TURING

6

2013/06/18

码农

codeMaker()

网络“安全”

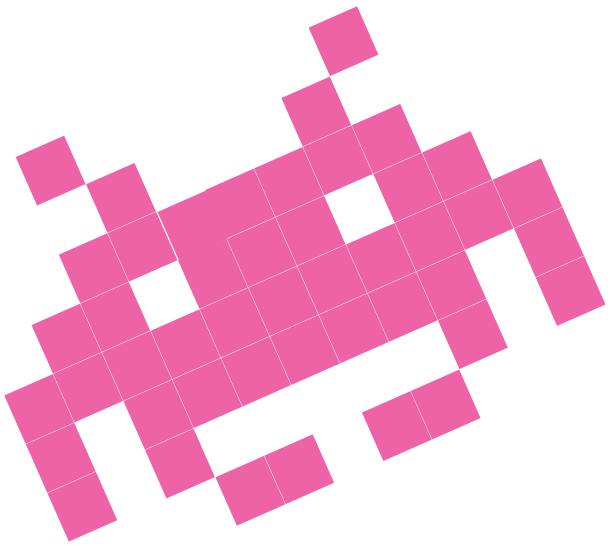
安全？看看谁是偏执狂

针对无线网络的攻击

Android数据安全

解密iOS越狱

人物：女码农特辑



目　录

鲜阅

出版的未来

八++

书榜

成书手记

编者的话

无知有害

编者 / 李盼

我们都爱互联网，一千年后，我们身上最大的标签很可能就是“开始上网”的一代。作为这个陌生领域的初探者，我们和所有近乎无知的人一样，还不知道自己正身处一个危机四伏的原始森林。

网络时代的魔法师——码农们，大部分都是勤勤恳恳、努力工作的老实人。他们对黑魔法知之甚少，可是这并不代表地下生态圈停止了发展。更糟糕的是，现在世界范围内的司法系统无法让这些邪恶的魔法师们接受应有的制裁，他们中大多数人变成了低调的法外之徒。而大额的利润正吸引越来越多的边缘人加入到黑客的行列中。

作为一个善良的麻瓜，假如你不知道病毒、网络钓鱼以及类似的风险是怎么回事，那你受害的机率一定会大大增加；假如你不知道每天使用的无线网络中也有藏龙卧虎，那你泄露的很可能比自己想象的还要多；假如你对智能手机操作系统的安全性没有足够认识，那你就会在不知不觉中引狼入室。值得庆幸的是，作为麻瓜，你伤害最深的可能只是自己。

可是如果善良的码农们对安全问题没有清醒的认识时，他们将置整个世界的安危于不顾。我相信枪是不会杀人的，只有人才会杀人。所以善良的码农们，拿起你们的武器来，无知有害，持枪有理。

专题：网络“安全”

万维网安全？看看谁是偏执狂

无论你是总统，还是平民百姓，都应该了解这个计算机世界，因为它对每个人都有切身影响。无论工作和生活与技术距离有多遥远，你总有机会接触技术和搞技术的人。了解一些计算机和通信的常识都将对你大有助益，最低限度也能让你知道推销人员或服务热线什么时候向你隐瞒了事实。没错，无知有害。万维网扩展了我们的生存空间，但也前所未有地让我们在陌生人面前暴露无遗，因此伤害也可能不期而至。

作者 / Brian W. Kernighan

Brian W. Kernighan 是世界顶尖的计算机科学家，在贝尔验室工作 30 多年，与 Unix 系统的创造者 Ken Thompson 和 Dennis Ritchie（C 语言发明人）长期共事，对 Unix 系统也做出了突出贡献。他与 Dennis Ritchie 合著的 C Programming Language 是世界上第一本被广泛认可的 C 语言教程，平实、优雅、简洁，已成为编程语言教程中的绝佳典范，被称为 “K&R C”。他还发明了 AWK 和 AMPL 编程语言。

译者 / 李松峰、徐建刚

万维网引发了很多安全难题。一般来说，可以把万维网遇到的安全威胁分成三类：对客户端（也就是使用浏览器的你）的攻击、对服务器（例如网上商店或者银行帐户）的攻击和对传输中信息的攻击（比如窃听无线网络）。我们将在本节依次讨论这三种情况，以弄清问题出在哪里、如何减轻危害。

对客户端的攻击

对你的攻击不仅包括垃圾邮件、跟踪等惹人讨厌的攻击，还包括泄露你的信用卡和银行账号等私密信息、假冒你花掉你的钱等更严重的攻击行为。

后面（本文之外）我们会详细讨论 cookie 和其他跟踪机制如何监控你的上网行为、给你发送看似吸引人从而不是那么烦人的广告。实际上，通过禁用第三方 cookie（也就是从别的网站而不是你正在访问的网站发送过来的 cookie），就可以堵住大部分跟踪攻击。如果再使用其他浏览器插件来禁止跟踪程序运行、关闭 JavaScript、禁用 Flash，采取这些防护手段就能把跟踪和广告的数量限制到可控的程度。也许有时你会嫌这些防护手段麻烦，因为这样披挂上阵会导致有的网站无法访问。遇到这种情况时，就要把防护的级别临时调低一些，但要记得访问完这些网站后再复原。

垃圾邮件是不请自来的邮件，其内容通常是发家致富秘籍、洗髓易筋偏方、职场蹿升宝典和很多没用的其他商品和服务。垃圾邮件业已泛滥成灾，严重影响了电子邮件的正常使用。我自己通常每天收到 50 到 100 封垃圾邮件，远比真正的邮件多得多。据可信调查称，所有电子邮件中的 90% 是垃圾邮件。最近我看到一篇文章则说，万维网上每天发送的垃圾邮件多达 2500 亿封。垃圾邮件之所以如此泛滥，主要原因在于发送成本极低（上面那篇文章里说，

发送 100 万封垃圾邮件只要 80 美元成本）。就算几百万收件人里回复的只占个零头，也够那些发件人赚回来了。

可以使用垃圾邮件过滤器对邮件进行分拣，去芜存菁。常用过滤方法有查找已知的垃圾邮件模式（比如“你想每天都收到 eBay 寄来的钱吗？”这样的邮件，一看就是群发给大量人群的格式邮件），发现不太可能出现的名字和离奇的拼写（比如 /I/-\GR/-\）[1]，或者把经常发送垃圾邮件的地址列入黑名单，拒收它的邮件。只用一种办法来过滤是远远不够的，因此要打出组合拳。垃圾邮件过滤仿佛是军备竞赛，只要防守者道高一尺，学会防范某一类垃圾邮件，进攻者就会魔高一丈，发明出新的规避方法。

1 “/I/-\GR/-\”是用象形字符拼写的一种药物名 Viagra（Viagra 在中国大陆正式译名“万艾可”，更广为人知的民间译名是“伟哥”），这么拼写是为防止关键词过滤。

——译者注

要想从源头制止垃圾邮件非常困难，因为垃圾邮件的源头通常都极其隐蔽。很多垃圾邮件是通过被入侵的个人计算机（通常运行 Windows 系统）发送出来的。由于存在安全漏洞，再加上用户疏于管理，这些计算机很容易被装上**恶意软件**（malware）。有一种恶意软件会在收到上游控制指令时，群发垃圾邮件，而它的上游计算机也可能进一步受更上游控制。每多一步都会使发现垃圾邮件的源头更加困难。

网络钓鱼（phishing）攻击则通过骗取收件人的信任，让他们主动奉上窃贼需要的信息。你十有八九收到过“尼日利亚骗局”式的诈骗邮件。真的会有人相信这些天方夜谭，给骗子回信吗？听起来不可思议，但确实一直有人上当。钓鱼攻击比这更狡猾，因为它们发来的邮件看起来就像来自正规金融机构甚至你的朋友们。试想，如果你收到一封道貌岸然的邮件（像是你熟知的公司发来的）让你核实自己的证件……如果你照着做了，坏蛋就会知道你的某些信息，以此来窃取你的钱财或者套取你的真实身份。几年前，这可是相当前卫的玩法。我曾收到这样的一封钓鱼信，其中的文法谬误和错别字出卖了它，当然我的名字不叫比尔，也没有花旗银行帐号。

发件人：花旗银行 cash@citibank.com

收件人：bill@research.bell-labs.com

主题：请更新您的花旗帐号

亲爱的花旗客户： 我行最近发现有人从国外 IP 地址频繁尝试登陆您的花旗银行账号，我们一定能相信这是在试图用暴力破角您的密码。登陆成功没有，您的账户已受到全面保护。如果您最近在旅行中访问了自己的账户，这些异常的登陆也许是您自己。

发起登陆的 IP 地址是 173.27.187.24，ISP 主机是 cache-82.proxyserver.cis.com

我们使用了多种技术来核兑用户在我们网站注册时填写信息的准确性。然而，油于通过互联网核实用户有困难，花旗银行无法做到也并不会确认每位用户所声称的身份。因此，我们起用了一套离线验证系统帮您评估与您交易的是谁。这套系统叫做旗安宝，是花旗银行迄今为止最安全的钱包。如果您是此账户的合法人，只要点击下面的连接，填写表单并提交，我们就会核实您的身份并为您免费注册旗安宝系统。这样，您在我行的所有账户都会受到保护，远离遭受欺诈的风险。

只要轻轻一点，远离欺诈风险！

为了让 Citibank.com 成为最安全的网站，每一位用户都要被注册到旗安宝。

注意！如果您不理会我们的请求，我们在别无选择之下将不得不暂停您的账号。

另，请勿回复此邮件，这个账号不能收邮件。

此致，敬礼。花旗银行客户支持部。

要编造一封看起来正规的邮件是很容易的，版式和图片都可以从真的网站上复制。使用无效的回信地址也很容易，因为邮件系统发件时并不检查发件人的真实性。那个链接下面的 IP 地址指向一个美国之外的网站，而我是在检查了 HTML 源代码后用 traceroute 才发现这一点的。和垃圾邮件一样，钓鱼也几乎不花任何成本，即便上钩的人少之又少，也足够骗子们赚个盆满钵溢了。

更新式的攻击则更难发现。2010 年末，我收到了一位朋友发来的这封信，该信是群发给一个很小的私密邮件列表的：

> 日期：Wed, 1 Dec 2010 04:55:37 -0800 (PST) 发件人：查尔斯 [...] <[...]@gmail.com>
>
> 实在不好意思酱紫打扰你，但是我刚遭遇了人在囧途之英国囧，包包被偷，护照和银行卡随之不见。幸好大使馆为人民服务，给我搞了张临时护照。我现在得买张票，还要把旅馆账单结了。
>
> 就不跟你客套了，现在手头没钱，跟银行联系过，他们说寄给我一张新卡，最快也要 2 到 4 天才能到手，还得是万恶的工作日！我看还是跟哥们儿你借点闲钱吧，回来马上还你，我得去赶几个小时之后的末班飞机。
>
> 一会儿告诉你怎么打钱给我。你可以发我的 AOL 帐号 ([...]@aol.com)，我拿手机登在上面刷着，要不打旅馆的座机也行，号码是 +447045749898。
>
> 万分感谢！
>
> 查理

看上去真是像模像样啊。查理是个驴友，在外旅行是常态，我已经有好几个星期没见过他了。于是我满怀疑虑地打了那个电话，没人接，有点奇怪。然后我回了那封信，收到他的回复：

嗨，布莱恩。

我有点小麻烦，借我 1800 美元吧。用西联打钱就行，需要汇款操作细节的话你就说。回国以后马上还你。

查理

我问他要了汇款指南，同时也问了个问题，只有查理本人才能答出来。他回给我西联的详细操作步骤，却没搭理我的那个问题。我顿时疑心大起，马上搜索了一下那个电话号码，发现是一起知名的骗局，已经连骗了好几个月。骗子竟然连剧情和电话号码都懒得换。后来我得知，查理的 Gmail 帐号遭遇过入侵，攻击者把他的联系人名单作为下手目标，并了解到他的一些私人信息。因此，这种精确瞄准的攻击方式有时叫做**鱼叉式网络钓鱼**（spear phishing）。鱼叉式网络钓鱼是一种社会工程方法，即假装是受害人的私交密友，或者冒称同事，诱使受害人犯傻。

间谍软件（spyware）是指运行在你的计算机上、会把你的信息送到别处去的程序。有些间谍软件显然是恶意的，但有些只是为了商业目的而私自收集用户信息。例如，我有几台笔记本预装了包探测器程序 pinger，该程序开机自动运行，然后定期连接到厂家的服务器。至少在我某天发现并将之禁用之前，它一直在这么做。我相信如果质问厂家的话，他们一定会告诉我这个包探测器只是向他们报告我机子持续运行时的健康状态并检查软件更新。问题是：我从没收到关于这个程序的说明，也没见过打开这个功能的选项，更不要说给我机会关掉它了。

还有一个臭名远扬的例子。索尼公司在 2005 年卖出过一批音乐 CD，其中使用了自动运行技术，能在 Windows 系统上安装间谍

软件：光盘插入计算机后，就会悄无声息地自动安装监控程序。这个间谍软件用了一些手法来隐藏自己，这说明某些人也知道这样做不够厚道。这个程序的实现有缺陷，会让计算机完全开放，导致任何人都能安装软件。索尼这样做想必并非出于恶意，而只是为了防止非法复制。但这个程序发现计算机播放音乐时，不管是不是索尼发行的，都一股脑报告给索尼。这个做法曝光之后，索尼的反应是发布了一个让安全性变得更加糟糕的卸载程序。

此外，入侵者往往会在个人计算机上安装**僵尸**（zombie）程序，这类程序平时潜伏着，一旦控制者从互联网发来指令，就会发作并执行诸如发送垃圾邮件等攻击。这些程序一般也叫**肉鸡**（bot），控制它们的各个网络通常叫**僵尸网络**（botnet）。万维网上已知的僵尸网络有上千个，可能控制了几十万到上亿台的肉鸡待命出击。

攻击者入侵客户端计算机之后，就可以查看其文件系统，或者悄悄安装**按键记录器**抓取用户输入的口令和其他数据，这样就能从源头窃取信息。按键记录器是在客户机上监控所有按键行为的程序，所以能抓到用户输入的口令。而且在这种情况下，加密也没用[2]。这种程序也可能开启计算机的话筒和摄像头。

如果浏览器或别的软件有缺陷，导致不怀好意的人在你的机器上安装了他们的软件，风险就更大了。浏览器程序庞大而复杂，存在种种可能导致用户遭受攻击的缺陷。为此，要让浏览器始终更新至最新版本，还要配置浏览器让它不要向外发送不必要的信息，也不要允许随意下载。另外，请注意你自己下载回来的东西，不要仅仅因为网页或程序让你点鼠标，就盲目去点。稍后我们将讨论更多的防御措施。

2 有些网站为了防止按键记录器窃取用户敏感信息，自己开发了 ActiveX 控件处理口令输入，但这样做有严重的负面影响。参见：http://t.cn/h5VPkQ。——译者注

对服务器的攻击

对服务器的攻击并不是用户自己的问题，因为普通用户对此无能为力。但覆巢之下无完卵，服务器遇到这类攻击时，普通用户也将沦为受害者。

为阻止对服务器的攻击，服务器的编程和配置必须非常仔细，不论客户端发来的请求构造得多么精心，都要确保不泄露未授权信息，不允许未授权访问。但实际上，服务器上运行着的大型复杂程序，使服务器容易出现可能被攻击者利用的编程缺陷和配置错误。**SQL 注入**（SQL injection）就是一种常见的针对服务器的攻击方式。在万维网上，服务器一般在后台运行数据库，访问数据库时通常使用标准接口 SQL（Structured Query Language，结构化查询语言）。如果没有严格限定访问权限，机灵的攻击者就可以提交精心构造的 SQL 查询指令来检查数据库结构，从而提取未授权的信息，甚至有可能在服务器上运行他们自己写的代码，进而控制整个系统。尽管上述攻击及其防御机制是众所周知的，但服务器被攻击的事件仍然不断发生。

系统一旦沦陷，能制约入侵者恶行的限制就很少了。特别是当入侵者设法获得了 root 权限（也就是系统的最高级别管理权限）时，他们就更会为所欲为。不论是对服务器，还是对你家里的个人计算机，都是如此。获得服务器 root 权限后，入侵者就能摧毁网站，或者在网站上发布令人尴尬的言论以丑化所有者的形象。还可能下载破坏性程序，或者在网站上存储并发布色情图片和盗版软件等违法内容。服务器可能会因此丢失大批数据，个人计算机一样在劫难逃。2011 年 4 月，黑客从索尼 PlayStation 网窃取了七千万用户的姓名、家庭住址、电子邮件地址等信息。在几乎同一时期的另一桩案子中，为多家大企业提供电子邮件营销服务的

