



Silic Security Handbook

习科道展网络信息安全顾问

最具实力的网络安全专家

*本手册作者 Silic Corporation 第一任技术主管 Juliet, 版权归习科所有
手册建于 2012 年末, 已由 21 节完善至 44 节, 并收录至习科 166 页第三版黑皮书第六章

索引

1) 网站容器安全

- 1.1 修改 HTTP 头部
- 1.2 去掉 IIS 的 HTTP Trace 协议
- 1.3 隐藏 Apache 的签名
- 1.4 IIS 日志设置
- 1.5 nginx 的配置方法
- 1.6 Nginx 隐藏 http 头部中的版本信息

2) Apache + PHP

- 2.1 Apache + PHP 高级静态伪装
- 2.2 禁用 PHP 敏感函数
- 2.3 Apache 禁用文件索引
- 2.4 虚拟主机配置文件防读取
- 2.5 禁用危险的 php socket
- 2.6 防止 php 跨目录读写其他文件
- 2.7 禁止特定目录执行 CGI 脚本
- 2.8 对特定文件进行登录验证
- 2.9 PHP 非法 Session 值爆物理路径相关
- 2.10 Apache 索引目录中文乱码解决方式：

3) 服务器安全

- 3.1 卸载(关闭)Wscript.Shell
- 3.2 创建和清除特殊属性的文件
- 3.3 服务器时间查看器安全性日志失效

4) 数据库安全设置

- 4.1 Mysql.user 安全设置
- 4.2 MySQL 修改工作端口

需要做参考的自留即可，不要传阅外部人员。随时在群共享更新。

1) 网站容器安全

1.1 修改 HTTP 头部

这是一个正常的 HTTP 头部信息

HTTP/1.1 200 OK

Date: Fri, 20 May 2011 18:37:50 GMT

Server: Apache/2.2.17 (Win32) PHP/5.2.8

X-Powered-By: PHP/5.2.8

Set-Cookie: PHPSESSID=hakhoeidtb1kv78dh4aik8arc6; path=/

Expires: Thu, 19 Nov 1981 08:52:00 GMT

Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0

Pragma: no-cache

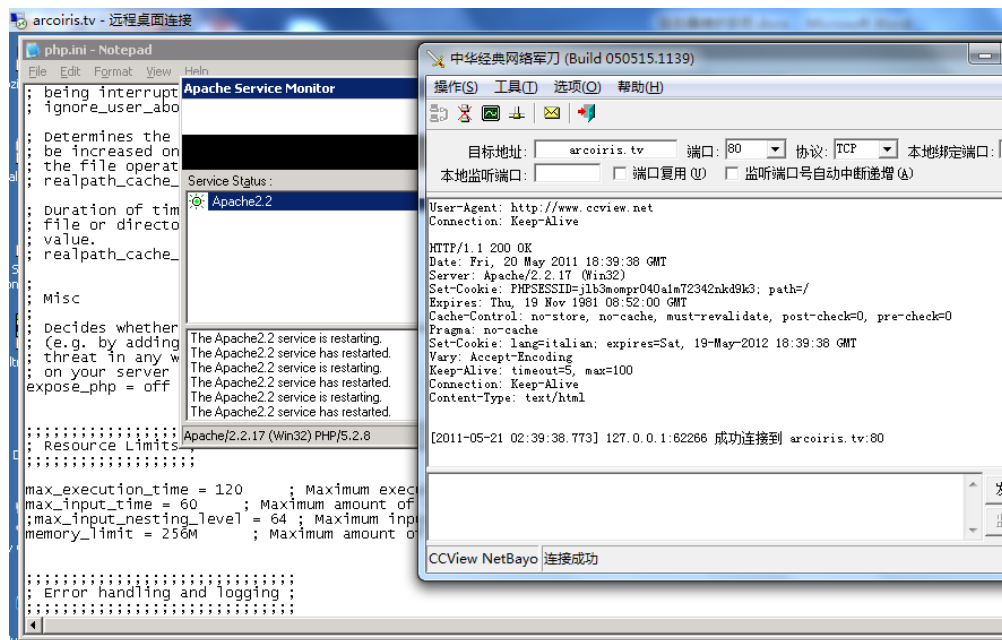
Set-Cookie: lang=italian; expires=Sat, 19-May-2012 18:37:50 GMT

Vary: Accept-Encoding

Keep-Alive: timeout=5, max=100

Connection: Keep-Alive

Content-Type: text/html



先是 PHP 的头部：修改 `php.ini` (Windows 下可能是 `php5.ini`，具体名称看服务器的配置情况) 也有可能是在 `apache/bin/php.ini`，修改下列参数：

```
Expose_php = off
```

保存后，重启网站容器后即可生效。

1.2 禁用 IIS 服务器 HTTP 的 trace 协议

几乎每份针对国内网站的 Nessus 报告或者 X-Scan 的报告都会提示服务器支持 trace 协议容易造成跨站等攻击。

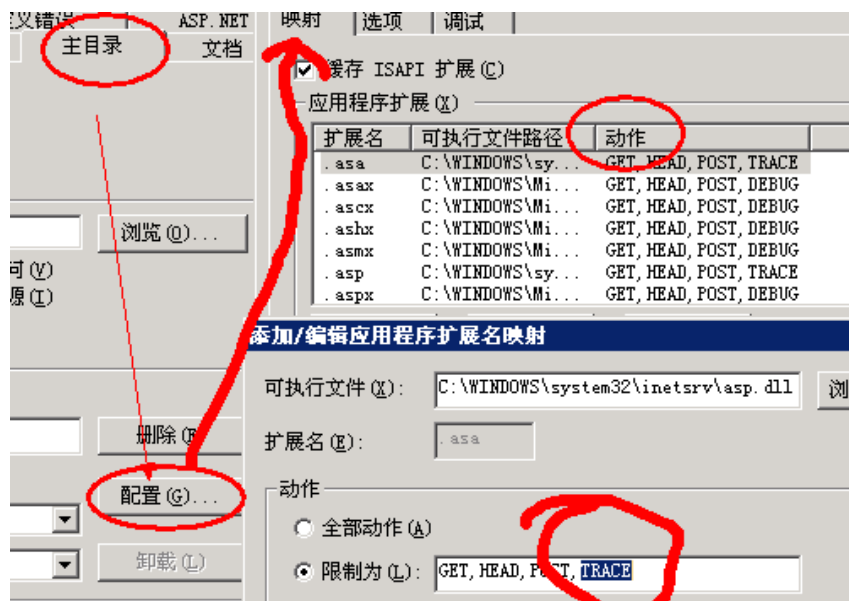
找一下 IIS 的配置文件：

```
c:\windows\system32\inetsrv\MetaBase.xml
```

