实用功能的安全检测工具，利用IceSword的进程功能，可查看系统中是否存在隐藏进程，若存在，则该进程会被自动标注为红色





第五步：Rootkit Unhooker具有Hook检测和恢复等功能，是用于检测以HOOK实现隐藏的后门程序的工具，根据界面中的“Hooked”可查看HOOK情况：若在Hooked下显示为YES，则应检查对应的Module中的文件，查看该文件是否为恶意程序。



第六步：系统产生的日志默认分为三类：

a） 应用程序日志

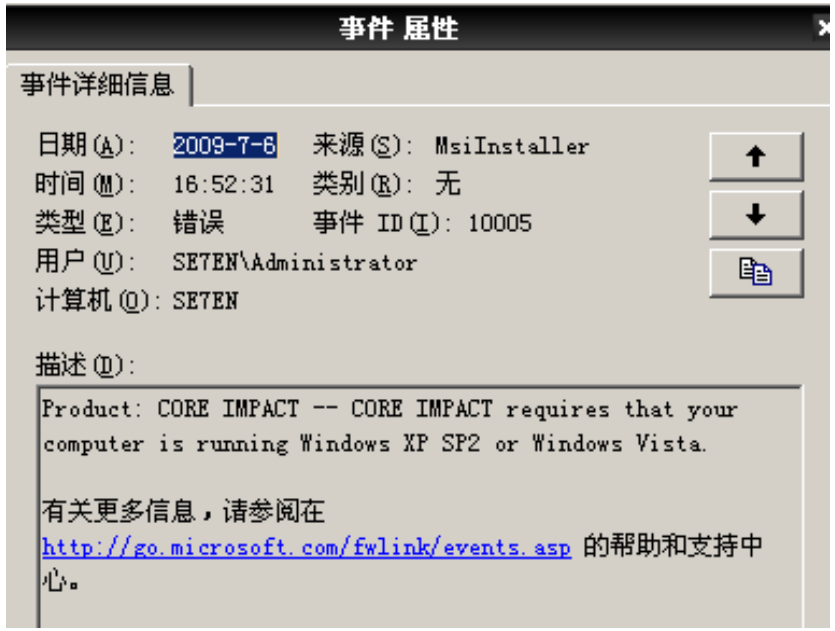
b） 安全性日志

c） 系统日志

这些日志以evt文件形式存储在%systemroot%\system32\config目录下，使用日志查看器可查看这些日志（开始 - 运行 - eventvwr）