Epsilon 公司丢失了几百万用户的姓名和电子邮件地址。（两家有业务往来的公司都就 Epsilon 的事故给我发送了提醒，一家说只泄露了电子邮件地址，另一家则说还泄露了姓名。两家说法不一致让我谁的说法也没法相信了。）类似这种事故，虽然规模未必总是这么大，却经常发生。

服务器还经常遭遇**拒绝服务攻击**（DoS，Denial of Service）。在这种攻击中，发起者把大量网络流量引导到一个网站，利用密集的访问使其停止响应。拒绝服务攻击是精心策划的，通常用僵尸网络来完成：沦陷的肉鸡接到命令后，会在指定时间访问指定网站，从而导致流量骤增。从多个来源同时发起的拒绝服务攻击通常称为**分布式拒绝服务攻击**（DDoS，Distributed Denial of Service）。

不幸的是，防御者需要防住所有可能的攻击，而攻击者只要找到木桶的一块短板就行。因此，这场攻防战的双方是不对等的，天平倾向于攻击一方。

对传输中信息的攻击

尽管**对传输中信息的攻击**仍然很严重也很常见，但这个问题在万维网安全中往往是放到最后考虑的。无线网络的普及，或许会提高人们对这个问题的重视程度。例如，坏蛋可能会偷听你和银行的对话，窃走你的帐号、口令等信息。但如果你和银行之间的通信是加过密的，偷听者就很难分析出他听到的数据是什么意思了。HTTPS 协议是加强型的 HTTP，对请求和应答双向数据都做了加密。这样，偷听者通常就不能读取信息，也无法伪装成通信中的任何一方。然而，由于 HTTPS 并没有被广泛使用，攻击者仍然可以用 Firesheep 之类的程序在咖啡馆、机场和提供无线上网服务的其他公共场所偷听连接，从而可以让他们假扮成你，而你根本

无法察觉到。靠窃听商店里未加密的终端机通信就可以获得信息卡信息：窃贼坐在商店外的车里，一有交易发生，窃听设备就可以捕获信用卡数据。

还有一种攻击形式是“中间人攻击”，即攻击者截获消息，并按自身需要加以修改后发出去，该消息看起来是直接来自源头的。（第 8 章讲到的《基督山伯爵》中的一个故事，采用的就是这种攻击手法。）

VPN（Virtual Private Network，虚拟专用网）在两台计算机间建立起加密的通道，以保证其间双向通信的安全。企业通常通过部署 VPN 来让员工在家里办公，或者到国外出差时连回公司。用户登录 VPN 不仅需要输入口令，还需要持有一个物理设备。这种“双因子”认证比只输入口令要安全，因为它不仅需要用户知道某些信息（口令），还需要拥有某些实物（设备）。在实际应用中，这个设备可以是手机里运行的程序，能按某种算法生成动态数码以便跟服务器上用同样算法生成的数码匹配，也可以是一个能显示数码的专用设备，在输口令时把设备显示的数码一起输进去。双因子认证技术不仅用于 VPN，也可用于银行和其他在线账号服务。

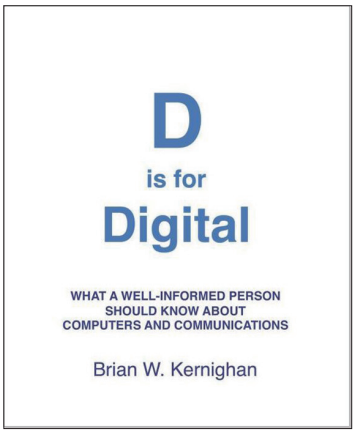
颇具讽刺意味的是，制造出业界广泛使用的双因子认证设备 SecureID 的 RSA 公司在 2011 年 3 月遭遇入侵，安全信息失窃，从而使某些 SecureID 设备不再安全。随后，军火商洛克希德·马丁公司受到攻击，看样子黑客充分利用了 SecureID 失窃案中得到的信息。

自我防御

个人计算机用户如何进行自我防御？有人向我征求这个意见时，我会用本节的内容告诉他。我把计算机用户的防御手段分成三类：第一类非常重要，第二类中等重要，第三类则要看你的偏执程度。（如你所料，第三类我就不细讲了，因为大多数人根本不会偏执到这种程度。）

非常重要

- 选择安全的口令，让别人不会轻易猜中，即使反复尝试也不至于马上就被破解。一串随机的大小写字母、数字、特殊字符混搭而成的口令，显然要比词典里的单词、生日、家人和宠物名来得安全。至于信奉“最危险的才是最安全的”，从而用“password”作为口令的搞笑念头，还是及早打住为妙。
- 不要在重要网站和无关紧要的网站用同一口令，前者包括银行网站和电子邮件，后者譬如在线报纸。工作账号和私人账号不要用同一口令。经常修改口令，但不要改得太有规律，比如在最后一位数加 1。
- 不要用开放的无线网络做重要的事。保证用 HTTPS 连接无线网络，但别忘了 HTTPS 只加密内容。
- 收到来历不明的邮件不要打开附件。收到朋友寄来的邮件，如果发现有意想不到的附件，也不要打开。不要因为软件提示你接受、点击或安装就全部照做。不要从可疑来源下载软件，除了可信来源之外，下载和安装软件都要保持警惕。
- 使用 Windows 要安装防毒软件并保持更新，不要去点那些声称要对你的计算机进行安全检查的网站链接。关闭微软 Office 程序的宏，尽量禁用 ActiveX 控件。如果用的是苹果电脑，现



《世界是数字的》没有高深莫测的专业术语，但它全面解释了当今计算和通信领域的工作方式，包括硬件、软件、互联网、通信和数据安全，而且讨论了新技术带来的社会、政治和法律问题，让你理解这些问题以及在解决问题时作出的折中。这本书结构紧凑、深入浅出，即便没有技术背景也能由此了解这个数字世界的全貌和真相。本文节选自《世界是数字的》。

在还不太需要防病毒软件，但同样要小心。其他软件，比如浏览器和操作系统本身，既然厂家经常发布安全补丁，就可经常更新。

中等重要

- 关闭浏览器弹窗和第三方 cookie（讨厌的是，每个浏览器都分别保存自己的 cookie，所以每个浏览器要分别设置一遍）。用 Adblock 或 Flashblock 等软件拒绝广告图片。
- 用垃圾邮件过滤器筛选电子邮件。
- 关闭 Adobe Reader 的 JavaScript。
- 关掉不用的服务。比如我的苹果电脑可以共享打印机、文件和设备，还能从别的计算机远程登录进来管理我的计算机。Windows 也有类似的一套服务。我只保留了打印机共享，其他统统关掉了。
- 打开计算机上的防火墙程序，监控进出的网络连接，禁止违反访问规则的连接。
- 只要有可能，就对重要账户使用双因子认证。

偏执狂专用

- 禁用邮件阅读程序读取 HTML 和 JavaScript。
- 用 NoScript 限制 JavaScript，用 Ghostery 限制跟踪。
- 关掉所有 cookie，只对自己明确想启用的网站打开。
- 选用那些不易成为攻击目标的系统，比如用 Linux 或 Mac OS X 系统而不是 Windows。使用 Chrome、Firefox、Safari 和 Opera 等浏览器，而不使用 Internet Explorer。
- 鉴于手机也越来越多地成为攻击目标，在手机上采用类似的预防措施也是很必要的。

假如你不知道病毒、网络钓鱼以及类似的风险是怎么回事，那你受害的机率一定会大大增加；假如你不知道社交网络怎么泄露甚至任意传播你认为是个人隐私的信息，那你无意间泄露的很可能比自己想象的还要多；假如你对商业利益集团不顾一切从你的个人信息中挖掘线索这件事毫不知情，那你就会为了蝇头小利而出卖自己的隐私；假如你不知道在咖啡店和飞机上使用个人银行服务是有风险的，那么你的钱和身份就会让网络窃贼有可乘之机。

至于我的写作目标，是希望读者能对计算机和通信技术有一个深入的了解，真正明白它们的工作原理，它们的起源，还有未来的发展趋势。然后，能够从对自己有益的角度重新看待这个世界。果能如此，吾愿足矣。■

几种针对无线网络的攻击和对策

作者 / Kevin Beaver

资深信息安全专家，专业讲师，已在信息安全领域工作 20 余年，为《财富》1000 强企业、安全产品供应商、独立软件开发商、大学、政府机构、非营 利组织和企业进行信息安全评估。曾受邀在 CNN 电视节目和业内专业会议上普及他的安全研究成果，并发表了数百篇技术论文。Kevin 毕业于佐治亚理工学院，是国际注册信息系统安

很多恶意攻击（包括拒绝服务攻击）都能针对无线局域网展开。这包括在脱离网络和重新接入的过程中强迫接入点泄露它们的 SSID。除此之外，黑客还可以阻塞接入点的射频信号，特别是 802.11b 和 802.11g 系统的信号，从而迫使无线客户端重新连接到伪装成受害接入点的流氓接入点。

黑客们可以通过恶意使用 ESSID-jack 和 monkey-jack 这样的工具制造中间人攻击，并可以通过使用 Gspoof 和 LANforge 这样的数据包生成工具以每秒数以千计的数据包淹没整个无线网络，这足以使网络瘫痪了。这种拒绝服务攻击在无线局域网中是很难防范的，比在有线网络中防范起来还要难很多。

我们可以对自己的无线局域网进行多种攻击。与之相关的对策有助于保护自己的网络不受这些漏洞以及上述恶意攻击的影响。在测试无线局域网的安全性时，要注意以下缺陷：

- 未加密的无线流量；
- 脆弱的 WEP 和 WPA 预共享密钥；
- 未经授权的接入点；
- 容易被绕过的 MAC 地址控制；
- 能够亲自接触到的无线设备；
- 默认的配置设置。

全专家（CISSP），读者可通过 http://securityonwheels.com 了解更多关于他的信息。

译者 / 傅尔也

尝试以外人身份连接到无线局域网并运行 LANguard 这样的漏洞评估工具是个不错的开始。该测试使我们可以看到别人能看到些什么，包括操作系统版本、接入点上开放的端口，甚至是无线客户端的网络共享等信息。图 1 展示了可能被泄露的与网络接入点有关的信息类型。

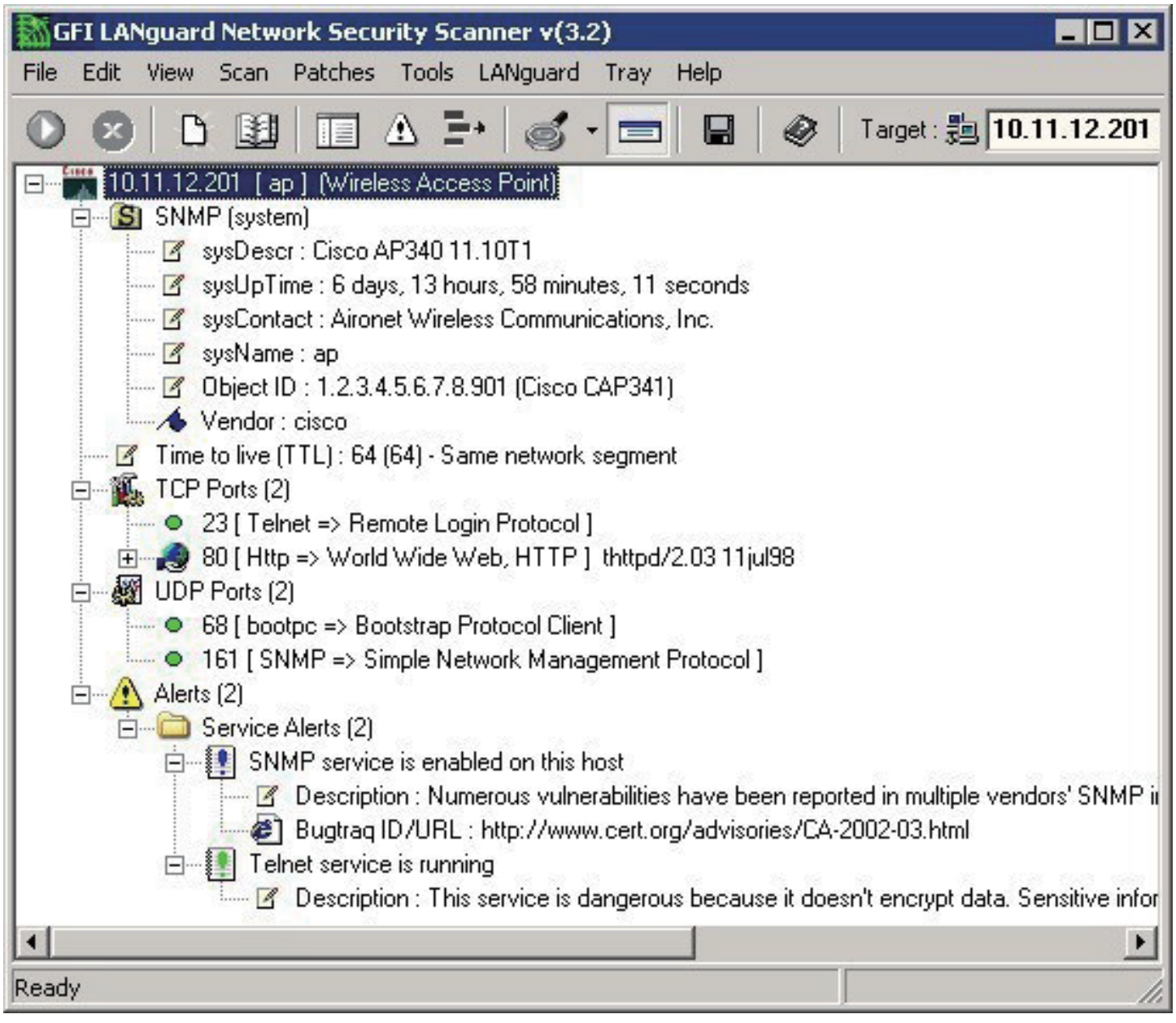


图 1　对可能很脆弱的接入点进行的 LANguard 扫描

别忽略蓝牙

组织中无疑会有人在使用一些启用了蓝牙无线连接的设备，比如笔记本和智能手机等。虽然漏洞并不像在基于 802.11 的 Wi-Fi 网络中那么普遍，不过蓝牙也存在漏洞，目前 http://nvd.nist.gov 上就列出了逾 60 种与蓝牙相关的漏洞，而且有不少黑客工具能利用这些漏洞。通过组装使用蓝牙狙击枪（BlueSniper Rifle，长得很像狙击步枪的一种蓝牙攻击设备），我们甚至能够克服蓝牙信号的个人区域网络距离限制（通常只有几米），从而对蓝牙设备进行远程攻击。下面列出了一些用于测试蓝牙认证 / 配对和数据传输弱点的资源和工具。

- Car Whisperer（http://trifinite.org/trifinite_stuff_carwhisperer.html）。
- Blooover（http://trifinite.org/trifinite_stuff_blooover.html）。
- BlueScanner（https://labs.arubanetworks.com）。
- Bluesnarfer（http://www.alighieri.org/tools/bluesnarfer.tar.gz）。
- 蓝牙狙击枪（http://www.tomsguide.com/us/how-to-bluesniper-pt1,review-408.html）。
- Bluejacking 社区网站（http://www.bluejackq.com）。
- Windows XP 版 BTScanner（http://www.pentest.co.uk/src/btscanner_1_0_0.zip）。
- Smurf（http://www.gatefold.co.uk/smurf）。
- 各种蓝牙攻击的详细展示（http://trifinite.org/Downloads/21c3_Bluetooth_Hacking.pdf）。

移动设备已然成为信息安全的全新困局。老实说，我觉得它们是各企业面临的一大风险。移动设备不仅会通过蓝牙受到攻击，还具有严重的物理安全缺陷，这可能让恶人从组织中获得大量敏感信息。可以参考 NIST 的第 800-48 号特别出版物，看看该如何锁定自己的蓝牙系统。该出版物的网址是 http://csrc.nist.gov/publications/nistpubs/800-48/NIST_SP_800-48.pdf。