把网站配置里面的 TRACE 删除没然后重启 IIS 就可以了：

```
\inetsrv\asp.dll,5,GET,HEAD,POST,TRACE
```

这里的 TRACE 去掉，如果服务器有装 WebDAV，要把 PUT、Copy、Delete 等危险协议一并删除，否则果断悲剧。IIS 为 Windows 的网站容器，也有图形界面的操作，见这里(Apache 等默认不支持这些协议)：



错误!未找到引用源。

1.3 隐藏 Apache 签名

签名 1 (位于 404、403、500 等页面中)：

Error 404

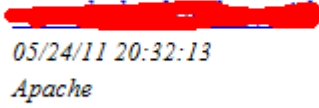
05/24/11 20:28:38

Apache/2.2.9 (Win32) DAV/2 mod_ssl/2.2.9 OpenSSL/0.9.8h mod_autoindex_color PHP/5.2.6

在 extra 的 http-default.conf 中设置 `ServerSignature Off` 即可隐藏默认签名。下图

是设置签名为 *off* 的效果：

Error 404



签名 2(服务器 HTTP header 上面的签名)：

```
Server: Apache/2.2.3 (CentOS) Server at xxxxxx Port 80
```

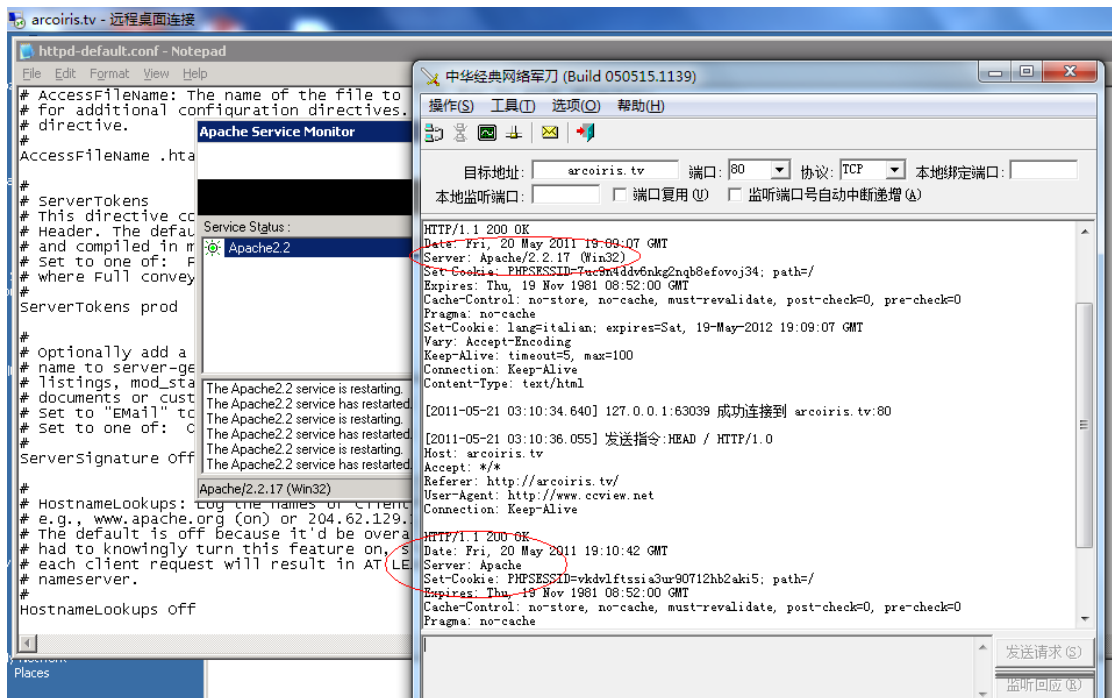
设置在这里：*extra/httpd-default.conf*

```
# ServerTokens
# This directive configures what you return as the Server HTTP response
# Header. The default is 'Full' which sends information about the OS-Type
# and compiled in modules.
# Set to one of: Full | OS | Minor | Minimal | Major | Prod
# where Full conveys the most information, and Prod the least.
```

ServerTokens Full

将上面的这个配置项修改为：**ServerTokens Prod**

注：网上的配置方法 **ServerTokens ProductOnly** 是**错的**，没有后面那部分



这样这样只显示：

```
server:Apache
```

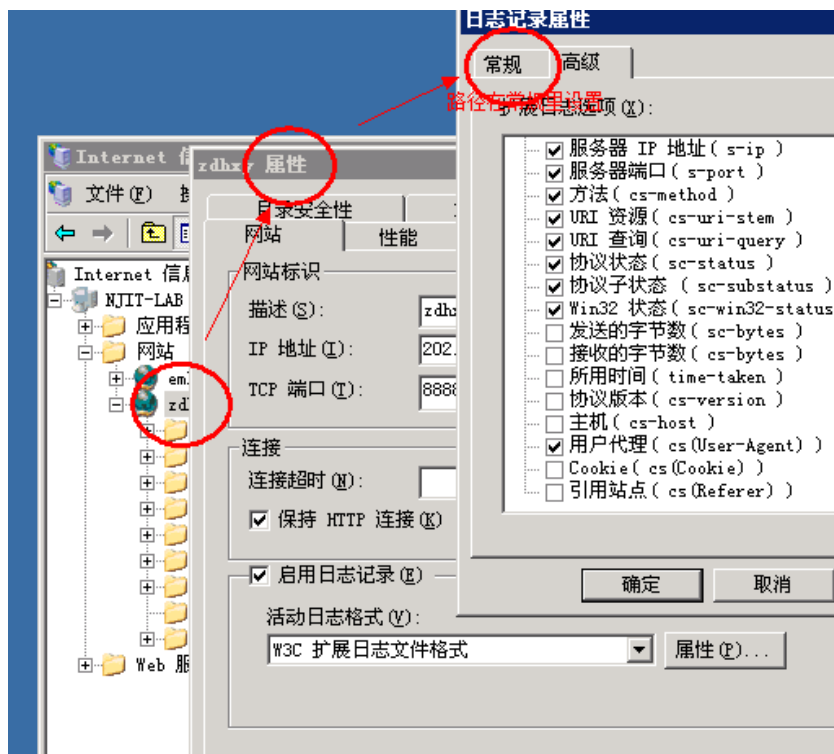
如果是 `ServerTokens Major` 则显示(`Minor` 则显示 `2.0`)：

```
server:apache 2
```

1.4 IIS 日志设置

访问量大的话，有话日志记录选项也是门学问呀

下图演示的是正常情况下需要记录的：



如果日志是按天来分记录文件，则不需要记录日期，文件名就是服务器日志的日期，所以只需要记录具体时间，访问量大的话可以节省硬盘空间，建议这么设置。

下列项是必须保留的：

客户端 ip,方法,uri 资源和查询，用户代理(这里指客户端的 User-Agent 头部)