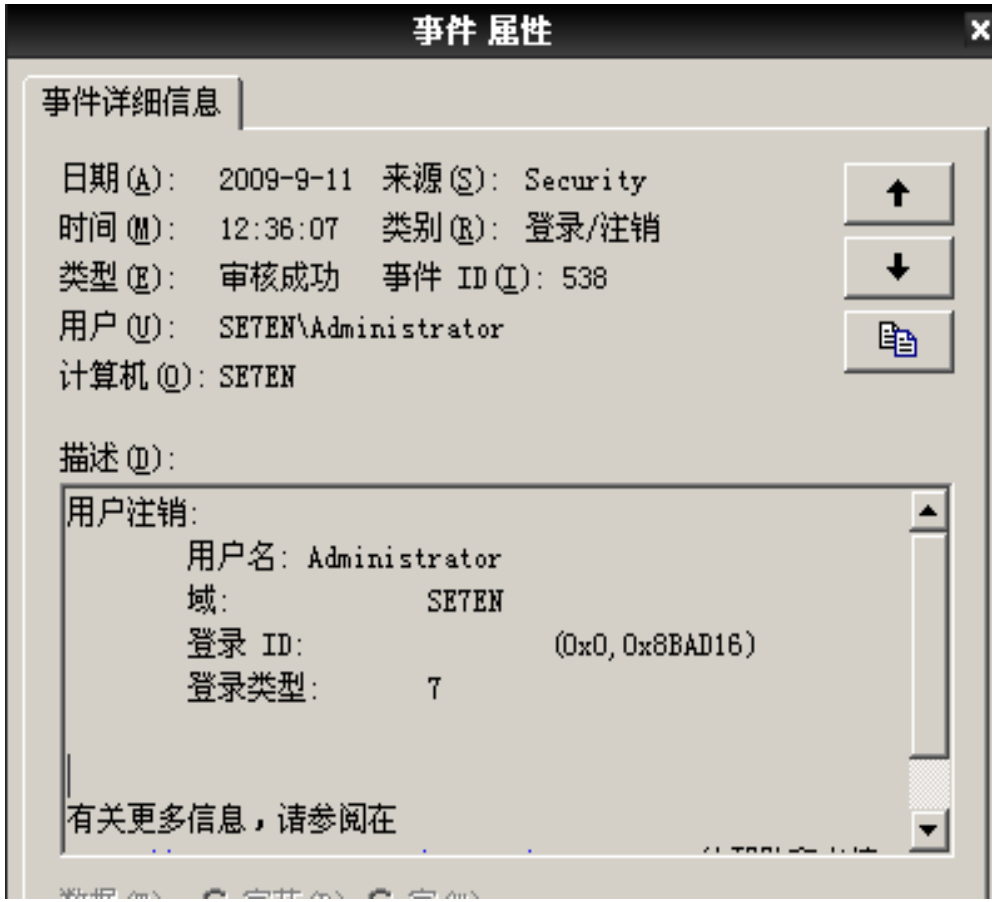
a）应用程序日志

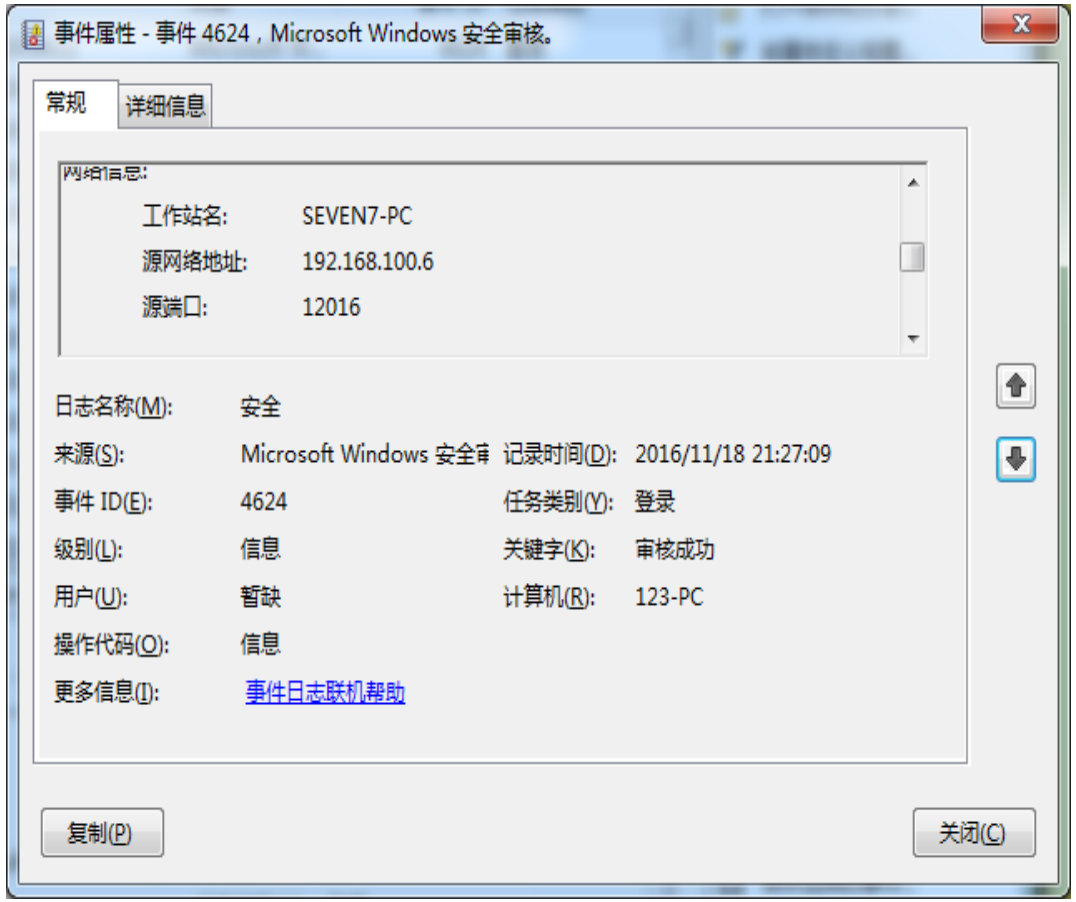
一些应用程序运行错误的话，可能会记录在应用程序日志中，利用这些日志可判断哪些程序运行错误以及错误内容：