加密流量

无线通信流量是可以直接从无线电波中捕获的，使得这种通信介质很容易被窃听。除非流量经过加密，否则都是像在标准有线网络中那样以明文的形式收发的。最重要的是，802.11 的加密协议、有线等效保密（Wired Equivalent Privacy，WEP）和 Wi-Fi 保护访问（Wi-Fi Protected Access，WPA）自身也存在弱点，使攻击者可以破解加密密钥并解密捕获的流量。可以这么说，这些漏洞真是把无线局域网推上了风口浪尖。

WEP 在一定意义上讲也算是名副其实的，它提供了相当于有线网络的保密性。不过，这个结论是在 WEP 不是很容易被破解的前提下才成立的。WEP 使用了相当强的对称（共享密钥）加密算法 RC4。黑客们可以观察加密的无线流量，并恢复 WEP 密钥，因为 RC4 初始化向量在该协议中的实现存在漏洞。缺陷就是该初始化向量只有 24 位的长度，从而使每 1670 万个数据包就会出现重复的初始化向量。根据进出网络的无线客户端数量的不同，很多情况下甚至会更快出现重复。

多数 WEP 实现都会在初始化无线局域网硬件时将初始化向量置为 0，然后每发送一个数据包就将其递增 1。大约每 5 个小时就会使初始化向量重新初始化，即再次从 0 开始计数。因此，客户端数量较少、无线数据包的传输率相对较小的无线局域网，因为没有足够多的无线流量生成，就要比那些需要传输大量无线数据的大型无线局域网更安全。

使用 WEPCrack（http://wepcrack.sourceforge.net）、AirSnort（），或我的最爱——aircrack 套件（http://aircrack-ng.org），黑客们只需要花上几小时，至多几天时间（这取决于网络中无线流量的多少）

收集数据包，就能破解 WEP 密钥了。图 2 展示了 airodump（aircrack 套件的一部分）捕获 WEP 初始化向量的情况，而图 3 展示的是 aircrack 正在破解我的测试网络的 WEP 密钥。

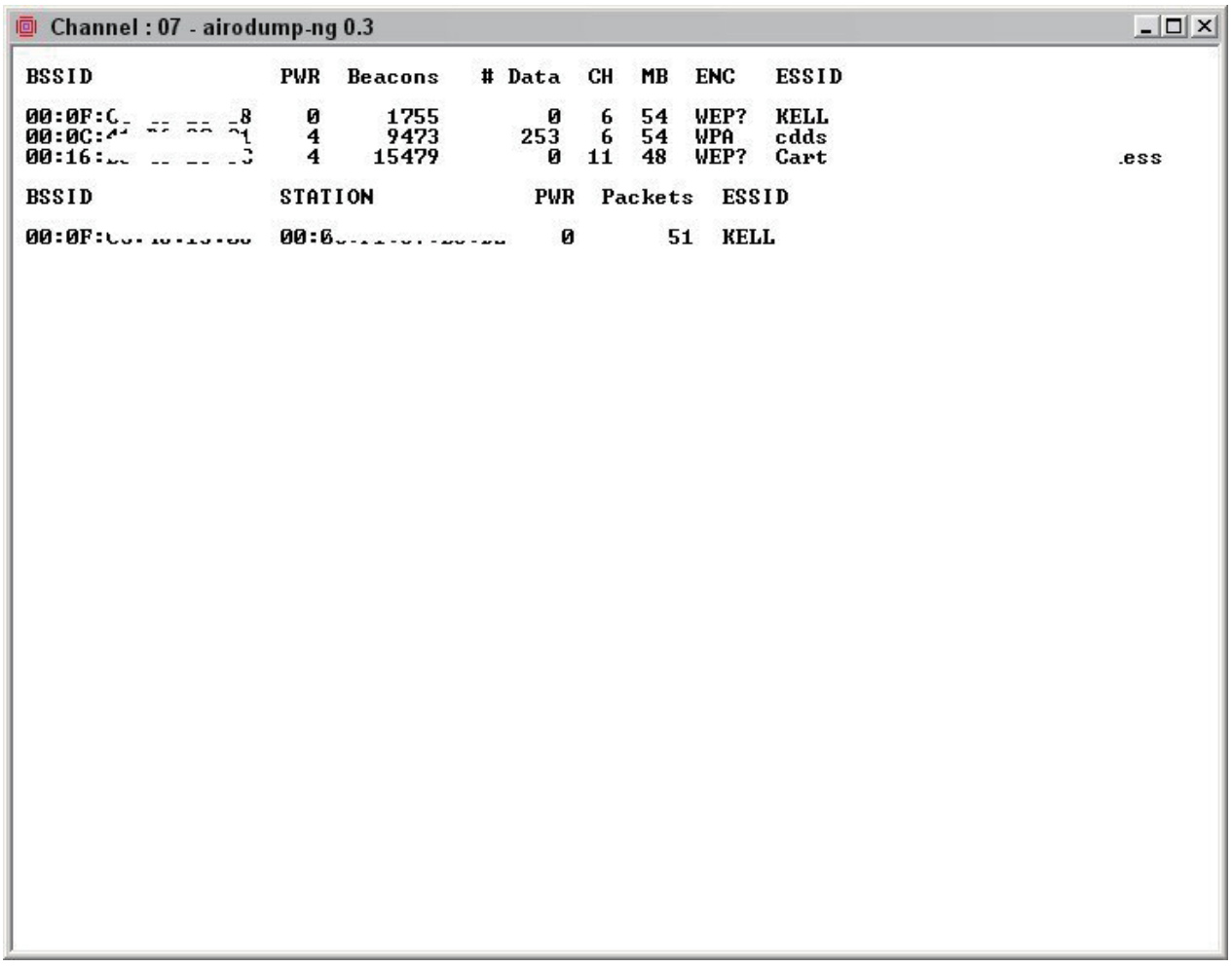


图 2　使用 airodump 捕获 WEP 初始化向量

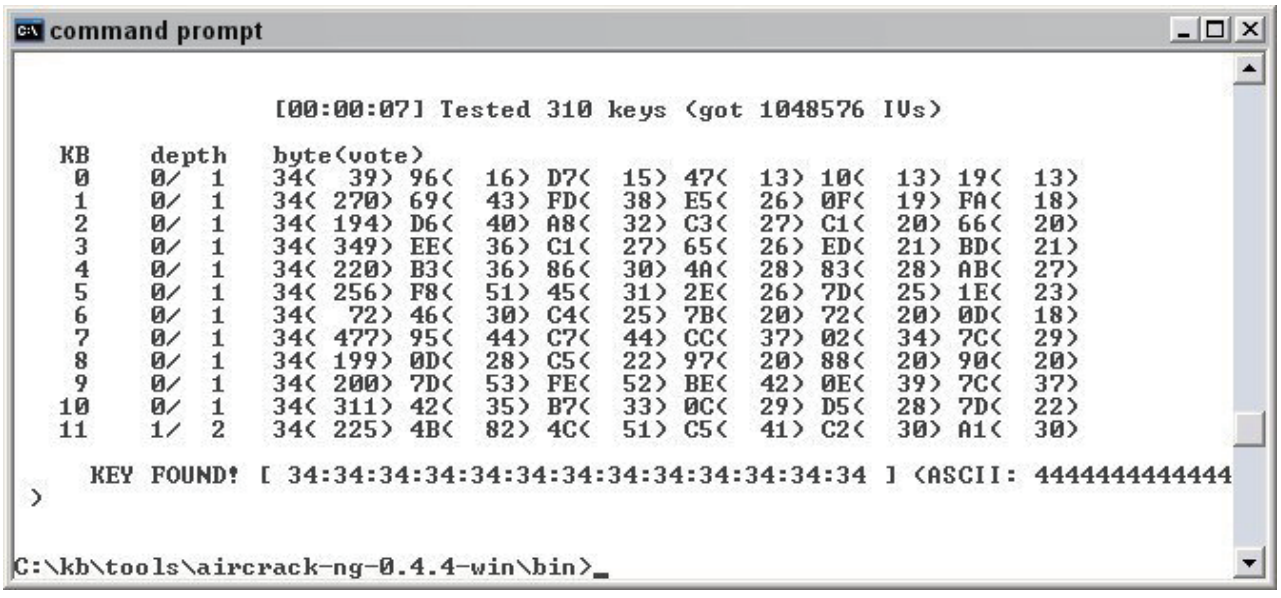


图 3　使用 aircrack 破解 WEP

我不是 Mac 用户，不过我听说过基于 MAC OS 的 KisMAC（http://trac.kismac-ng.org）也是种不错的 WEP 密钥破解工具。

在 Windows 上运行 airodump 和 aircrack 是件十分简单的事。只要根据先前提供的项目网址下载并提取 aircrack 程序集、cygwin Linux 模拟环境，以及支持 PEEK 文件，就可以开始捕获数据包并着手破解工作了！

更长的密钥长度（比如 128 位或 192 位）也不会让破解 WEP 变得难于登天。这是因为 WEP 的静态密钥调度算法，使得密钥长度每增加 1 位，破解密钥所要捕获的数据包只需要增加 20 000 个左右。

无线行业已经提出了一个解决 WEP 问题的方案，该方案被称为 Wi-Fi 保护访问（WPA）。WPA 使用了临时密钥完整性协议（Temporal Key Integrity Protocol，TKIP）加密系统，该

系统修复了所有已知的 WEP 问题。替代 WPA 的 WPA2 使用了更为强大的加密方法——基于高级加密标准（Advanced Encryption Standard，AES）的计数器模式及密码区块链信息认证码协议（Counter Mode with Cipher Block Chaining Message Authentication Code Protocol，或简称 CCMP）。运行在“企业模式”下的 WPA 和 WPA2 要求有 802.1x 认证服务器（比如 RADIUS 服务器）为无线局域网管理用户账户。与自己的供应商联系下 WPA 更新的事宜吧。

我们也可以使用 aircrack 破解 WPA 和 WPA2 的预共享密钥（PSK）。要破解 WPA-PSK 加密，就必须等待无线客户端认证到其接入点的时机。有一种快速（而且很坏）的强制重新认证的方法，那就是向广播地址发送解除验证的数据包。我的合著者彼得 •T• 戴维斯和我在我们的另一本著作 Hacking Wireless Networks for Dummies 中详细介绍了这些内容。

我们可以使用 airodump 捕获数据包，然后启动 aircrack，也可以同时运行二者，输入以下命令开始破解预共享密钥。

```
#aircrack-ng -a2 -w path_to_wordlist <capture file(s)>
```

另一种可用来破解 WPA 和 WPA2 密钥而且相对较新的工具是商业软件 Elcomsoft Wireless Security Auditor（EWSA）。要使用 EWSA，只要以 tcpdump 格式（每一种无线局域网分析器都支持该格式）捕获无线数据包，将捕获文件载入该程序，然后没多久就能得到预共享密钥了。EWSA 是略显独特的，因为它能在比通常所花时间更短的时间内破解 WPA 和 WPA2 预共享密钥，不过前提是必须要有支持 NVIDIA 或 ATI 显卡的计算机。没错，EWSA 不仅能使用 CPU 的处理能力，它还可以利用显卡中图形处理器（GPU）的处理能力和强大加速能力。这就是创新之处！

EWSA 的主界面如图 4 所示。

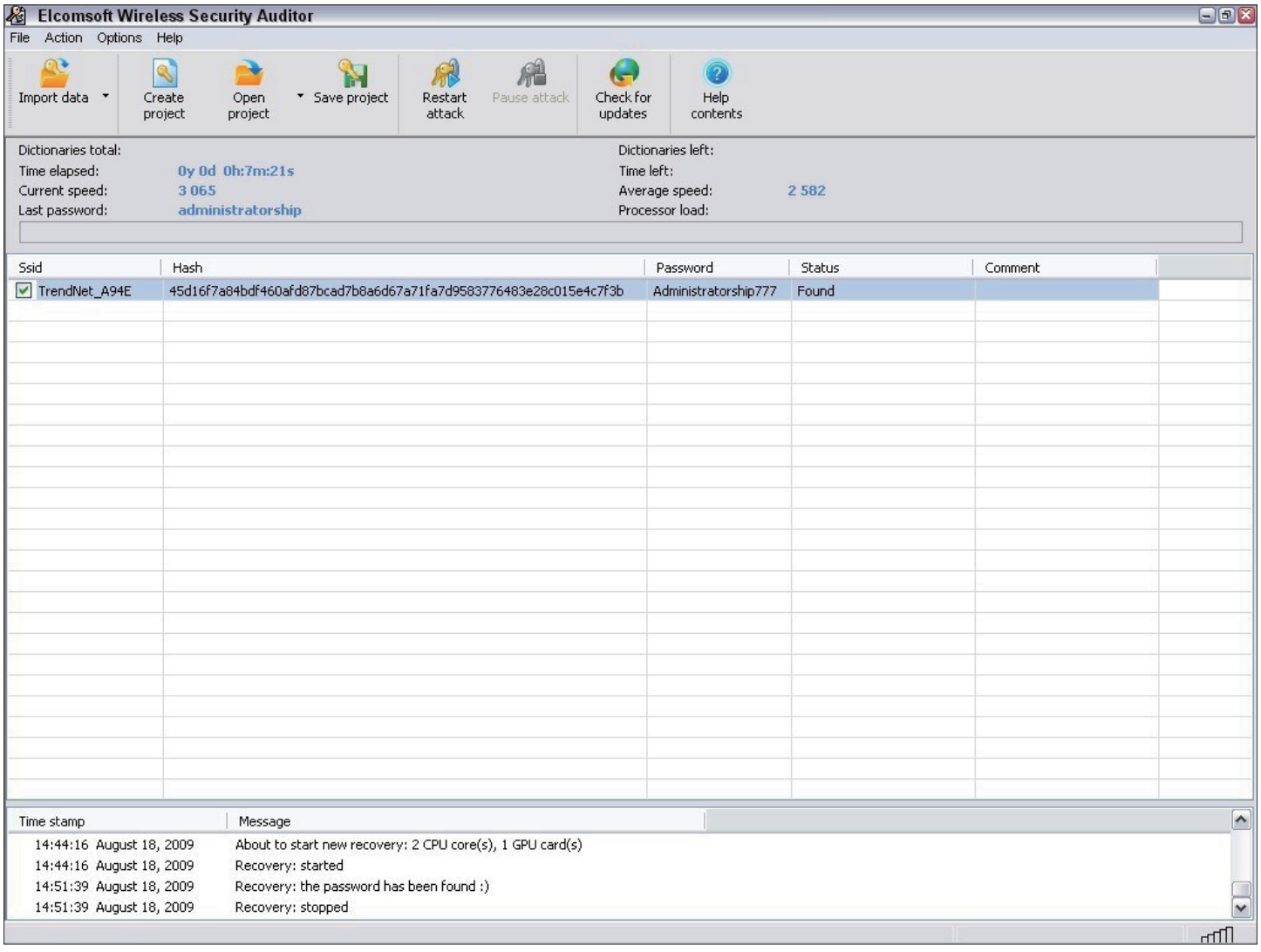


图 4　使用 Elcomsoft Wireless Security Auditor 破解 WPA 预共享密钥

有了 EWSA，我们能以每秒高达 50 000 的速率破解 WPA/WPA2 预共享密钥。将其与只使用 CPU 时每秒几百个密钥的速度相比，就能看到这类工具的价值。我就说了，这钱花得值。

在进行安全评估的过程中需要使用无线局域网分析器查看流量时，如果网络启用了 WEP，除非知道与网络关联的 WEP 密钥，否则是没法看到任何流量的。大家可以将密钥输入到分析器中，不过要记住的是，如果黑客们使用我之前提到的工具破解了我们的 WEP 密钥，那么我们的无线局域网对他们来说也是一览无余的。

图 5 展示了在开始捕获数据包之前，往 OmniPeek 中输入 WPA 密钥之后，如何通过该工具的 Capture Options（捕获选项）窗口查看无线局域网中的各种协议。