下列是非必须的：

用户名，服务器端口，协议状态和子状态，cookie

日志要么不记录，记录的话一定要分文件，否则容易出现几个 GB 的大日志，等于没记录，而且还空耗资源。

1.5 nginx 的配置方法

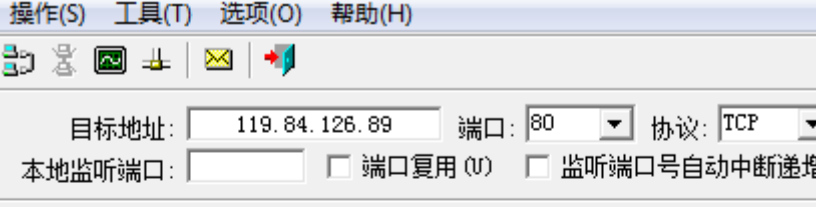
nginx 配置多网站说明，可以直接复制代码到 nginx 的配置文件：*nginx.conf*

```
server {
#监听端口
listen      80 default;
#域名
server_name localhost;
#日志
#access_log  logshost.access.log main;
#禁止解析到别的域名
server_name_in_redirect off;
#网站目录和索引文件
    location {
        root    E:/Web/forum/htdocs;
        index  index.html index.htm;
    }
#脚本语言
    location ~ .php$ {
#注销掉这个 root 路径
#        root    html;
#phpfastcgi 程序端口
        fastcgi_pass 127.0.0.19000;
#脚本文件索引
        fastcgi_index index.php;
#解析脚本的路径
        fastcgi_param SCRIPT_FILENAME E:/Web/forum/htdocs$fastcgi_script_name;
#无视
        include    fastcgi_params;
    }
#error_page 404          404.html;
#错误 50x 页面
error_page 500 502 503 504 50x.html;
    location = 50x.html {
        root    html;
    }
}
```

这份配置文件中，只包含了 `server{}` 部分，即单一网站配置部分，配置文件中另外有线程使用、连接数、存活时间等等，这些配置请依据服务器的具体情况进行具体配置。

1.6 Nginx 隐藏 http 头部中的版本信息

如下图，网站的 HTTP Header 中包含了 `Server : nginx/1.1.15` 的信息：

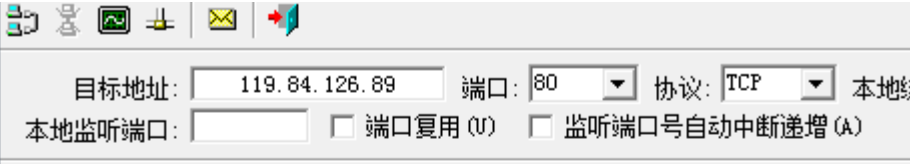


```
[2012-02-28 18:07:15.691] 发送指令:HEAD / HTTP/1.0
Host: 119.84.126.89
Accept: /*/*
Referer: http://119.84.126.89/
User-Agent: http://www.ccview.net
Connection: Keep-Alive

HTTP/1.1 200 OK
Server: nginx/1.1.15
Date: Wed, 29 Feb 2012 00:07:15 GMT
Content-Type: text/html
Content-Length: 9
Last-Modified: Tue, 28 Feb 2012 20:50:07 GMT
Connection: keep-alive
Accept-Ranges: bytes
```

隐藏方法：打开 `nginx.conf`

在 `http {` 后面加：`server_tokens off;` 这么一句即可。然后重启：



```
[2012-02-28 18:12:13.632] 192.168.0.12:53448 成功连接到 119.84.126.89:80

[2012-02-28 18:12:15.161] 发送指令:HEAD / HTTP/1.0
Host: 119.84.126.89
Accept: /*/*
Referer: http://119.84.126.89/
User-Agent: http://www.ccview.net
Connection: Keep-Alive

HTTP/1.1 200 OK
Server: nginx
Date: Wed, 29 Feb 2012 00:12:14 GMT
Content-Type: text/html
Content-Length: 9
Last-Modified: Tue, 28 Feb 2012 20:50:07 GMT
Connection: keep-alive
Accept-Ranges: bytes
```


2) Apache + PHP

2.1 Apache + PHP 高级静态伪装

找到 *apache* 的 *conf/httpd.conf* 文件并编辑, 搜索关键词: `AddType application/x-gzip`

应该可以找到如下配置项:

```
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
```

这里是 *gzip* 的类型的配置 (一般默认配置文件都有这个配置项, 即使服务器不支持 *gzip* 也会有, 不支持的配置项前面是用 *#* 注释掉的)

在最后添加一句:

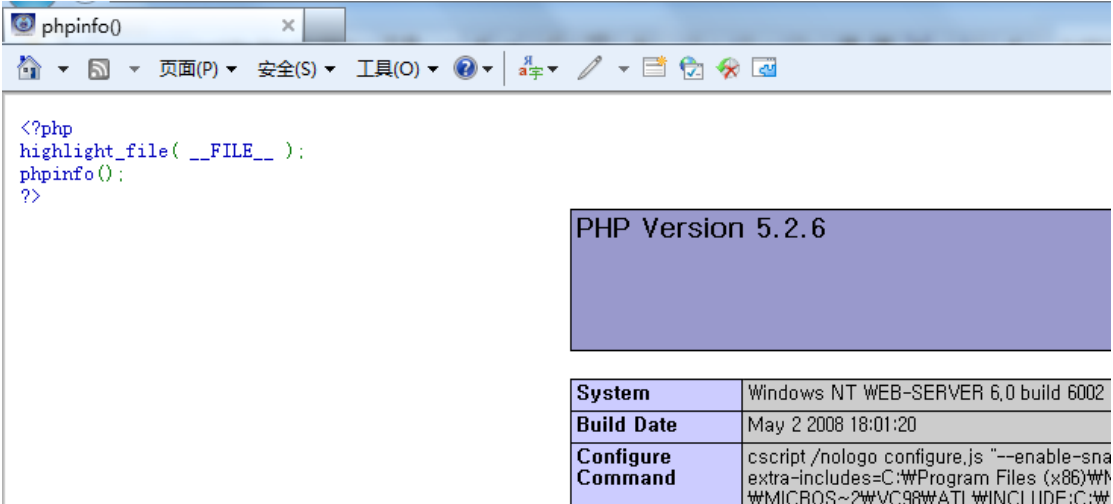
```
AddType application/x-httpd-php .abc
```

这里的 *.abc* 即要解析的文件后缀类型, 这里可以换成 *.html*, *.htm*, *.xoo* 和 *.jsp*, *.asp* 等等, 当然图片后缀也可以, 例如 *.gif*, 没试过, 但是如果果要改, 可以把默认配置的 *.gif* 项修改为 *x-httpd-php*

保存后重启网站容器, 测试结果:

```
<?php
highlight_file( __FILE__ );
phpinfo();
?>
```

保存为 *a.abc* 访问即生效, 此时可以把所有 *php* 改成 *abc* (文件内部链接地址要注意修改):