b）安全性日志

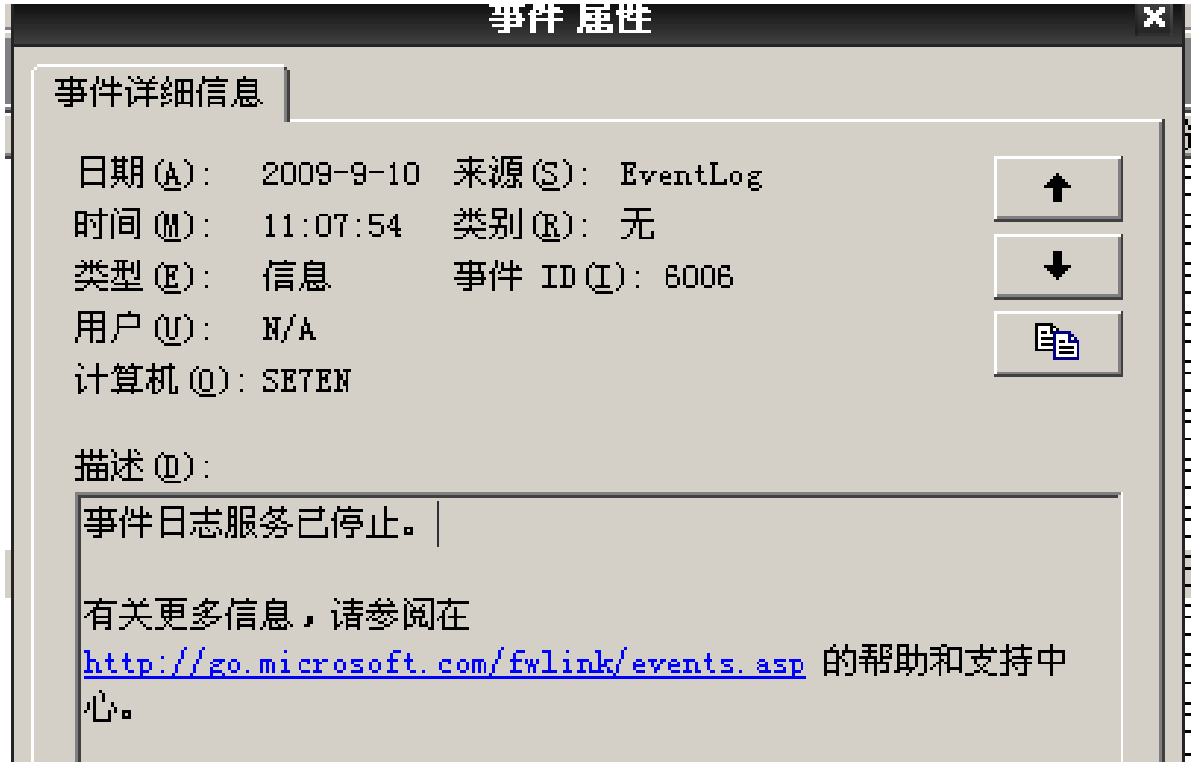
登录审计日志需保证系统启用了正确的日志审计功能。例如下图，该日志记录了administrator账号的登陆及源IP地址。





c）系统日志

Windows 默认情况下没有额外的系统启动日志或相关记录程序，因此，需依靠一些服务来判断系统的系统，而其中 event log 服务是最好的参考标志，event log服务的启动和停止就意味着Windows系统的启动和停止：可能是由于远程溢出失败而引起蓝屏，或由于某些误操作引起意外关机等行为。



linux操作系统安全检查

第一步：进行用户敏感文件的检查，查看是否存在异常的情况。

ls - l /etc/passwd /etc/passwd默认权限为644，其最小权限为444

- cat /etc/passwd 查看是否存在可疑帐号
- ls - l /etc/shadow shadow默认权限为600，最小权限为400
- awk -F : '$3==0{print}' /etc/passwd 检查除root用户外是否存在其他用户的UID为0
- find / -perm -004000 -type f 输出所有设置了SUID的文件
- rpm -Va 列举全部软件包的变化情况
- rpm -V package 列举某个程序包的变化情况

第二步：系统后门检查，两个知名的后门检查工具chkrootkit、Rootkit Hunter

● chkrootkit编译： tar xvzf chkrootkit.tar.gz cd chkrootkit-xx make sense chkrootkit使用： ./chkrootkit -q	● Rootkit Hunter编译： tar xvzf rkhunter-xx.tar.gz cd rkhunter-xx ./install.sh --layout default --install Rootkit Hunter使用： rkhunter -check

第三步：检查系统进程及服务是否存在异常

- 进程信息

Linux系统中使用命令ps -aux查看进程

Solaris系统中使用命令ps -eaf查看进程。

- 服务信息

Linux系统下可以使用chkconfig -list查看服务启动信息，各服务的启动脚本存放在/etc/init.d/和/etc/xinetd.d目录下。

第四步：检查分析系统日志

- messages日志

messages中记录有运行信息和认证信息，对于追查恶意用户的登录行为有很大帮助，例如，下面即为一条su日志：

Mar 22 11:11:34 abc PAM_pwdb[999]:authentication failure;cross(uid=500)->root for su service

- cron日志

RedHat的cron日志默认记录在 /var/log/cron 中

- secure日志

Linux的ssh登录日志会存储于/var/log/secure中，若日志中出现连续大量的登录错误信息，则可能意味着远程主机在尝试破解ssh登录口令。

- last日志

last命令用于查看最近的用户登录情况，last命令读取wtmp内容。

安全运维小贴士

- 定期化执行应用系统安全漏洞检查，先于恶意攻击者发现，及时调整防护策略；
- 定期化执行应用系统木马后门检查，第一时间发现存在的威胁并清理，保证发生安全事故的影响损失最小化；
- 定期化执行应用系统的数据备份工作，保证发生安全事故后能够最短时间内的业务恢复工作；
- 构建高度安全的应用防护框架，可参考专业安全的web安全防护解决方案，确保web应用及其支撑平台运行在高度安全、可信的环境中；
- 将应用系统变更、维护、升级等基础工作流程规范化；

author：田国强

Email：guoqiangtian@yxlink.com

Mobile：131 7672 0813

Info：12年信息安全工作经验，300+信息安全项目技术支持，擅长路由及交换、安全技术、安全管理服务咨询及电子取证等，勤勤恳恳，以学习和分享信息安全知识而充实和快乐。