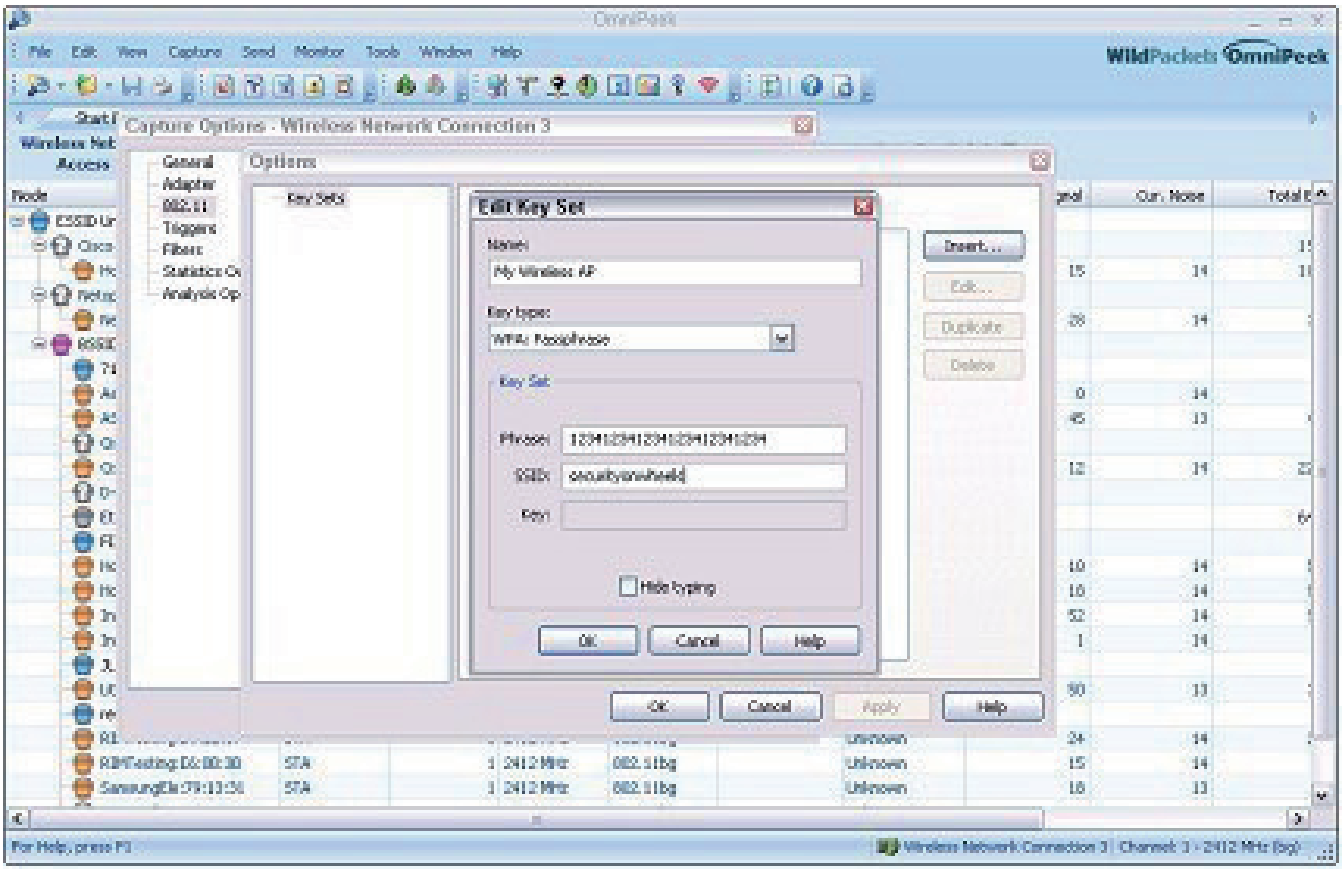


图 5　使用 OmniPeek 查看加密的无线流量

防御加密流量攻击的对策

解决 WEP 问题最简单的办法就是为所有的无线通信使用 WPA，或者更理想的情况是使用 WPA2。我们也可以通过为客户端通信启用点对点隧道协议（Point-to-Point Tunneling Protocol，PPTP），在 Windows 环境中使用 VPN 来解决这一问题（免费的）。还可以使用 Windows 内置的 IPSec 支持、Secure Shell（SSH）、安全套接层 / 传输层安全（Secure Sockets Layer/Transport Layer Security，SSL/TLS），以及其他专有厂商的解决方案，来保证网络流量的安全。要记住，针对 PPTP、IPSec 和其他 VPN 协议的破解程序也是存在的，不过总的来说，这样已经相当安全了。

还存在更新的基于 802.11 的解决方案。如果将无线主机配置成在发送一定量的数据包后动态生成新的密钥，WEP 的漏洞就不会被利用了。很多接入点厂商已经将这种修补程序设置成一个单独的配置选项了，所以请查看一下厂商有没有提供具备密码轮换管理功能的最新版固件。例如思科专有的 LEAP 协议使用的是每个用户都不同的 WEP 密钥，为每个使用思科硬件的用户提供了保护层。还有就是，要注意 LEAP 也不是无懈可击的，也存在针对它的破解程序，比如 asleap（http://asleap.sourceforge.net）。

IEEE（也称为 WPA2）的 802.11i 标准整合了 WPA 的修复和更多内容。该标准是对 WPA 的改进，不过因为它使用了高级加密标准（AES）进行加密，所以与较旧的 802.11b 硬件并不兼容。

如果在使用带有预共享密钥的 WPA（这对小型无线局域网来说绰绰有余了），请确保密钥至少包含 20 个随机字符，这样它才不容易被诸如 aircrack 和 Elcomsoft Wireless Security Auditor 等工具中的离线字典攻击功能所影响。

要记住，虽然 WEP 和弱 WPA 预共享密钥是可以被破解的，但仍比完全不加密要好得多。家庭安保系统标志对那些潜在的非法入室者具有震慑效果，同样，使用了 WEP 或弱 WPA 预共享密钥的无线局域网，对黑客而言，就不像那些未采取这类措施的无线局域网那般具有吸引力。黑客们往往会转而攻击较易的目标，除非他们真的想在一棵树上吊死。

流氓无线设备

要注意那些未经授权连接到网络，并运行在点对点（ad-hoc）模式下的接入点和无线客户端。

要告诉自己的用户，在他们离开办公室时也要注意安全使用 Wi-Fi。告诉他们连接到未知无线局域网的危险，并不断地定期提醒他们。否则，他们的系统可能被黑客入侵，或是受到恶意软件的感染，到时候只要他们再连接回自己公司的网络，那麻烦就找上门来了。

通过使用 NetStumbler 或无线客户端管理软件，大家可以对接入点和不属于自己网络的点对点设备进行测试。还可以使用无线局域网分析器（比如 OmniPeek 和 AirMagnet WiFi Analyzer）的网络监控功能。

流氓接入点一般具有下列特征。

- 奇怪的 SSID，包括常见的默认名称 linksys 和 wifi。
- 奇怪的接入点系统名称——如果硬件支持该功能的话，就是指接入点的名称。别将其与 SSID 弄混了。
- 不属于自己网络的 MAC 地址。看看 MAC 地址的前三个字节（也就是前六位数字），这是表明厂商名称的。对那些不确定其

来源的接入点，可以在 http://standards.ieee.org/regauth/oui/index.shtml 上查找 MAC 地址对应的厂商。

- 弱无线信号，这可能表明接入点在隐藏自身，或是接入点处于组织所在的建筑之外。
- 出现在不是自身网络通信所用无线信道上的通信。
- 无线局域网客户端网络吞吐量的下降。

在图 6 中，NetStumbler 找出了两个可能未经授权的接入点。SSID 分别是 BI 和 LarsWorld 的两个接入点显得很可疑。注意看它们运行在两个不同的信道，有着不同的速率，而且是由两家不同的硬件厂商生产的。如果知道在自己的无线网络上应该有哪些设备在运行（应该知道的，对吧？），那么未经授权的系统很容易就暴露了。

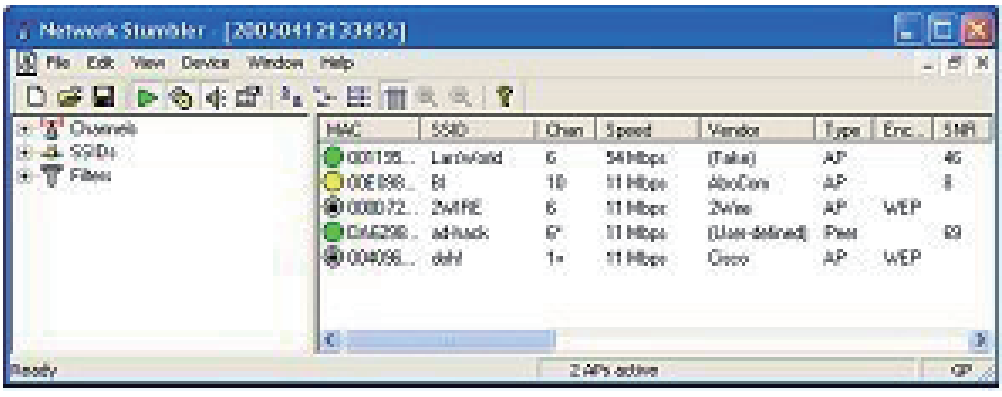


图 6　NetStumbler 显示了可能未经授权的接入点

不过 NetStumbler 确实存在局限性，它没法找到那些禁用了探测响应（SSID 广播）数据包的接入点。而 Linux 和 BSD 系统上常用的无线嗅探器 Kismet，则不仅像 NetStumbler 那样寻找探测响应，而且会查找其他的 802.11 管理数据包，比如关联响应和信标。这使 Kismet 可以检测到那些“隐藏的”无线局域网。

如果 UNIX 平台不是大家的菜，那么还是有种快速卑鄙的方法揪出那些隐藏的接入点，就是使用解除认证的数据包将客户端重新连接到接入点，从而强制广播 SSID。大家可以在我和彼得·戴维斯合著的 Hacking Wireless Networks For Dummies 一书中找到更详细的介绍。

找出隐藏接入点最安全的方式还是使用无线局域网分析器（如 AirMagnet WiFi Analyzer 或 OmniPeek）搜索 802.11 管理数据包。Tamosoft 的 CommView for WiFi（www.tamos.com/products/commwifi）也是种不错的分析器，而且它价格低廉。

通过配置 OmniPeek 启用 802.11 管理数据包捕获过滤器，就可以用 OmniPeek 搜索 802.11 管理数据包，从而找出隐藏的接入点，配置选项如图 7 所示。

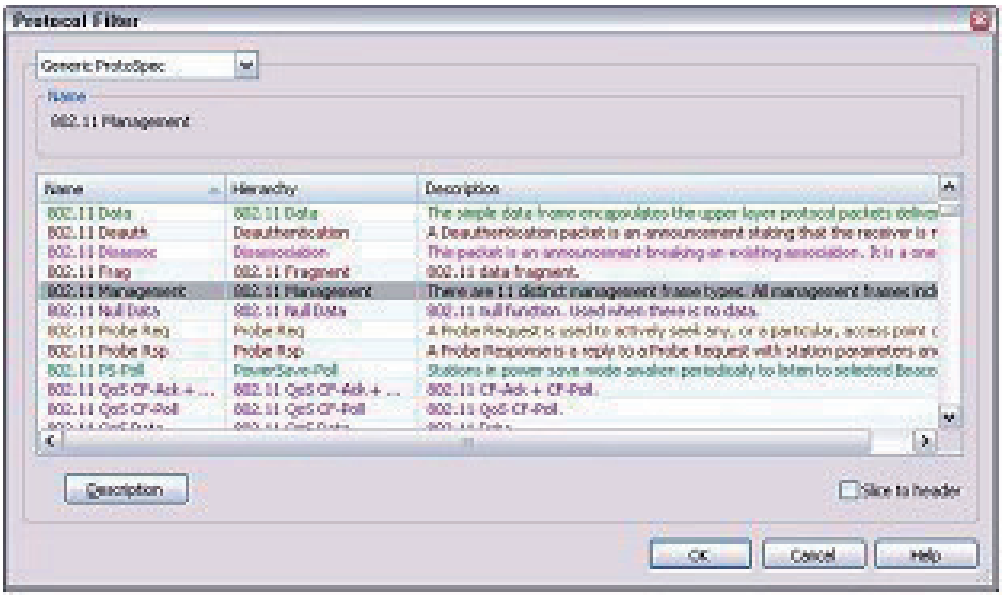


图 7　通过配置，OmniPeek 可以探测那些没有广播 SSID 的接入点

图 8 展示了如何使用 CommView for WiFi 找出奇怪的网络主机。例如，如果知道自己的网络中只使用了思科（Cisco）和 Netopia 的硬件，那么出现 Hon Hai（鸿海）和 Netgear（网件）系统就很不正常。

我在本例中所使用的测试网络可能比大家要处理的网络小，不过这里主要是希望大家大概知道如何让奇怪的系统无所遁形。

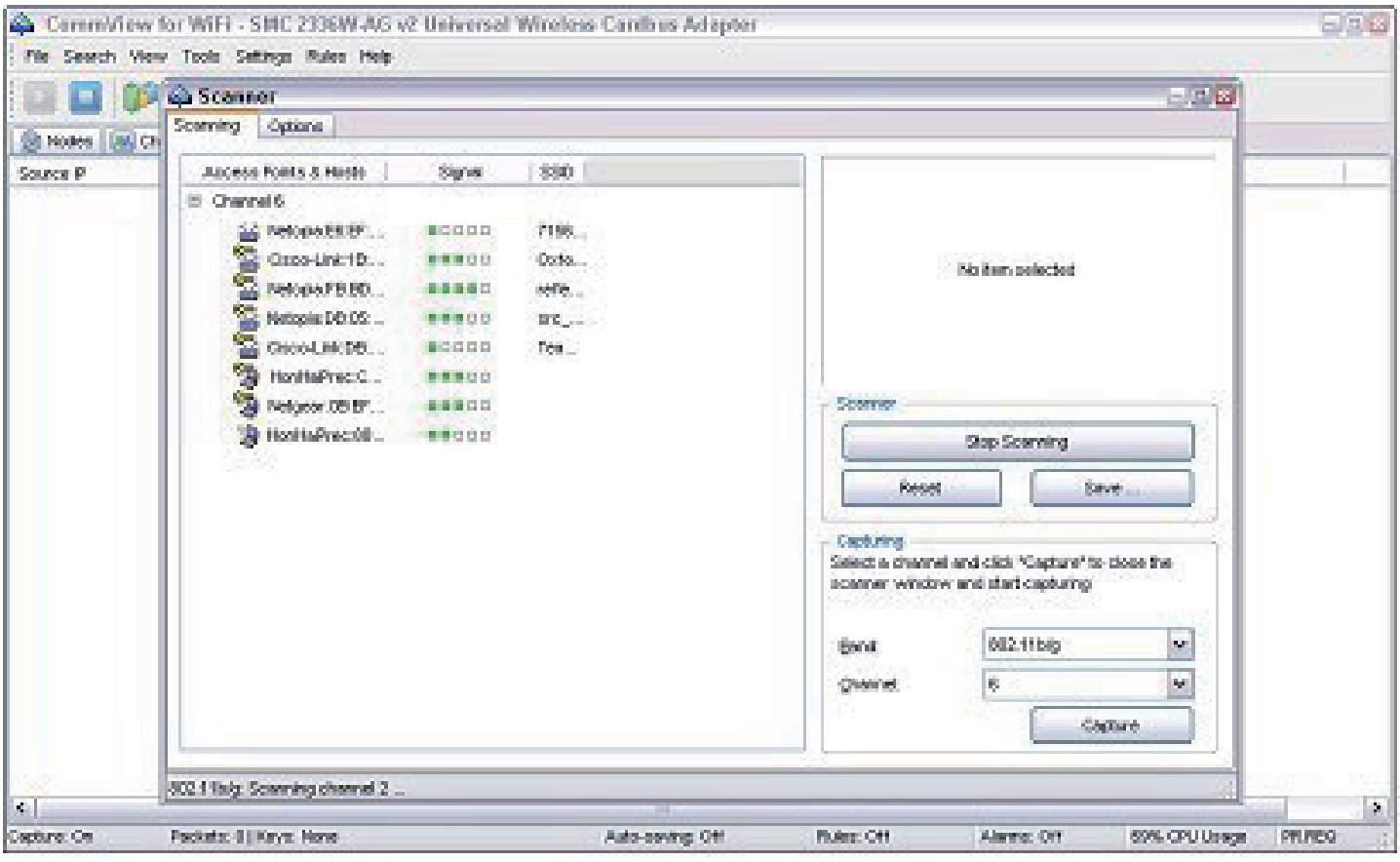


图 8　使用 CommView for WiFi 找出不相干的无线系统

设置在点对点模式下的无线局域网使无线客户端相互之间可以直接进行通信，而不需要经过接入点。这类无线局域网未采取常见的无线安全控制措施，可能造成常见 802.11 漏洞之外的严重安全问题。检测这种流氓网络的好方法之一就是使用 NetStumbler。