System	Windows NT WEB-SERVER 6,0 build 6002
Build Date	May 2 2008 18:01:20
Configure Command	cscrip /nologo configure.js "--enable-sna extra-includes=C:\Program Files (x86)\M MICROS~2\VC98\ATL\INCLUDE;C:\

补充:

设置为 *.gif* 后缀解析以后, 真正的 *gif* 文件访问正常, 但是 *php* 格式的 *gif* 后缀文件出现解析稍缓, 但是 *php* 改名 *gif* 以后可以正常解析为 *php* 的, 说明默认 *gif* 后缀的解析没有修改。

所以建议只做 *html* 伪装, 可增加 *google* 的收录情况, *Google* 收录 *gif* 的速度和数量远不如 *html* 页面的数量。

如果是虚拟主机，还可以这样配置：直接写入网站根目录的 `.htaccess` 文件中，代码如下：

```
<IfModule mod_setenvif.c>
  AddType application/x-httpd-php .php
  AddType application/x-httpd-php .html
  AddType application/x-httpd-php .htm
</IfModule>
```

第二行的 `.php` 一定不能去掉，去掉的话则 `Apache` 无法解析 `php` 文件

另外，如果使用 `.htaccess` 进行 `urlrewrite` 配置，除了确认 `AllowOverride None` 配置项已经更改为 `AllowOverride All` 之外，还要确认 `httpd.conf` 已经加载了 `rewrite` 模块。

如果上面的配置代码不能正常解析，可以用载入另一种 `mod` 的写法，写法类似：

```
<IfModule mod_mime.c>
  TypesConfig conf/mime.types
  AddType application/x-tar .tgz
  AddType application/x-rar-compressed .rar
</IfModule sapi_apache2.c>
  AddType application/x-httpd-php .php
  AddType application/x-httpd-php-source .phps
</IfModule>
</IfModule>
```

同样的，这里的 `.php` 后缀的解析不能够去掉，否则服务器的解析会出现不稳定、不解析等情况

2.2 禁用 PHP 敏感函数

`PHP` 的敏感函数可以在 `PHP` 的配置文件 `php.ini` 中设置。

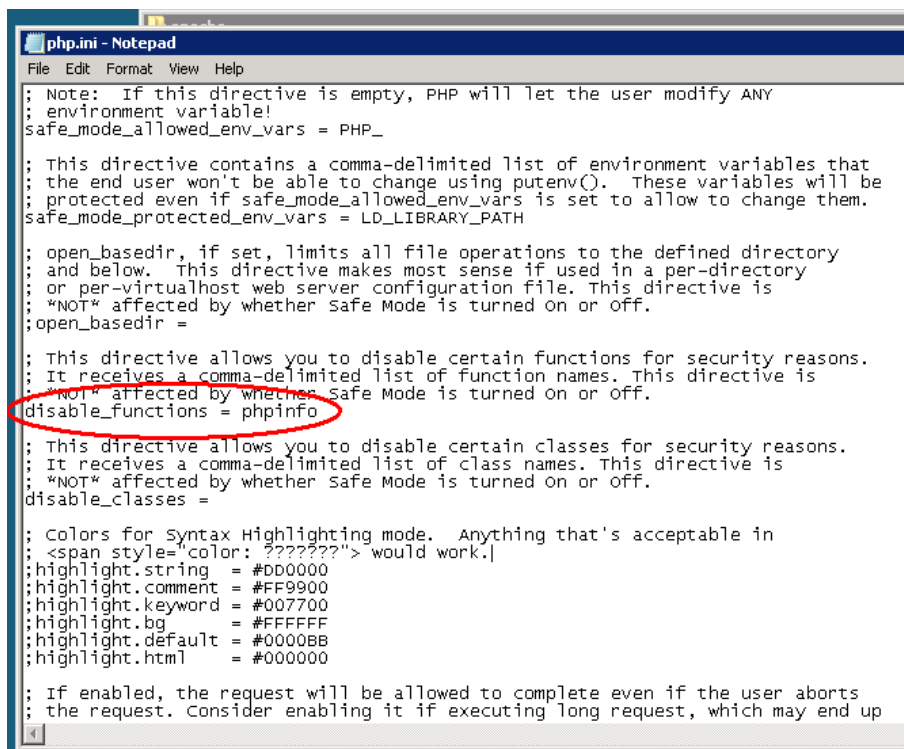
配置文件中直接有这么个禁用函数的功能，编辑 `php.ini` 找到 `disable_functions` 项，在配置项的等于号后面填上要禁用的函数即可。

禁用的函数中不需要再填写函数的括号，例如要禁用 `exec()` 函数，则不需要填写括号，只需要填写：

```
disable_functions = exec
```

如果是 `Windows` 系统下跑的 `PHP`，除了需要禁用所有的敏感函数以外，还要把 `dl` 函数也禁用。因为 `Windows` 系统目录权限的问题，禁用 `dl` 函数可以防止黑客通过写入自定义 `php` 扩展链接库，并载入链接库进行提权。

下图是演示禁用 `phpinfo` 函数的示范：



```
php.ini - Notepad
File Edit Format View Help
; Note: If this directive is empty, PHP will let the user modify ANY
; environment variable!
safe_mode_allowed_env_vars = PHP_

; This directive contains a comma-delimited list of environment variables that
; the end user won't be able to change using putenv(). These variables will be
; protected even if safe_mode_allowed_env_vars is set to allow to change them.
safe_mode_protected_env_vars = LD_LIBRARY_PATH

; open_basedir, if set, limits all file operations to the defined directory
; and below. This directive makes most sense if used in a per-directory
; or per-virtualhost web server configuration file. This directive is
; "NOT" affected by whether Safe Mode is turned on or off.
;open_basedir =

; This directive allows you to disable certain functions for security reasons.
; It receives a comma-delimited list of function names. This directive is
; "NOT" affected by whether Safe Mode is turned on or off.
disable_functions = phpinfo

; This directive allows you to disable certain classes for security reasons.
; It receives a comma-delimited list of class names. This directive is
; "NOT" affected by whether Safe Mode is turned on or off.
disable_classes =

; Colors for Syntax Highlighting mode. Anything that's acceptable in
; <span style="color: ???????"> would work.
;highlight.string = #DD0000
;highlight.comment = #FF9900
;highlight.keyword = #007700
;highlight.bg = #FFFFFF
;highlight.default = #0000BB
;highlight.html = #000000

; If enabled, the request will be allowed to complete even if the user aborts
; the request. Consider enabling it if executing long request, which may end up
```

另外，Windows 下使用 Apache 做容器的话，即使 PHP 禁用了其他敏感函数而不禁用 *dl* 函数，黑客其实不需要使用自建的 PHP 链接库，只需要使用 *dl* 函数载入 PHP 原始的动态链接库，就可以重新使用被禁用的函数，所以，如果禁用了敏感函数而不禁用 *dl* 函数等于白搭。