随便使用哪种无线局域网分析器都可以找出网络中未经授权的点对点设备。如果遇到了不少的点对点系统，就像图 9 中 AirMagnet WiFi Analyzer 的 STA 部分的截图所列出的那些设备，就很能说明有人在运行未受保护的无线系统，至少是启用了点对点无线连接。不管怎样，这些设备都有可能将我们的网络和信息置于风险之中。

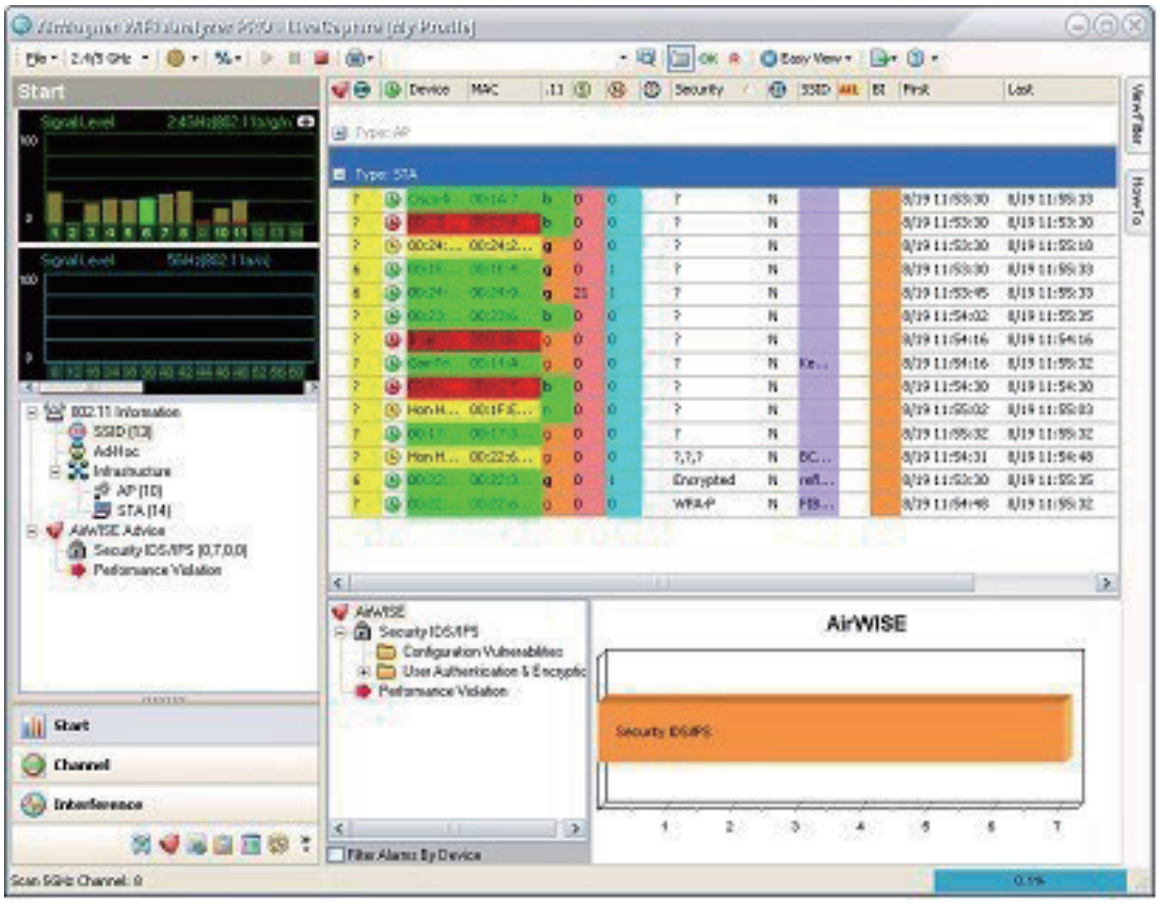


图 9　AirMagnet 显示了一些未经授权的点对点客户端

大家还可以使用我在本章提到的手持式数字热点追踪器（Digital Hotspotter），或者甚至是使用无线入侵防御系统，来搜索 ESS 字段的值不为 1 的信标数据包。

在建筑物或校园内走动（如果愿意的话，可以进行“步行攻击”），执行这项测试，看看自己都能找到些什么。亲自去寻找那些不相干的设备，而且要记住，设置合理的接入点或无线局域网客户端在关闭之后是不会出现在网络分析工具中的。要在建筑中心的附近，或是公共开放区域附近进行搜索。勘察会议和高管办公室附近有没有未经授权的设备。这些区域通常是闲人免入的，但往往会被黑客利用，成为流氓接入点的老窝。

当我们在网络上搜索未经授权的无线设备时，要记住自己可能会接收到附近办公室或家庭中的信号。因此，如果找到了一些蛛丝马迹，也不要立即假设这就是流氓设备在作怪。方法之一是根据检测到的信号强度来判断，办公室外的设备信号应该要比办公室内的弱。AirMagnet WiFi Analyzer 有种很不错的办法来监控用户所发现的无线设备的信号强度。图 10 就是 AirMagnet 的“Geiger counter”（盖革计数器）界面的截图，图中显示了在步行攻击中所找到的接入点的相对信号强度。

以这种方式使用无线局域网分析器有助于缩小范围，并防止在检测到合法的临近无线设备时出现误警的情况。

要确定自己发现的接入点有没有连接到自己的有线网络中，一个好办法就是执行反向 ARP，并将 IP 地址映射到 MAC 地址上。可以在命令提示符中使用 arp –a 命令以达到这一目的，或是直接将 IP 地址与对应的 MAC 地址相比较，看看它们是否匹配。

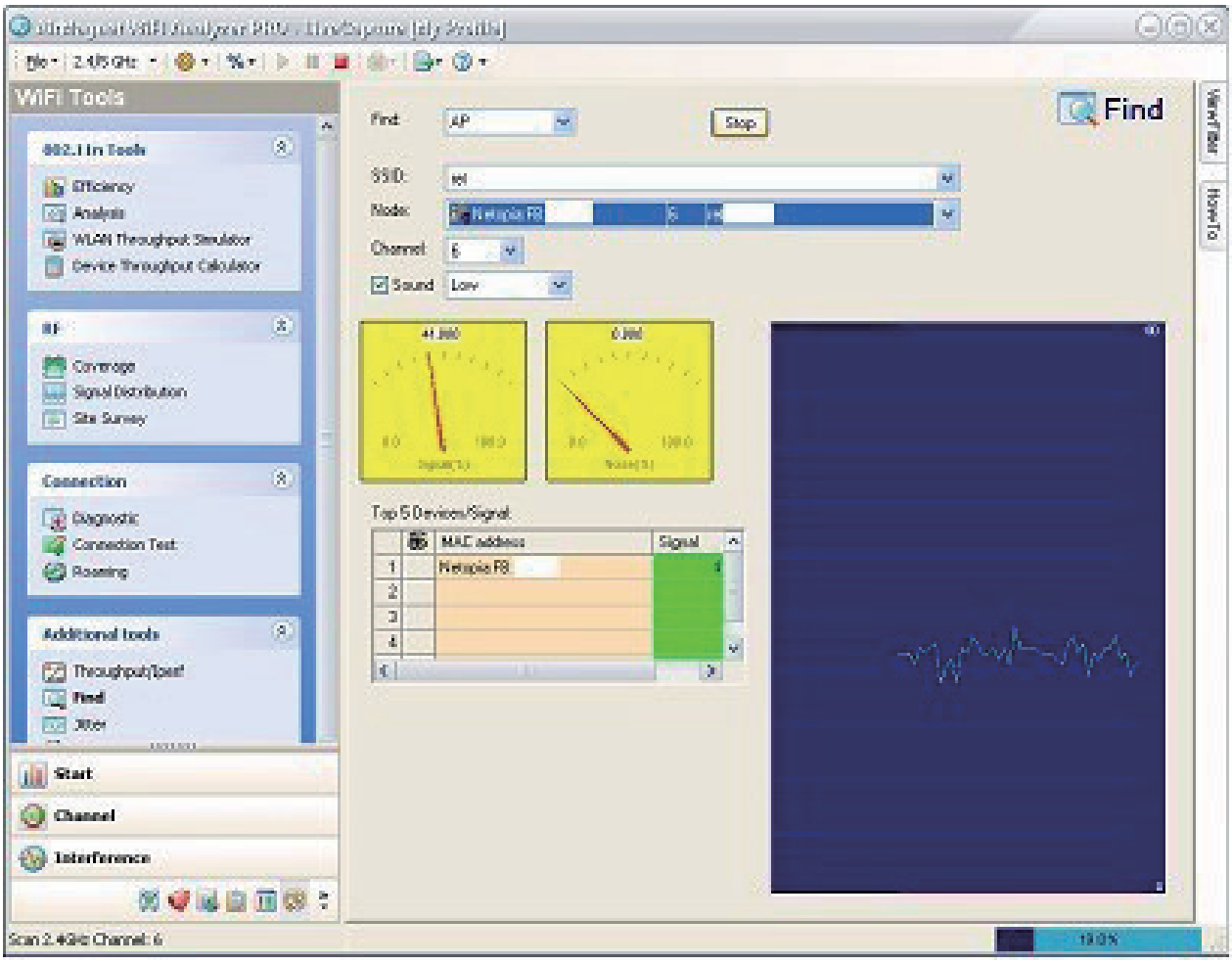


图 10　使用 AirMagnet WiFi Analyzer 监控附近无线系统的信号强度

还有，请记住无线局域网认证的是无线设备，而不是用户。通过使用远程访问软件（比如 telnet 或 SSH），或利用已知的应用程序或操作系统漏洞取得对无线客户端的访问权，黑客们就能利用这一点进入网络。之后，他们就有可能取得网络的完全访问权。

防御流氓无线设备的对策

检测网络中流氓接入点和无线主机的唯一方式就是主动监控无线局域网，比如，每月、每周或使用无线入侵防御系统实时地进行，寻找那些无线客户端或流氓接入点留下的蛛丝马迹。不过就算流氓接入点或客户端没有出现在 NetStumbler 或客户端管理软件中，也不表示我们就高枕无忧了。我们可能还需要无线局域网分析器、无线入侵防御系统或其他网络管理程序的介入。

根据自己使用的接入点，修改一些配置，可能会防止黑客们对我们的接入点执行这些黑客攻击。

> 如果可能的话，将无线信标广播间隔增大到最大设置，这大概是 65 535 毫秒（约 66 秒）。这有助于隐藏自己的接入点，不被那些进行驾驶攻击或在建筑物周围快速移动的黑客们发现。请务必首先进行测试，因为这样做可能会带来其他意想不到的后果，比如使得合法的无线客户端不能连接到网络。
>
> 禁用探测响应，以防止自己的接入点响应这样的请求。

为所有的无线主机使用个人防火墙软件，比如 Windows 系统内置的 Windows 防火墙，防止出现未经授权的远程访问。

最后，别忘了信息安全的不二法门：用户教育。在涉及无线网络的安全使用时，确保安全意识永在心头要比其他措施都更有效。

MAC 欺骗

无线网络中非常常见的一种防御措施就是对 MAC 地址的控制。就是说将接入点配置成只允许具备已知 MAC 地址的无线客户端连接

到网络。这样一来，MAC 地址欺骗也就成了针对无线网络的一种非常常见的黑客攻击。

坏人们在 UNIX 系统中可以轻易地伪造 MAC 地址，只要使用 ifconfig 命令就行了，而在 Windows 中，则可以使用我在第 8 章中提到的 SMAC 实用程序。不过，就像 WEP 和 WPA 那样，基于 MAC 地址的访问控制又是另一层保护，而且总比什么都没有要好。如果有人伪造了 MAC 地址，检测恶意行为的唯一方式，就是找出无线局域网中用在两个或多个地方的同一 MAC 地址。要知道这是件相当棘手的工作。

要确定接入点是否启用了 MAC 地址控制，一种比较简单的方法是，试着关联到该接入点并通过 DHCP 获得 IP 地址。如果可以获得 IP 地址，那么说明该接入点没有启用 MAC 地址控制。

以下内容简述了测试 MAC 地址控制的步骤，并展示了绕开这种控制是多么容易。

（1）找到可供连接的接入点。
（2）使用无线局域网分析器，查找向广播地址发送探测请求数据包的无线客户端，或是以探测响应进行回复的接入点。
（3）将测试机的 MAC 地址改成步骤 2 中所发现无线客户端的 MAC 地址。
（4）确保无线网卡为适当的 SSID 进行过配置。
（5）在网络中获取 IP 地址。
（6）通过 ping 其他主机或浏览互联网，确认自己已经连接到网络中。

要做的就是这些了！完成这六步，就已经绕过无线网络的 MAC 地址控制了。小菜一碟啊！

防御 MAC 欺骗的对策

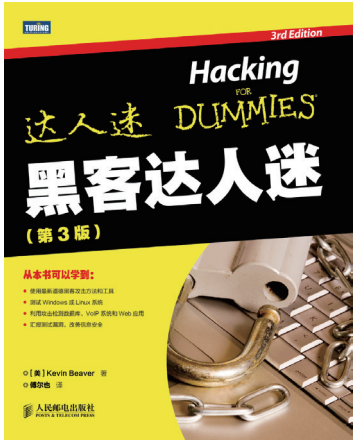
要防止 MAC 地址控制被绕过，避免有人未经授权连接到无线网络最简单的方法就是启用 WPA，或者最好是启用 WPA2。另一种控制 MAC 欺骗的方式就是使用无线入侵防御系统。第二种选择的成本肯定更高，不过考虑到入侵防御系统还可以提供其他主动监控和阻截功能，花这些钱也还是值得的。

昆士兰拒绝服务攻击

相对较新，而且多数人闻所未闻的一次针对 802.11 协议的攻击是 2004 年 5 月由昆士兰科技大学信息安全研究中心（www.kb.cert.org/vuls/id/106678）的研究人员发现的。“昆士兰”攻击也被称为空闲信道检测（Clear Channel Assessment，CCA）攻击，会影响 802.11 中管理无线通信媒介的载波侦听多点检测 / 冲突避免（Carrier Sense Multiple Access/ Collision Avoidance，CSMA/CA）协议中的直接序列扩频（Direct Sequence Spread Spectrum）功能。

无线系统（客户端和接入点等）使用 CSMA/CA 确定无线媒介是否已准备就绪，以及系统能否传输数据。昆士兰攻击利用了 CSMA/CA 中的空闲信道检测（Clear Channel Assessment，CCA）功能，伪造出无线电繁忙的状态，有效地阻止其他无线系统传输数据。实现这一切只是通过将无线网卡置于连续发射模式而已。

有了合适的工具，要执行这种攻击就相对简单了。它可以肆虐整个无线网络，致使其陷入瘫痪。基本上没什么好对策，尤其是在攻击者的信号要比无线系统的信号更强的时候。



摘下白帽吧，快来看看我们的系统是否不堪一击！我们都是好孩子，为什么要学习如何进行黑客攻击呢？那是因为：只有知己知彼，方能百战不殆！保证系统安全的唯一方法就是看看那些坏家伙如何进行黑客攻击，从他们的视角来查看我们的系统是否安全。《黑客达人迷（第3版）》详细介绍了测试方法、一般攻击、搜查目标以及保障敏感商业信息安全的工具。本文选自《黑客达人迷（第3版）》。

进行这种攻击所要做的就是找到一块旧款的 D-Link DWL-650 无线网卡（通过 eBay 或是其他途径），再加上名为 Prism Test Utility（PrismTestUtil322.exe）的 Prism 芯片组测试程序。该程序曾经是可以在 Intersil 的网站上公开下载的，不过现在还是可以在互联网上找到该程序的下载地址（试着用 Google 搜索一下我之前提到的文件名）。使用其他定制软件或硬件也很容易发动这种攻击。这里就没必要截图为证了。简单地说，在将无线网卡置于连续发射模式之前，还是有无线信号的。但在利用了这个弱点之后，无线都不见了！

这项测试可能对无线网络的健康造成危害！只有在测试无线入侵防御系统的受控制的环境中才能进行此项测试，而且要注意不要影响到附近其他人的无线网络。

防御拒绝服务攻击的对策

要防御这种以及其他形式的无线拒绝服务攻击，唯一可能奏效的对策就是在 802.11b/g 网络中安装并使用无线入侵防御系统。不然的话，就要使用具有跳频扩频（frequency hopping spread spectrum）或正交频分复用（orthogonal frequency division multiplexing，OFDM）功能的无线技术，比如 802.11a，802.11n，以及从技术上运行速率超过 20Mbps 的 802.11g。■

专题：网络“安全”

Android 手机

数据安全问题如何产生?