这里也给我们一个提示：

在 Windows 的 Apache + PHP 下可以将自建 PHP 动态链接库作为后门。

函数的结尾不需要加分号；函数与函数中间以逗号相隔，间隔无空格，例如禁用 *phpinfo* 以后，再使用这个函数就会：

Warning: phpinfo() has been disabled for security reasons in XX.XX line x

这里不遵循上面的原则，就会显示报错。

这里是 PHP 的一份敏感函数列表：

```
System()
shell_exec()
passthru()
exec()
popen()
proc_open()
allow_url_fopen()
fsockopen()
pfsockopen()
```

还有变态一点的禁用(直接复制)：

```
allow_url_fopen,apache_child_terminate,apache_get_modules,apache_get_version,apache_getenv,apach
e_note,apache_setenv,chgrp,chown,closelog,dbmopen,debugger_on,debugger_off,define_syslog_variabl
es,dl,dll,error_log,escapshellcmd,escapshellarg,exec,fsockopen,ftp,ftp_exec,fpassthru,ini_alter,leak,l
ink,listen,ln,lynx,myshellexec,readlink,shell_exec,show_source,symlink,system,ocinumcols,openlog,pass
thru,pcntl_exec,pclose,pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifsto
pped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dis
patch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pc
ntl_exec,pcntl_getpriority,pcntl_setpriority,pfsockopen,popen,proc_open,proc_close,proc_get_status
,proc_nice,proc_terminate,prus,popen,posix_getpuid,posix_kill,posix_mkfifo,posix_setsid,posix_setui
d,posix_setpgid,readfile,show_source,shell,socket_bind,suexec,symlink,syslog,system,virtual,wget
```

*上面部分函数具体用法和作用未知

2.3 Apache 禁用文件索引

避免 Apache 自动文件所以，防止遍历目录漏洞的发生。依然是在 Apache 的配置文件中，找到下图中的位置：

```
# This should be changed to whatever you set DocumentRoot
#
<Directory "C:/xampp/htdocs">
    #
    # Possible values for the Options directive are "No
    # or any combination of:
    #   Indexes Includes FollowsSymLinks SymLinksifOwnerE
    #
    # Note that "Multiviews" must be named *explicitly*
    # doesn't give it to you.
    #
    # The Options directive is both complicated and imp
    # http://httpd.apache.org/docs/2.2/mod/core.html#op
    # for more information.
    #
    Options Indexes FollowsSymLinks Includes ExecCGI

    #
    # AllowOverride controls what directives may be pla
    # It can be "All", "None", or any combination of th
    #   Options FileInfo AuthConfig Limit
    #
    AllowOverride All
#
```

将 *options Indexes* 中的 *Indexes* 去掉，重启 Apache 即可。

也可以在 *.htaccess* 文件(放置于网站根目录)中加入这句话：

```
Options -Indexes
```

不需重启 Apache 就不会自动列文件目录了。

2.4 虚拟主机配置文件防读取

如果网站中不能避免使用 *MySQL* 的 *root* 账户，这是 *Apache* 多站点配置防止因注入读取 *Apache* 配置文件的方法（*Windows* 为例）。

在 *httpd-vhost.conf* 中的虚拟主机配置文件单独拿出做成 *xx.conf* 的配置文件，然后 *vhost* 的配置语句这样写：

```
Include "D:/Apache/vhosts/[^.#]*"
```

这样的话，在 *apache/vhosts* 中就可以这样配置了：

```
2/htdocs/vhosts/[^.#]*"
```



MySQL 的注入可以使用 *load_file()* 函数以 *text* 格式读取文件，但是这个函数不能列目录，所以通过这个方法防止配置文件被读取，这样最大的好处其实还是方便对服务器上的网站进行配置管理。

不过最好的办法是设置 *MySQL.user* 表中的账户 *file_priv* 字段取值为 *N*。这样最大的好处其实是方便管理。

2.5 禁用危险的 php socket

php 的 *Socket* 函数可以在禁用 *PHP* 敏感函数的情况下反弹 *Web* 权限下的 *cmd* 然后来提权，禁用方法先打开 *php.ini*，找到这里：

```
extension=php_sockets.dll
```

在前面注释掉就 *OK* 了。否则，如果是 *Windows* 服务器，就会导致这样的悲剧发生：



2.6 防止 php 跨目录读写其他文件

防止 PHP 跨目录可以设置 `php.ini` 的安全模式：

```
safe_mode = On
```

然后 `Gid` 为 `off`：

```
safe_mode_gid = Off
```

最后再设置 `basedir` 即可，此目录设置是以最后一条斜杠为准

例如设置限定的目录是 `x:/a/b/c`，实际限定访问目录为 `x:/a/b` 下。

若配置为 `x:/a/b/c/`，则实际限定访问目录为：`x:/a/b/c`

```
safe_mode_exec_dir =E:/Web/forum
```

`safe_mode_include_dir` 可不配置，另外，`dir` 目录 `windows` 用分号，`linux` 用冒号

2.7 禁止特定目录执行 CGI 脚本

在当前目录添加 `.htaccess` 文件，内容：

```
Options -ExecCGI
```

```
AddHandler cgi-script .php .php5 .pl .py .jsp .asp .aspx .shtml .sh .cgi .sql .rb
```

访问禁用的脚本后缀将显示 500 错误

2.8 对特定文件进行登录验证

在当前目录添加 `.htaccess` 文件，内容：

```
AuthType Basic
```

```
AuthName "Silic Group Hacker Army"
```

```
AuthUserFile /home/blackbap/blackbap.org/bbs/.htpasswd
```

```
Require valid-user
```

然后在上面的 `.htpasswd` 文件：

```
#silic:bbs
```

```
silicbbs:532WkhU9SF/iQ
```

```
#bbs:silic
```

```
bbs:85QINg/2GZWGY
```

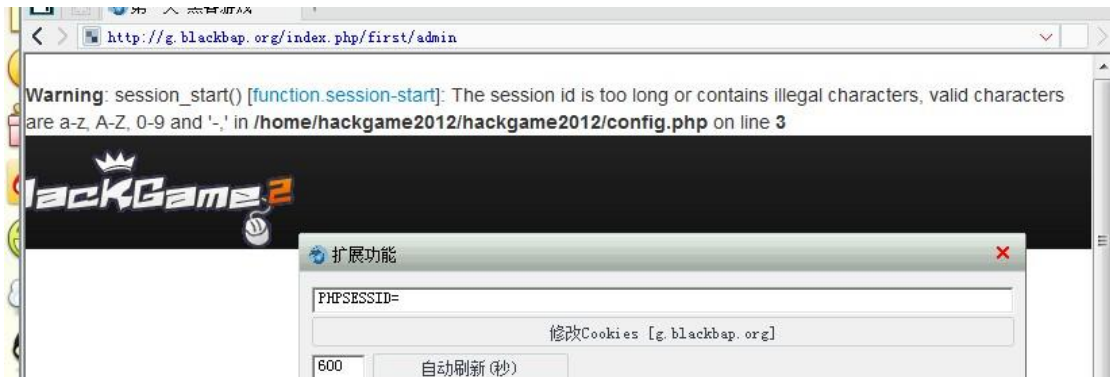
2.9 PHP 非法 Session 值爆物理路径相关

PHP 如果没有设置屏蔽错误回显，或者设置错误报告登记低，则当客户端提交一个自定义的 `session`，并且这个 `session` 是含有恶意非法字符的时候，`php` 文件就会报错，并爆出 `php` 文件的物理路径。