Android 手机中存放着许多与用户个人相关的数据，例如：手机号码、通讯录、短信息、聊天记录、电子邮件、网络软件的账号密码等等。这些数据都是用户的隐私，是神圣而不可侵犯的，然而在现实中，这些数据的存储并没有我们想象的那么安全。本文我们主要从编程的角度出发，来看看数据安全问题是怎么产生的。

作者 / 丰生强（网名非虫）

Android 软件安全专家。看雪论坛 Android 安全版版主；安卓巴士开发交流版版主。对 Android 软件与系统安全有狂热的爱好和独到的见解，对 Android 系统的全部源代码进行过深入地研究和分析。逆向分析实战经验丰富。在国内信息安全杂志上发表过多篇有价值的软件安全文章，目前就职于国内某 Android 开发企业，常年混迹于看雪论坛（ID 非虫）。

外部存储安全

数据安全的首个安全问题就是数据的存储，用户的隐私数据处理的不好，就会暴露给系统中所有的软件，这是最不应该却经常发生的事情。Android SDK 中提供了一种最简单的数据存储方式。那就是外部存储，外部存储是所有存储方式中安全隐患最大的，任何软件只需要在 AndroidManifest.xml 中声明如下一行权限，就可以读写外部存储设备。

```
<uses-permission android:name="android.permission.WRITE_
EXTERNAL_STORAGE"/>
```

外部存储的方式是直接使用 File 类在外部存储设备上读写文件，例如在 SD 内存卡上创建一个 config.txt 文件，并往其中写入“Hello World”的代码如下。

```
File configFile = new File("/sdcard/config.txt");
FileOutputStream os;
try {
    os = new FileOutputStream(configFile);
    os.write("Hello World".getBytes());
    os.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

对于上面生成的 config.txt 文件，其它软件只要拥有内存卡读写权限，就可以访问它的内容。正如您所看到的，外部存储的数据是完全暴露的，这就给很多恶意的软件留下了获取其它软件数据的可乘之机。国内有些 IM 软件，较早的版本中聊天记录就是存放在外部 SD 卡上的，而且数据也没有加密，这就是典型的由第三方软件造成的隐私泄露。

面对这个问题，笔者在此建议，对于不涉及用户隐私的数据，可以适当地采用外部存储来保存，但只要涉及到用户隐私的，哪怕是数据经过加密了，也最好不要放到外部存储设备上，因为分析人员要是掌握了软件数据的解密方法，同样可以简单地获取用户隐私。

内部存储安全

其次是内部存储，它是所有软件存放私有数据的地方。Android SDK 中提供了 openFileInput() 与 openFileOutput() 方法来读写程序的私有数据目录。一段常见的使用内部存储保存数据的代码如下。

```
try {
     FileOutputStream fos = openFileOutput("config.txt",
MODE_PRIVATE);
    fos.write("Hello World".getBytes());
    fos.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

openFileOutput() 方法的第 2 个参数指定了文件创建的模式，如果指定为 MODEPRIVATE，表明该文件不能够被其它程序访问，Android 系统又是如何控制上面生成的 config.txt 不能被其它程序访问的呢？下面我们进入 adb shell 中看看 config.txt 文件的权限。如图 1 所示，config.txt 文件属于 app45 用户，并且只能被 app_45 用户组与自身进行读写操作。Android 系统为每个程序分配了一个独立的用户与用户组，因此可以看出，Android 内部存储的访问是通过 Linux 文件访问权限机制控制的。

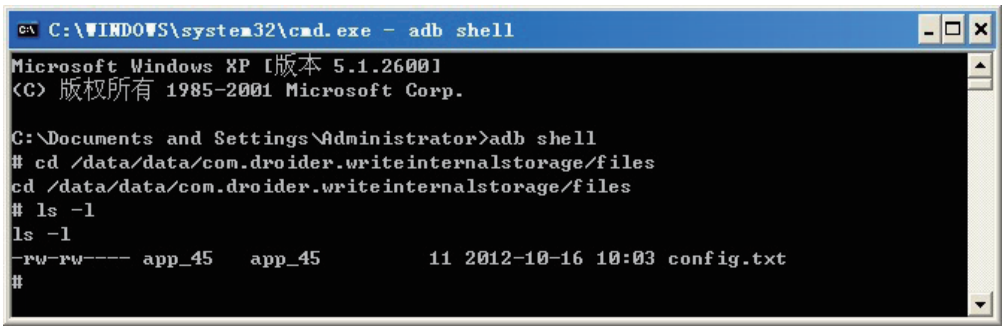


图 1　config.txt 的文件权限

下面尝试将 MODE*PRIVATE* 更改为 MODEWORLD_READABLE，然后在命令提示符下查看其文件权限。如图 2 所示，文件允许其它用户进行读操作。

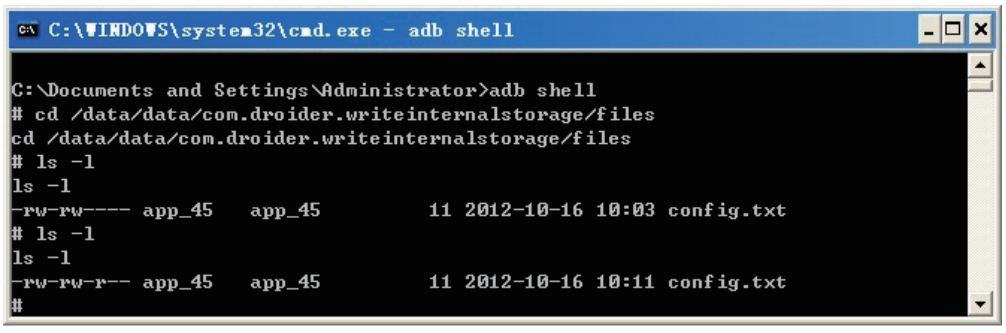


图 2　config.txt 的文件权限

当内部存储文件可以被外部访问时，可以使用以下代码来获取它的内容。

```
try {
     Context context = createPackageContext("com.droider.
writeinternalstorage",
         Context.CONTEXT_IGNORE_SECURITY);
```

```
     FileInputStream fis = context.openFileInput("config.
txt");
    StringBuffer sb=new StringBuffer();
      BufferedReader br = new BufferedReader(new
InputStreamReader(fis));
    String data = null;
    while((data = br.readLine()) != null) {
        sb.append(data);
    }
    fis.close();
      Toast.makeText(MainActivity.this, sb.toString(),
Toast.LENGTH_SHORT).
    show();
} catch (Exception e) {
    e.printStackTrace();
}
```

createPackageContext() 方法允许程序创建其它程序包的上下文（Context 对象），通过这个 Context，可以启动其它程序的 Activity、访问其它程序的私有数据。但前提条件是，其它程序赋予了相应的权限，不会引发安全异常。Context.CONTEXTIGNORESECURITY 指定忽略创建 Context 时的安全异常，始终创建 Context 对象。

从上面可以看出，使用 openFileOutput() 创建文件时，第 2 个模式参数是引发安全隐患的关键。很多读者可能会说：我在使用这个方法时，都是使用 MODE_PRIVATE 模式来创建文件的，你所说的安全隐患对我来说完全不存在。其实，永远不要忽略了，程序可能通过其它途径获取高访问权限，也有可能通过系统漏洞提升进程权限，用户的手机也有可能已经 ROOT，这些情况下恶意程序都能够访问到软件内部存储的数据。

Shared Proferences 与 Sqlite 数据库同样都属于内部存储的范畴，只是在表现形式上不同而已。它们的安全问题与直接使用内部文件存储的安全问题是一样的。

笔者在此提醒广大的软件开发人员，无论采用什么样数据存储方式，存储用户隐私数据时都要进行加密，如果掉以轻心，就有可能造成用户隐私的泄漏。

数据通信安全

数据通信安全是指软件与软件，软件与网络服务器之间进行数据通信时，所引发的安全问题。

首先是软件与软件的通信，Android 系统中的 4 大组件是通信的主要手段，而通信过程中，数据的传递就是依靠 Intent 来完成。Intent 能够传递所有基础类型与支持序列化类型的数据，往 Intent 中添加数据是通过 Intent 类的 `putExtra()` 方法来完成的。例如本小节实例 saveInfo 中有如下代码。

```
Intent intent = new Intent();     / 创建 Intent 对象
intent.setAction("com.droider.saveinfo");   //Action
intent.putExtra("username", "droider"); // 用户名
intent.putExtra("password", "123456");       // 密码
startService(intent);         // 启动服务处理用户名与密码
sendBroadcast(intent);        // 发送广播处理用户名与密码
```

这段代码中通过 Intent 传递了需要保存的用户名与密码，然后分别通过本程序的 Service、Broadcast Receiver 进行保存处理。理想情况下，运行实例程序后，效果如图 3 所示（所有组件只是输出了接收到的用户名与密码信息）。

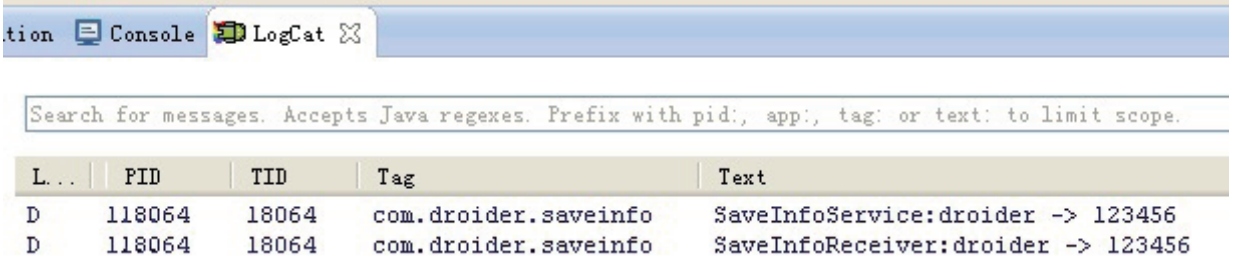


图 3　程序处理了用户名密码保存请求

然而，由于 Intent 中没有明确指定目的组件的名称，导致 Intent 中的数据可能被第三方程序“偷窃”。接下来先运行本小节的 stealInfo 实例，然后再运行 saveInfo 实例，神奇的一幕发生了，如图 4 所示，Intent 的数据被 stealInfo 截获了。

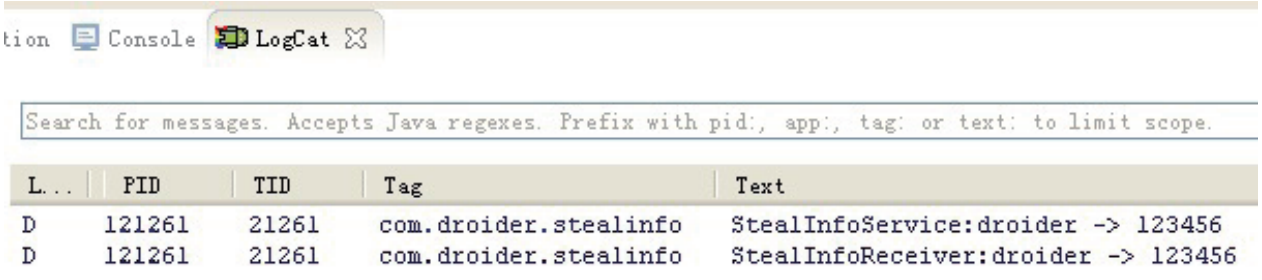


图 4　stealInfo 截获了 Intent 数据

虽然 saveInfo 实例中有处理 Action 为“com.droider.saveinfo”的服务与广播接收者组件，但由于启动组件时没有指定具体的组件名称，而系统中又同时存在多个处理该 Action 的组件，此时 Android 系统就会选择去启动优先级最高的组件，最先启动的程序其组件拥有更高的优先权，这也就是为什么 Intent 会被外部的 stealInfo 响应的原因。

这个安全问题在讲解 Broadcast Receiver 组件安全时曾经讲过，Broadcast Receiver 由于其自身的特殊性，使用 `sendBroadcast()` 传递的 Intent 本身就是暴露的，可以被其它程序获取。因此，它的安全问题是显而易见的，更多的问题可能是响应优先级的争夺。而其它的组件就不同了，传递的 Intent 数据可能不希望被其它程序截获，但如果在编写代码时采用上面的方法来使用 Intent，那势必会造成潜在的安全隐患。

《Android 软件安全与逆向分析》主要从以下几个方面对 Android 安全展开探讨：首先是 Android 程序的反汇编，这一部分是绝大多数开发人员所关心，但大多数人没掌握的内容。 其实是 Android NDK 程序的分析，这一部分是广大安全爱好者希望掌握的，但这方面的资料网上很难找到。 最后是 Android 架构本身的安全性，这一部分内容是所有关注 Android 平台的开发者和用户都应该了解的。本文选自《Android 软件安全与逆向分析》。

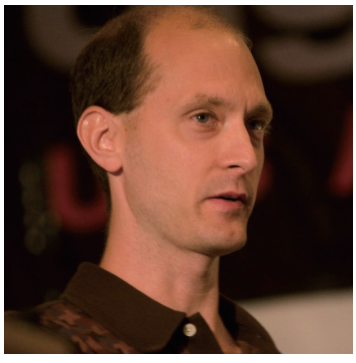
接下来是软件与网络服务器的通信。这样的情况在编写网络软件程序时会经常遇到：软件注册时提交用户注册信息、软件登录验证、聊天消息传递等。同样的，这些信息也是用户的隐私，在进行数据传送时需要谨慎处理。

网络数据通信可能面临的攻击是网络嗅探，如果网络上传送的数据未经过加密，网络嗅探软件截获到的数据中，就有可能包含用户的一些明文隐私数据（例如网银账号与密码），这样产生的后果是难以想象的。因为网络数据没有加密而造成的用户损失事件时有发生，前段时间，看雪安全论坛上就有人爆出，国内某聊天软件在申请账号时，提交的注册信息未经过任何加密，在笔者看来，这不是编程人员能力不足，也不是服务器条件无法达到，而是由于开发人员或者公司本身对安全的不重视造成的。

关于网络数据如何加密，笔者在此也没有太多的建议，网络数据安全不仅是 Android 系统上才有的，从第一款网络嗅探软件诞生的那天起，这个问题就一直存在。经过多年的经验积累，很多公司都有了一套成熟的网络数据传送协议与加密方案，笔者在此提出这个安全问题的目的，一方面是给没有认识到网络数据安全的新手提个醒，另一方面是希望那些目前仍然使用明文传输数据、且有能力解决这个问题的公司，认真做好数据加密工作，不要置用户的隐私安全于不顾。■

专题：网络“安全”

解密 iOS 越狱背后的过程



作者 / Charlie Miller

Charles Miller 现在是 Accuvant 公司的计算机安全咨询师。在此之前，他在美国国家安全局任职 5 年。他在苹果产品上的黑客能力得到广泛认可。2008 年他因为在 MacBook Air 上找到了一个致命漏洞，赢得了加拿大温哥华骇客大会 Pwn2Own 的 1 万美元奖金。次年，他成功黑掉 Safari 赢得了 5000 美元奖金。

译者 / 傅尔也

出于对诸多原因的考虑，用户会为他们的 iOS 设备越狱。有些人是需要一个可供他们开发软件的开放平台，有些则是想要完全控制他们的设备，一些人需要越狱设备从而运行 `ultrasn0w` 这样的软件绕过手机运营商的锁定，还有些人越狱 iPhone 是为了使用盗版应用。

安全研究人员为他们自己的 iOS 设备越狱则有着其他原因。正常情况下 iPhone 受到了严格的限制，不允许执行未签名的代码，这给评估系统安全或在系统中查找安全漏洞的工作带来了很大的障碍。

本文要介绍 `redsn0w`（红雪）越狱工具的内部工作机制。该工具是由 iPhone Dev Team 开发的，大家可以从他们的网站 http://blog.iphone-dev.org/ 下载该工具。这是当前为 A5 处理器出现之前的苹果设备越狱时最常用到的工具，因为它可以支持绝大多数的 iOS 版本，非常易于使用，是目前最稳定的越狱工具，而且在 Windows 和 Mac OS X 操作系统中都可以使用。