本问题发现于习科闯关游戏 2012 的黑盒内测期间，闯关游戏第一关的后台登陆的地方 = =

<http://q.blackbap.org/index.php/first/admin>

这个页面中，把 cookie 改成 `PHPSESSID=123'`;;; 然后刷新下，PHP 的错误回显就将 php 的物理路径给爆出来了：



The session id is too long or contains illegal characters, valid characters are a-z, A-Z, 0-9 and '\',\' in...

Config.php 第三行是 `session_start();`

解决方案，只要将这个函数修改为：`@session_start();`

php 函数前面加 at 符号 "@" 的意思是处理过程中有错误不提示。

2.10 Apache 索引目录中文乱码解决方式：

Index of /pdf

Name	Last modified	Size	Description
Parent Directory		-	
A variant of IIS alg...>	25-Nov-2012 03:10	73K	
Attack and Defense.pdf	25-Nov-2012 03:10	169K	
CMUcam2 Graphical Us...>	25-Nov-2012 03:10	668K	
DES Controller Synth...>	25-Nov-2012 03:10	117K	
Detecting DDoS Attac...>	25-Nov-2012 03:10	47K	
Digital UNIX.pdf	25-Nov-2012 03:10	438K	
Distributed Password...>	25-Nov-2012 03:10	246K	
Efficient Collision ...>	25-Nov-2012 03:10	201K	
Finding Collisions i...>	25-Nov-2012 03:10	212K	
Getting Started Unix...>	25-Nov-2012 03:10	3.7K	
How to Hack Java Lik...>	25-Nov-2012 03:10	164K	
Introduction to Data...>	25-Nov-2012 03:10	343K	
Mobile Computing.pdf	25-Nov-2012 03:10	7.7K	
Network Security.pdf	25-Nov-2012 03:11	4.0M	
Security Problems in...>	25-Nov-2012 03:10	59K	
Security-Lecture 19.pdf	25-Nov-2012 03:10	246K	
The rsync algorithm;...>	25-Nov-2012 03:10	149K	
Web Hacking.pdf	25-Nov-2012 03:11	1.5M	
WesternCivilizationD...>	25-Nov-2012 03:11	3.1M	
Writing Your Own Uni...>	25-Nov-2012 03:10	30K	
linux-kernel.pdf	25-Nov-2012 03:11	2.0M	
nas12*ï;¼Ö*á.pdf	25-Nov-2012 03:10	107K	
"ÖÖ°E-uÄE*ÖÖEg"Ä...>	25-Nov-2012 03:11	2.9M	
"pî;EEO» Hash Visua...>	25-Nov-2012 03:10	905K	
%I*é Ö*E*EFA*%...>	25-Nov-2012 03:11	2.3M	
%I_Öfo*%ÄN°Öbug.pdf	25-Nov-2012 03:10	932K	
ÄÄ*ü*ÄMüNÄ*ÉüÖ*ITÄ...>	25-Nov-2012 03:10	208K	
ÖÄ»S*îE"IPuÄuÄiDUs...>	25-Nov-2012 03:10	376K	

当我们的目录设置为：

```
options Indexes FollowSymLinks
```

这个时候，是故意在某个目录无索引文件遍历目录，但是 Apache 默认会将中文显示为西欧乱码，解决方法是再添加一句设置：

```
IndexOptions Charset=GB2312
```

Index of /pdf

Name	Last modified	Size	Description
Parent Directory		-	
A variant of IIS alg...>	25-Nov-2012 03:10	73K	
Attack and Defense.pdf	25-Nov-2012 03:10	169K	
CMUcam2 Graphical Us...>	25-Nov-2012 03:10	668K	
DES Controller Synth...>	25-Nov-2012 03:10	117K	
Detecting DDoS Attac...>	25-Nov-2012 03:10	47K	
Digital UNIX.pdf	25-Nov-2012 03:10	438K	
Distributed Password...>	25-Nov-2012 03:10	246K	
Efficient Collision ...>	25-Nov-2012 03:10	201K	
Finding Collisions i...>	25-Nov-2012 03:10	212K	
Getting Started Unix...>	25-Nov-2012 03:10	3.7K	
How to Hack Java Lik...>	25-Nov-2012 03:10	164K	
Introduction to Data...>	25-Nov-2012 03:10	343K	
Mobile Computing.pdf	25-Nov-2012 03:10	7.7K	
Network Security.pdf	25-Nov-2012 03:11	4.0M	
Security Problems in...>	25-Nov-2012 03:10	59K	
Security-Lecture 19.pdf	25-Nov-2012 03:10	246K	
The rsync algorithm□...>	25-Nov-2012 03:10	149K	
Web Hacking.pdf	25-Nov-2012 03:11	1.5M	
WesternCivilizationD...>	25-Nov-2012 03:11	3.1M	
Writing Your Own Uni...>	25-Nov-2012 03:10	30K	
linux-kernel.pdf	25-Nov-2012 03:11	2.0M	
nas12参考手册.pdf	25-Nov-2012 03:10	107K	
复杂影片的全局运动模...>	25-Nov-2012 03:11	2.9M	
哈希可视化Hash Visua...>	25-Nov-2012 03:10	905K	
进程组分布式计算方法...>	25-Nov-2012 03:11	2.3M	
静态分析工具寻找bug.pdf	25-Nov-2012 03:10	932K	
美国海军研究生院《计...>	25-Nov-2012 03:10	208K	
用户和权限的命令执行Us...>	25-Nov-2012 03:10	376K	