有了 `redsn0w`，越狱不过是点击几个按钮，并把 iPhone 设置成 DFU（Device Firmware Upgrade，设备固件升级）模式。这样一来，越狱就算对那些想为 iPhone 越狱的新手用户来说都足够简单了。图 1 展示了 `redsn0w` 的欢迎屏幕。

图 1　redsn0w 的启动画面

虽然在用户眼里看来这是个非常简单的过程，但幕后其实发生了很多事情，而且除了越狱领域的一部分人之外，没人真正了解发生了什么。在大家通读本文后，就能够成为为数不多的、了解 redsn0w 内部工作机制的人了。

因为 iPad2 或 iPhone 4S 这样的 A5 设备不存在已经公开的 bootrom 漏洞，任何针对这些设备的越狱都是用户空间级别的。

不过，这只是意味着前两步要从对 bootrom 进行漏洞攻击并引导 ramdisk，替换成类似先对 MobileSafari 进行漏洞攻击再利用内核漏洞的两步。剩下的越狱过程都是相同的。

对 bootrom 进行漏洞攻击

`redsn0w` 的越狱过程首先会利用 `limera1n`（绿雨）DFU bootrom 漏洞攻击程序，为了达到的最高特权等级执行代码。要攻击的漏洞是 A5 问世之前，在设备中 bootrom USB DFU 栈中的堆缓冲区溢出。我们在这里不会讨论该漏洞的具体信息。如果大家对该漏洞感兴趣的话，可以在 THEiPHONEWiKi 之类的地方找到对该漏洞的描述以及针对它的漏洞攻击程序。

对我们要介绍的内容来说，大家唯一需要了解的就是该漏洞攻击程序会给 bootrom 代码中的签名验证打上补丁，从而引导任意 ramdisk，并为 Low-Level-Bootloader（LLB，底层引导加载程序）、iBoot 和内核打补丁。Chronic Dev Team 在 GitHub 上放出了能执行这些操作的源代码。如果大家想从头开始编写自己的越狱工具，这是个不错的起点，因为 `redsn0w` 的源代码是没有公开的。

引导 ramdisk

redsn0w 利用 `limera1n` 漏洞攻击程序引导修改过的系统，该系统使用了打过补丁的内核和预先定制的 ramdisk。内核被打上了若干个越狱补丁，从而允许执行未签名代码。不过，这里面只包含了一部分在完美越狱的系统中常见的内核补丁。该 ramdisk 是每次执行越狱时按要求生成的，因为根据用户在执行越狱时各项设置的不同，需要在 ramdisk 的根目录中创建不同的文件。然后该 ramdisk 上的越狱可执行文件会检测出现了哪些文件，从

而决定应该激活 redsn0w 的哪些功能。例如，如果出现名为 /noUntetherHacks 的文件，就会跳过完美越狱漏洞攻击程序的安装。

在引导完该 ramdisk 后，内核就会执行其中的 /sbin/launchd 二进制文件，而此文件中包含了初始化越狱工具的小存根。该二进制文件首先会把根文件系统和数据部分挂接到系统中。因为要作出修改，所以挂接的这两项内容都是可读写的。最后，名为 jailbreak 的可执行文件会接管一切并执行接下来的步骤。

为文件系统越狱

默认情况下，iPhone 的文件系统是分为两个部分的。第一部分是根文件系统，其中包含着 iOS 操作系统文件以及 MobileMail 和 MobileSafari 这样的标准应用程序。在 iOS 的较早期版本中，根文件系统的大小约等于这部分所包含文件的大小，基本没剩下多少空闲空间。而现在的根文件系统大小约为 1GB，并且有 200MB 的空闲空间，这些空闲空间应该是不可修改的，因此默认是以只读方式挂接的。而设备存储空间的剩余部分则会分配给第二部分——数据部分，该部分是以可读写方式挂接到 /private/var 目录的。这是由根文件系统中的 /etc/fstab 文件配置的。

```
/dev/disk0s1 / hfs ro 0 1
/dev/disk0s2 /private/var hfs rw,nosuid,nodev 0 2
```

正如大家可以看到的，数据部分的挂接配置含有标志 nodev 和 nosuid。nodev 标志确保（因为文件系统级的攻击）可能出现在可写数据部分的设备节点将被忽略。nosuid 标志则会告诉内核忽略数据部分中可执行文件的 suid 位。而 suid 位标记的可执行文件需要以 root 权限运行，或一般情况下以另一个不同用户（并非

那个执行该文件的用户）的身份运行。因此，这两个标志可以在 iOS 内部设置一道防御权限，提升漏洞攻击程序的小型防线。

这种默认配置对所有的越狱工具来说都是个问题，不管是 bootrom 级的越狱还是用户空间级的越狱，因为这些越狱都需要对根文件系统进行修改，以期在重启后继续存活，或是能添加额外的守护进程和服务。这样一来，每种越狱在取得 root 权限后的第一项活动都是以可读写方式挂接（重新挂接）根文件系统。为了在重启后保持这种改变，下一步是要用下面的内容替换系统的 `/tec/fstab` 文件：

```
/dev/disk0s1 / hfs rw 0 1
/dev/disk0s2 /private/var hfs rw 0 2
```

新的文件系统配置会以可读写方式载入根文件系统，并从第二部分的挂接配置中移除 nodev 和 nosuid 标志。

安装完美越狱漏洞攻击程序

每当新版的 iOS 问世时，之前发现的漏洞就会被封堵。因此，会存在这样一个时期，就是 `redsn0w` 工具可以为旧设备的新固件越狱，但不能安装完美越狱漏洞攻击程序。

一旦出现新的完美越狱漏洞攻击程序，`redsn0w` 的作者就会对其进行修改，使 `redsn0w` 能安装该漏洞攻击程序。因为每套这样的漏洞攻击程序都是不同的，所以它们总需要不同的安装步骤。

不过，虽然实际的完美越狱安装过程是不同的，但这通常不过是重命名或移动根文件系统中的某些文件，然后将一些额外的文件复制到根文件系统中。当大家反编译当前版本的 `redsn0w` 时，可

以看到它能支持为从 4.2.1 到 5.0.1 的大多数 iOS 版本安装完美越狱程序，并能了解到每种完美越狱具体需要哪些文件。

安装 AFC2 服务

AFC（Apple File Connection，苹果文件连接）是每一部 iPhone 上都会运行的文件传输服务，它让用户可以通过 USB 连接访问 iPhone 媒体文件目录 /var/mobile/Media 中的文件。该服务是通过 lockdownd 守护进程提供的，而且名为 com.apple.afc。不过，lockdownd 只是提供对该服务的访问，它的实际实现是在 afcd 守护进程中。要访问该服务，在 Mac 机上可通过 MobileDevice.framework，而在 Windows PC 机上可通过 iTunesMobileDevice.dll 访问。

第二项 lockdownd 服务注册的名称是 com.apple.crashreportcopymobile，它也是由 afcd 实现的，用于将崩溃报告器的报告从设备复制到计算机上，而且为其提供的读写访问权限仅限于读写 /var/mobile/Library/Logs/CrashReporter 目录及其子目录。

因为这两项服务都只是以 mobile 用户的权限运行的，而且都被锁定在特定的目录，所以它们对越狱而言作用有限。因此，redsn0w 和其他一些较早期的越狱工具都会用 lockdownd 注册一个额外的服务 com.apple.afc2。该服务利用 afcd 守护进程提供对整个文件系统 root 权限的读写访问，这是越狱工具一项相当危险的功能，而且很多用户不知道有这个功能。这基本就意味着在将一台没有密码或处于未锁定状态的已越狱 iPhone 连接到 USB 外接电源或他人的计算机上时，就会在没有用户交互的情况下让另一端能读写整个文件系统。而这样另一端的人就能盗窃所连接 iPhone 上的所有数据或植入 rootkit 程序。

通过改变 /System/Library/Lockdown/Services.plist 文件中的 lockdownd 配置，就能安装 com.apple.afc2 服务了。这是个普通的 .plist 文件，因此可以用针对 .plist 文件的标准工具或 API 对它进行修改。在 redsn0w 工具中，是通过向该文件中添加以下代码安装这项新服务的。

```
<key>com.apple.afc2</key>
<dict>
    <key>AllowUnactivatedService</key>
    <true />
    <key>Label</key>
    <string>com.apple.afc2</string>
    <key>ProgramArguments</key>
    <array>
        <string>/usr/libexec/afcd</string>
        <string>--lockdown</string>
        <string>-d</string>
        <string>/</string>
    </array>
</dict>
```

因为只要简单改变配置就能提供文件系统的越狱和新的 AFC2 服务，而且不需要执行未签名的二进制文件，所以它们在重启后都还能起作用，甚至在设备没有完美越狱可用时都是可以的。

安装基本实用工具

苹果公司并没有为 iPhone 提供 UNIX shell，所以无怪乎根文件系统的 /bin 和 /usr/bin 目录中几乎是空的，并未包含大家认为能在这些目录中看到的可执行二进制文件。事实上，5.0.1 版的 iOS 只在这些目录中预装了 5 个可执行文件：

/bin/launchctl；

/usr/bin/awd_ice3；
/usr/bin/DumpBasebandCrash；
/usr/bin/powerlog；
/usr/bin/simulatecrash。

正因如此，像 redsn0w 这样的越狱工具通常会在这些目录中安装一些实现基本功能的实用工具，这些基本实用工具能让越狱文件的安装变得更简单。下面列出的工具是从 redsn0w ramdisk 上的越狱二进制文件中提取出来的，这些就是 redsn0w 要安装的基本实用工具。而越狱二进制文件中也会用到这些工具，比方说用来解压 tar 存档，或是修改 .plist 文件的内容。

```
/bin/mv
/bin/cp
/bin/tar
/bin/gzip
/bin/gunzip
/usr/sbin/nvram
/usr/bin/codesign_allocate
/usr/bin/ldid
/usr/bin/plutil
```

除了这些文件外，还安装了一些其他的库和文件，而它们只在越狱时有用，不是提供给 UNIX shell 的使用者的。因此，我们在这里没有列出它们。有意思的是，官方版的 iOS 固件现在也带有由 redsn0w 重写的 /usr/sbin/nvram 二进制文件。

应用转存

当应用是从苹果的 App Store 安装时，它们是直接安装到 /var/mobile/Applications 目录中的，而该目录位于 iPhone 上容量较大的数据部分中。因此，可以在 iPhone 上安装多少应用取决于

数据部分有多少空闲空间可用。这部分空间通常是以 GB 计算的，因此基本上不构成什么限制。

而对那些通过 Cydia（相当于越狱版的 App Store）安装的越狱应用而言，情况是不一样的。这些应用，比如 Cydia 本身和所有内置二进制文件，都是安装在根文件系统的 /Applications 目录中的。正如之前提过的，根文件系统的大小取决于固件的版本和设备的类型。通常情况下，它的大小在 1GB 到 1.5GB 之间，其中有 200MB 是空闲空间，这就没有为可安装的应用留下多少空间。

除此之外，墙纸和铃声文件也是存储在根文件系统中，分别存储在 /Library/Wallpaper 和 /Library/Ringtones 目录中。因此，所有通过 Cydia 安装的墙纸或铃声都会蚕食已经很有限的应用安装空间。

为了解决这一问题，多种越狱工具实现了一种名为“应用转存”的机制。这种机制的思路是在 iPhone 的数据部分创建名为 /var/stash 的新目录，并将通常位于根文件系统中的若干目录放到该目录中。然后原始目录就会被通向新位置的符号链接替代。

下面列出了当前被转存到 /var/stash 目录中的目录：

/Applications；
/Library/Ringtones；
/Library/Wallpaper；
/usr/include；
/usr/lib/pam；
/usr/libexec；
/usr/share。

不过，并非所有的越狱工具或这些工具的所有版本都能执行应用

转存。如果越狱工具不能完成这一工作，那么在第一次调用 Cydia 时，Cydia 会检测并完成这一工作。这就是 Cydia 中耗时很长的“Reorganizing Filesystem”（重新组织文件系统）步骤。

应用包安装

越狱安装过程的下一步是应用包的安装。根据所使用的越狱工具不同，这一步既可以是安装由高端用户自己创建的定制应用包，也可以安装越狱工具通常会默认包含的 Cydia 应用包。例如 `redsn0w` 接受的应用包是可以用 gzip 打包的 `tar` 存档。它们可以由之前安装的基本实用工具解包，因此越狱程序不需要用于存档解包的代码。

应用包安装过程会遍历 ramdisk 上的每个应用包，一个接一个地将它们解包。在解包期间，`tar` 被告知要保持 UNIX 的权限，这让应用包都设置了值为 root 的 `suid` 位。Cydia 需要这一设置，因为如果没有 root 权限，Cydia 是没法安装新应用的。很有意思的是，由于苹果公司耍了些花招，GUI 应用可能没有在它们的主二进制文件中设置 suid 位。Cydia 的正常工作要利用名为 Cydia 的 shell 脚本，该脚本接着会调用名为 `MobileCydia`、`suid` 为 root 的主二进制文件。

不过，在将应用包解包到 `/Applications` 目录后，应用包的安装工作还没有完成。安装的应用全部都必须注册到特殊的全系统安装缓存中，该缓存是存储在 `/var/mobile/Library/Caches/com.apple.mobile.installation.plist` 文件中的。这是个普通的 .plist 文件，格式如下所示：

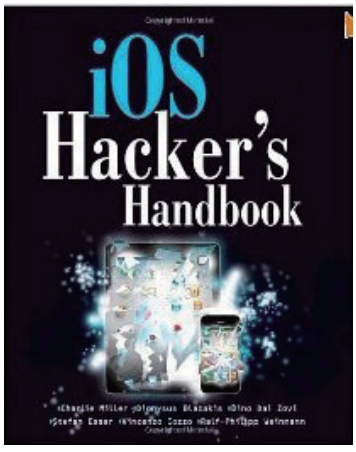
```
<plist version=”1.0”>
```

```
<dict>
    <key>LastDevDirStat</key>
    <integer>…</integer>
    <key>Metadata</key>
    <dict>…</dict>
    <key>System</key>
    <dict>
        <key>com.apple.xxx</key>
        <dict>…</dict>
    </dict>
    <key>User</key>
    <dict>
        <key>someuserapp</key>
        <dict>…</dict>
    </dict>
</dict>
</plist>
```

该缓存包含了一个时间戳，一些元数据，以及与所有的系统应用和用户应用有关的信息。系统应用是那些在 /Applications 目录中的应用，而用户应用则是从苹果 App Store 下载到 /var/mobile/Applications 目录中的。因此，所有的应用包都要在 System 缓存项中注册。在 redsn0w 中，这一工作是通过读取应用的 Info.plist 文件并利用其中包含的信息创建新缓存项完成的。首先，要读取 CFBundleIdentifier 密钥，并将其用作缓存的新密钥。然后要把值为 System 的新密钥 ApplicationType 添加到 Info.plist 文件的字典中。最后，再把整个字典的新内容复制到缓存中。

安装后的过程

在安装好所有的文件后，redsn0w 就会调用 sync() 系统调用，确保所有文件都已写入磁盘。然后，会以只读方式重新挂接根文件



iOS Hacker's Handbook 详细介绍了 iOS 操作系统的安全架构和安全风险，以及如何发现、利用这些漏洞，同时还详细说明了如何开发 iOS 越狱工具，以及 iOS 应用的加密、代码签名及内存保护、沙盒动作机制等。书中涵盖了所有已发现的 iOS 漏洞以及这些漏洞会为 iOS 系统带来的危害等内容，详细地解释了 iOS 的运行本质、系统构架、系统风险，向人们展示 iOS 的真实面目，并以此来尽可能地避免 iOS 漏洞的危害。本文节选自 *iOS Hacker's Handbook*。

系统，以确保所有的写缓冲区都被同步到磁盘上。挂接到 `/var` 目录的数据部分然后就会被取消挂接。为防挂接操作失败，这一过程会一直重复，直到操作成功或是达到一定的重试次数。

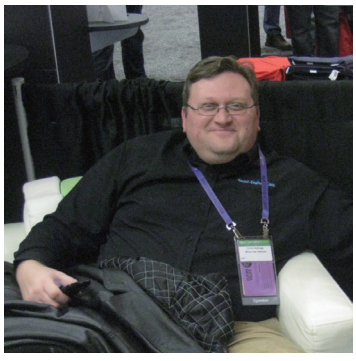
然后就会利用 `reboot()` 系统调用重启系统，完成越狱。对不完美越狱来说，设备会重启到非越狱状态，除非已安装的某个应用包被引导所需的某个文件篡改过。然后需要利用 `redsn0w` 将连接在计算机上处于越狱状态的设备重启。

而对完美越狱而言，设备重启后就会进入越狱状态，因为安装的某个完美越狱漏洞攻击程序会在引导过程中对某个应用进行攻击，然后利用另一个内核漏洞攻击程序在内核中执行代码。

结语

大多数 iPhone 安全研究人员基本都只是越狱工具的使用者，而越狱工具的开发是由 iPhone Dev Team 或 Chronic Dev Team 这样的团队完成的。不过，随着硬件和软件的不断进步，为 iOS 设备越狱变得越来越难了，因此最好是有更多安全界的人士能助越狱团队一臂之力。希望本文能让大家有兴趣在以后参与到越狱开发中。■

SET——社会工程师的工具包



作者 / Christopher Hadnagy

Christopher Hadnagy 从事计算机技术 14 年之久，现在全心投入在计算机安全中“人”的研究。他成立了 Social-Engineer.Com，致力于帮助公司提高安全性，并教授给他们“坏人”的做法。他研究并破解了最近发生的很多恶意攻击事件。他推荐的实践测试法被 500 强公司采用，用来教育员工安全方面的知识。你可以在 www.social-engineer.org 找到他。

译者 / 陆道宏

什么是社会工程?