这里就将西欧乱码识别为 `gb2312` 编码的简体中文了。

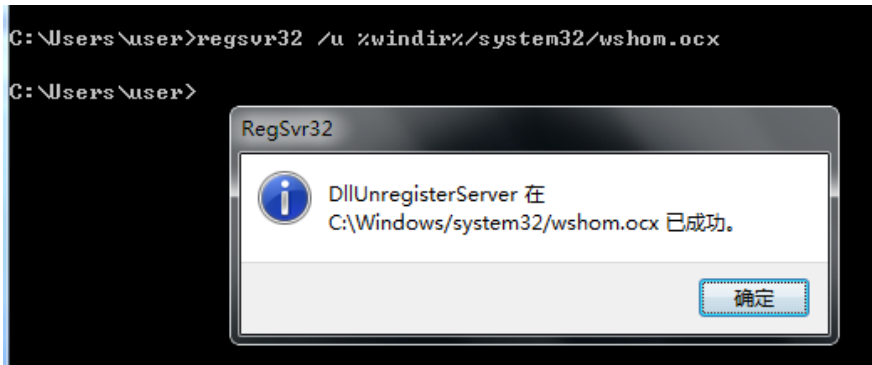
3) 服务器安全

尚未建立 Linux 服务器安全相关文档，会完善

3.1 卸载(关闭)Wscript.Shell

禁用 Wscript.Shell 组件，直接在 cmd 中执行以下命令即可：

```
regsvr32 /u %windir%/system32/wshom.ocx
```



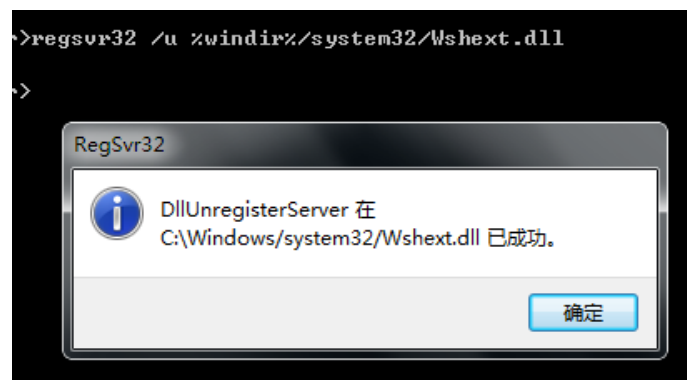
图为卸载过的

卸载(关闭)FSO : `regsvr32 /u %windir%/system32/sccrun.dll`

卸载(关闭)Shell-application : `regsvr32 /u %windir%/system32/shell32.dll`

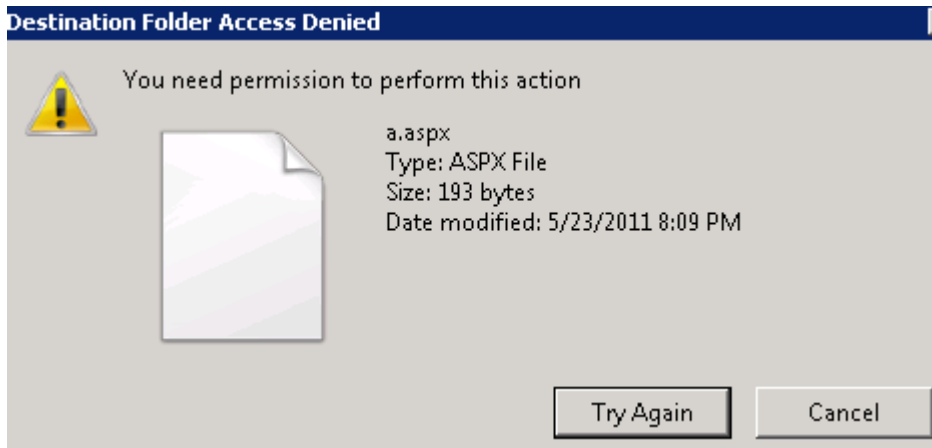


卸载(关闭)Wscript.network : `regsvr32 /u %windir%/system32/Wshext.dll`



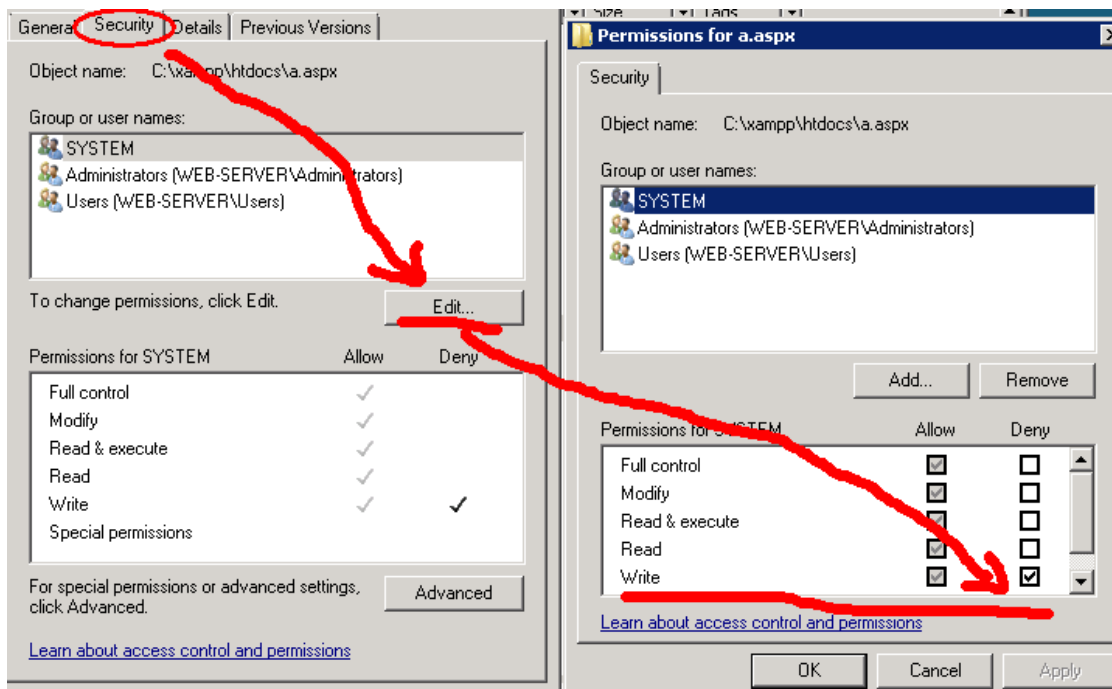
3.2 创建和清除特殊属性的文件

清理一些 webshell 时候会发现无法删除，这就可能是被黑客留了后门，如图：



即使使用 cmd 去删也仍然拒绝访问，这就最可能的是设置了用户组权限。

设置权限有两种，一种是这样的：



右键这个文件，属性，安全，编辑

把 SYSTEM 用户组和 Administrators/Users 用户组都设置为拒绝写入，就删不掉了。要想删掉，只要把拒绝写入去掉就行了。

另一种是在 cmd 下加 system 权限：

attrib +a +r +s +h 文件路径

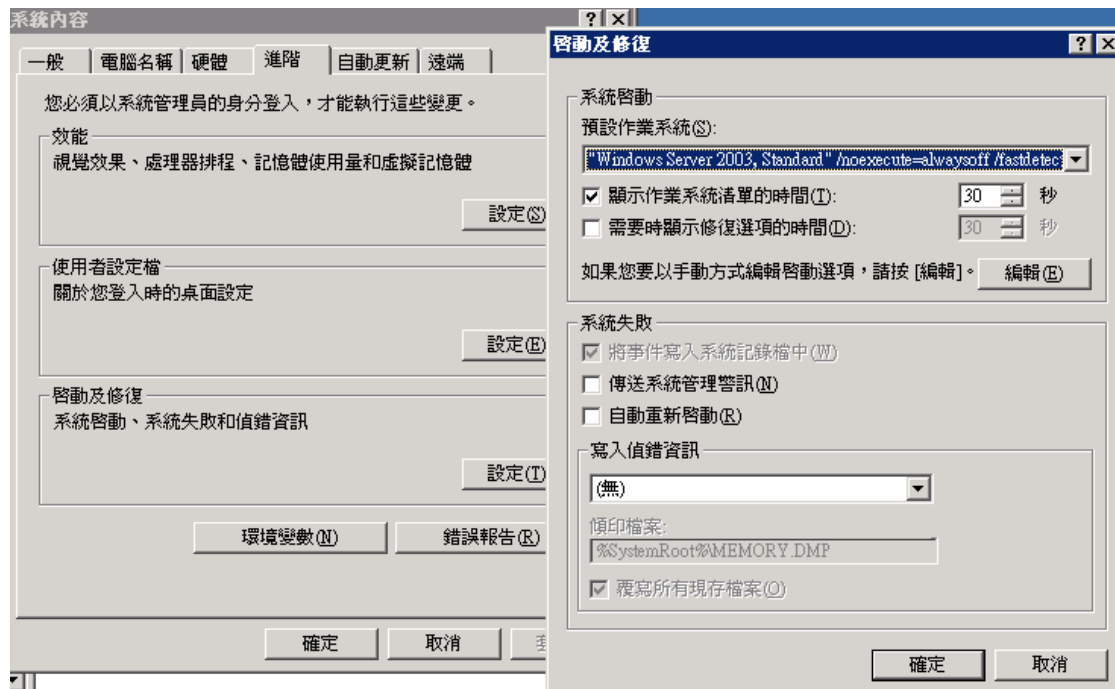
去掉这个权限只需要重复执行上面的 cmd 命令，但需要将 加号 + 修改为 减号 - 执行

3.3 服务器时间查看器安全性日志失效

失效方法：去掉日志记录文件的 SYSTEM 继承权

首先去掉系统管理日志记录：

右键“我的电脑”“属性”“高级”选项卡，“启动及修复”设定如下配置：

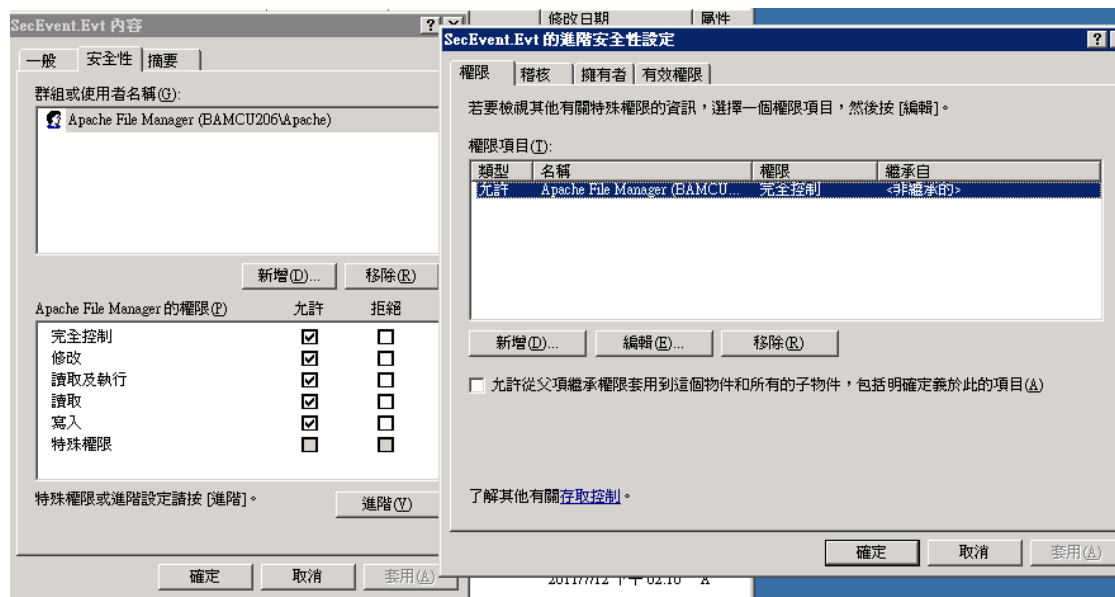


然后找到路径：`C:\windows\system32\config`

安全性的日志是：`SECURITY.Evt`

2008 的路径：`C:\Windows\System32\winevt\Logs`

最后 右键该文件的“属性”，如图设置：



将“允许从父项继承权限用到这个文件的所有子文件，包括明确定义于此的项目”选项去掉
在新增那里增加一个非法的用户，然后应用，将非法用户权限设置为全部，将 SYSTEM 和
Administrator 的权限全部去掉，日志就无法记录和访问了。

4) 数据库安全相关

4.1 Mysql.user 安全设置

MySQL 数据库中自带名为 mysql 的系统表，其中的 db 表段中设置着 MySQL 账户对哪些数据库有控制权。

在 User_info 表段（默认没有这个表）中要注意，有时会有账户信息泄露。

在 User 表中有 mysql 所有的账户信息，尤其要注意 file_priv 字段，最好直接设置默认为 N，root 也可以设置为 N，虽然 root 设置为 N 有点鸡肋，但是可以阻挡一部分菜鸟脚本小子。还要注意 host 字段尽量避免通配符 “%” 的设置。

自定义函数表 Func，要注意 udf 提权自定义函数，有内容直接删，还要清理 dll(确认 func 中无自定义函数设置 func 为只读即可)。黑客通常会在这个表里面留后门。

MySQL 数据库可以直接所表，但是暂未测试 mysql 库锁表后能否突破及其安全性。

4.2 MySQL 修改工作端口

MySQL 的默认工作端口为 3306，但是修改方法很简单，找到 MySQL 安装目录的配置文件 my.ini，在文件中有这么个配置语句：

```
[mysqld]
# The TCP/IP Port the MySQL Server will listen on
port=3306
```

修改一下就可以了，例如：

```
[mysqld]
# The TCP/IP Port the MySQL Server will listen on
port=53306
```

重启 MySQL 才能生效。但是修改后 PHP 等得连接语句中 host 主机配置也应该做相应修改：

localhost 应改为 *localhost:53306*