我认为社会工程的真正定义是：一种操纵他人采取特定行动的行为，该行动不一定符合“目标人”的最佳利益，其结果包括获取信息、取得访问权限或让目标采取特定的行动。

社会工程有很多不同形式，既可以是恶意的，也可以是善意的，既可以具有激励作用，也可以具有毁灭性。社会工程人员也会有不同的目的和类型，他们可能是黑客、渗透测试者、间谍、身份窃贼、不满的员工、高明的骗子、高端猎头、销售人员、政府、医生、心理医生，甚至是律师。

信息收集是社会工程的一个关键方面。若在这一点上投入的精力不足可能会导致社工行动的失败。现在，社工人员可以通过多种工具来收集、分类以及运用这些数据信息。这些工具完全改变了社工人员查看和使用数据的方式。社工人员将不再局限于使用常规的搜索方式找寻数据，这些工具为他们打开了互联网的资源之门。

社工人员花费了大量时间来完善其自身的技能，然而，许多攻击方式需要通过创建附加恶意代码的邮件或 PDF 文档来实现。

这些事情都可以利用 BackTrack 中包含的诸多工具来手动完成。起初搭建 www.social-engineer.org 网站的时候，我曾和好友戴夫·肯尼迪（Dave Kennedy）交谈过。戴夫是流行工具 FastTrack 的开发者，FastTrack 使用 Python 脚本和网页界面能够自动实现渗透测试中的许多最常用的攻击。我告诉戴夫单独为社工人员开发一个类似 FastTrack 的工具是个不错的主意，这个工具让社工人员点击几下鼠标就能创建 PDF 文件、电子邮件及网站等，这样就可将注意力集中到社会工程中的“社会”这部分上来了。

戴夫仔细思考了这个问题，决定创建一些简单的 Python 脚本，让社工人员可以创建附加恶意代码的 PDF 文件并随邮件发送。于是社工人员工具包（Social Engineer Toolkit，SET）就诞生了。在写本文的时候，SET 已经被下载了超过 150 万次，而且很快成为社工人员进行审计时配备的标准工具包。本节介绍 SET 的主要特点及使用方法。

安装

安装步骤十分简单。前提是已经安装了 Python 和 Metasploit 框架。这两个软件在 BackTrack 发行版中已经存在，所以不用担心。BackTrack 4 甚至已经包含了 SET。如果需要从头开始安装，过程也十分简单。依据导航进入安装目录，在控制台窗口上运行如下命令：

```
svn co http://svn.secmaniac.com/social_engineering_
toolkit set/
```

执行完命令之后，将得到一个名为 set 的目录，该目录下包含了所有 SET 工具。

运行 SET

运行 SET 的过程也很简单。只需在 set 目录下输入 ./set，就会启动初始 SET 菜单。

请登录 social-engineer.org 网站，这里有 SET 菜单的完整展示图，以及对每个菜单选项全面、深入的介绍。接下来介绍 SET 中两个最常用的功能。

首先讨论鱼叉式网络钓鱼攻击，然后讨论网站克隆攻击。

鱼叉式网络钓鱼

网络钓鱼是一个术语，描述恶意诈骗犯如何通过定向设计的电子邮件“广泛撒网”，吸引人们访问特定的网站、打开恶意文件或者输入个人敏感信息，为以后的攻击做准备。要在今天的互联网世界生存，必须能够检测、防御此类攻击。

社工审计人员使用 SET 可以创建有针对性的电子邮件对客户进行测试，然后记录有多少雇员上当了。这个信息随后可用于培训，以帮助员工识别和避免这些陷阱。

使用 SET 进行鱼叉式网络钓鱼攻击，选择选项 1。选择 1 后，会看到如下几个选项：

（1）执行群发邮件攻击
（2）创建一个文件格式负载
（3）创建一个社工模板

要进行邮件式钓鱼攻击，选择第一个选项。第二个选项用于创建一个恶意的 PDF 或其他文件，以备作为邮件附件发送。第三个选项用以创建模板，待日后使用。

在 SET 中发起攻击十分简便，只需选择正确的菜单选项然后点击启动。例如，如果我想发动邮件攻击，向受害者发送伪装成技术报告的恶意 PDF 文件，我会选择选项 1——执行群发邮件攻击。

接下来，我会选择一个攻击向量（选项 6），这种攻击对很多版本的 Adobe Acrobat Reader 软件都有效——应用了 `Adobe util.printf() Buffer Overflow` [1] 漏洞。

1 请参见 http://www.nsfocus.net/vulndb/12573。——译者注

接下来几个选项会设置攻击的技术问题。点击选项 2——Windows Meterpreter Reverse_TCP。使用 Metasploit 接收反向会话、或者受害者电脑的 IP 和端口，以避免入侵检测系统(IDS) 或其他系统的报警。

选择 443 端口，使数据流看起来好像是加密的 SSL 数据。SET 会创建恶意 PDF 文件并设置监听功能。

执行上述步骤后，SET 会询问是否要更改 PDF 的文件名，例如改成类似 TechnicalSupport.pdf 等更加隐蔽的名称，然后输入邮件信息以备收发。最后，SET 发出一封看起来很专业的电子邮件，引诱用户打开附件中的 PDF 文件。受害者收到的邮件如图 1 所示。

邮件发送之后，SET 会创建一个网络监听器等待目标打开文件。一旦目标点击了 PDF，监听器就会执行恶意代码，让攻击者得以进入受害者的计算机中。

真是惊人（也许有人并不这样认为），所有这一切只需点击六七下鼠标，审计者便可将精力集中在攻击的真正的社会工程方面了。

图 1　一封无害的电子邮件与一个简单的附件

这是一个破坏性很强的攻击，因为它利用了客户端软件的漏洞，而且在大多数情况下，屏幕上不会显示有情况正在发生。

这只是应用 SET 可以发动的众多攻击中的一种。

Web 攻击

SET 也允许审计人员克隆任何网站并在本地运行。这种攻击类型的强大之处在于可以让社工人员以多种方式诱骗他人访问克隆网站并从中获利。社工人员既可以伪装成更新网站的开发者，也可以仅仅对网址进行细微的修改（填加或删除一个字母），最终诱使他人访问克隆的网站。

一旦有人访问了克隆网站，社工人员便可以发动多种不同的攻击，包括信息收集、证书收集和直接入侵等。

要在 SET 中运行此攻击可从主菜单中选择选项 2（网站攻击），选择之后，可以看到以下几个选项：

（1）Java Applet 攻击方法
（2）Metasploit 浏览器的入侵模式
（3）证书获取的攻击方式
（4）标签绑架攻击方法
（5）中间人攻击方式
（6）回到前面的菜单

选项 1 中的 Java Applet 攻击是一种特别邪恶的攻击。一般情况下，Java Applet 攻击会在用户界面上弹出一个 Java 安全警告，说该网站已被 ABC 公司签名，并让用户同意这一警告。

进行这种攻击，先选择选项 1，然后选择选项 2——网站克隆（Site Cloner）。

选择网站克隆的时候，需要输入你想克隆的网站地址。这里可以选择想克隆的任何网站——客户的官方网站、客户供应商网站或者政府网站。正如你所想象的，重点在于选择一个对目标有意义的网站。

在这个练习中，假设是克隆 Gmail 网站。屏幕上会显示如下信息：

```
SET supports both HTTP and HTTPS
Example: http://www.thisisafakesite.com
Enter the url to clone: http://www.gmail.com
[*] Cloning the website: http://www.gmail.com
[*] This could take a little bit...
[*] Injecting Java Applet attack into the newly cloned
website.
[*] Filename obfuscation complete. Payload name is:
DAUPMWIAHh7v.exe
[*] Malicious java applet website prepped for deployment
```

上述工作完成之后，SET 会询问你想要在自己与被害者之间创造什么类型的连接。要想使用本书讨论的技术，选择 Metasploit 的反向会话界面，也就是 Meterpreter。

SET 为负载加密提供了多种选项，这是为了避开反病毒系统的检测。

下一步，SET 启动内嵌的网站服务器为克隆网站提供服务，同时启动监听器准备捕获浏览该网站的受害者。

现在只需要社工人员构造一封电子邮件或给目标打个电话，让目标访问该假冒的网站。最后，用户会看到如图 2 所示的界面。

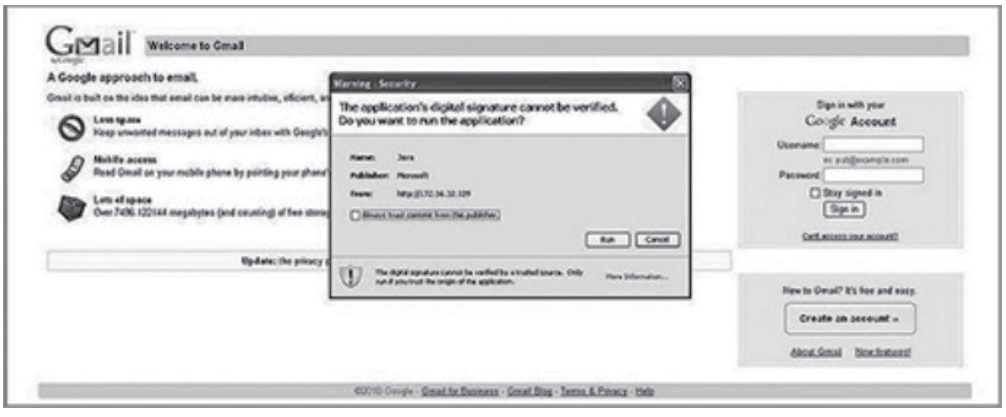


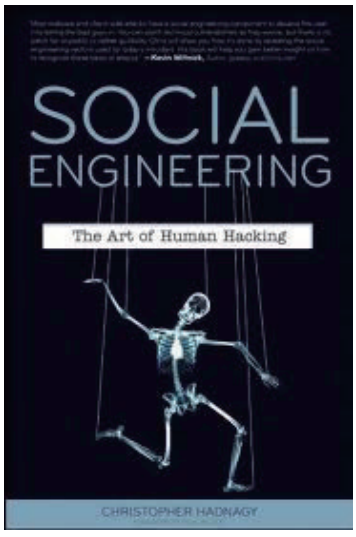
图 2　谁会不相信微软签名的小程序呢?

最终结果是，一个 Java Applet 出现在用户面前，告诉他该网站已被微软签名，他需要允许安全证书运行，才能继续访问网站。

只要用户允许了该安全证书，攻击者就可以立刻入侵他的计算机了。

SET 的其他特性

SET 是具有实战思维的社工人员开发出来的，所以工具集所提供的都是审计过程中常常会用到的攻击方法。



Social Engineering 首次从技术层面剖析解密社交工程手法，介绍了社交工程的所有方面，包括诱导、假托、心理影响和人际操纵等，通过实际的例子、个人经验、及背后的科学，探讨和解释了社交工程的奥秘。本文选自 *Social Engineering*。

SET 在不断更新和发展。例如，最近几个月，除网站克隆和网络钓鱼攻击之外，SET 又增加了一些其他的攻击方式，还增加了一个传染性媒体生成器。传染性媒体生成器允许用户创建带恶意文件的 DVD、CD 或 USB，这些传染源可以混杂在目标对象的办公大楼里。当它们被插入计算机时，将触发恶意负载程序的执行，从而开启受害人机器的入侵之门。

SET 也能创建简单的负载和相应的监听器。如果社工人员想要通过一个提供反向会话功能的 EXE 可执行程序连接回他的服务器，可以在审计过程中携带一个 U 盘。如果面前的机器是他想要远程访问的，便可将 U 盘插入，导入负载文件，然后点击运行。这样可以在目标机器和他自己的机器之间建立起一个快速连接。

有一种较新的攻击方式叫 Teensy HID 攻击。Teensy 设备是一个小的可编程电路板，可嵌入键盘、鼠标或其他可插入电脑的电子设备。

SET 可对 Teensy 编程，设置这个小电路板在插入电脑时将执行何种命令。常见的命令包括创建反向会话或监听端口等。

SET 的最新特性之一是提供了一个 Web 界面。这意味着 SET 会自动启动 Web 服务器程序，从而更易于应用。图 3 显示了这个网页界面的概貌。

SET 是一款强大的工具，它能帮助社会工程审计人员测试出公司存在的常见弱点。SET 工具的开发者总是善于听取他人的意见，在工具中增添新的应用，使得其不断完善、越来越流行。如果想更深一步了解这个强大的工具，可以登录 http://www.social-engineer.org 网站，上面包含每个菜单选项的详细说明。在使用

图 3　SET 的新 Web 界面

过程中，可以通过 http://www.social-engineer.org 和 http://www.secmaniac.com 这两个网站对 SET 不断更新。■

张文君（小包）：

和病毒对抗，和自己对抗

张文君（小包），网名“junzz”，前金山网络互联网安全研究员。负责严重安全事件快速分析和回应，精通木马和各种恶意程式的对抗，对操作系统内核底层机制有深入研究。2009 年底加盟金山网络公司任安全开发，曾负责金山毒霸相关的内核驱动程式开发和顽固木马查杀的工作。曾处理过众多知名流行病毒，如：隐身猫，极虎，鬼影，杀破网，淘宝大

从什么时候开始编程的？

大学的时候开始正式学习计算机，但其实很小的时候就开始接触计算机了，当时一直都在玩单机游戏什么的。我大学学的其实是园林设计，当时我报这个专业的原因是因为它很偏，不是大家都很热衷的专业，说不定我可以到处去旅游。后来发现不是这样的，于是就开始研究计算机了。

那具体是什么契机让你开始学习编程呢？

有一天我的 QQ 号被偷了，我就很不爽，开始研究怎么偷别人的 QQ 号。

（所以你当时的反映不是去腾讯公司挂失，而是去把它偷回来？）

我当时就是觉得这个东西很有意思，木马可以盗号，我很想知道这个东西的工作原理。那是大概 05 年的时候，好像黑客这个概念很火，刷个 QQ 钻啊，拿扫描器扫端口什么的。

（后来你把 QQ 号偷回来了？）

盗，极光，超级工厂，AV终结者等，曾代表金山公司在台湾黑客大会（HIT CON 2011）发表过技术演讲。目前负责金山猎豹浏览器的底层开发工作。

没有，我偷了另外一个 QQ 号，我看见一个以我生日结尾的 QQ 号，我就挺想要的，就试了一下（笑）。

最开始的时候，我并不是直接搞编程。其实黑客和编程最开始的阶段是没有交集的。因为浅阶段的黑客是拿着工具，人家写好的东西来用。这种就比较浅。当时我拿着一些比较有名的软件，比如流光，扫描一些 IP 段，我输入一些 IP 段，从某些部分开始扫，比如北京的，如果中间有漏洞，就攻击它。可以侵入别人的宽带，给你的 QQ 币充值，总之就是一些坏事。但是我当时好像是没充（值），只是试了一下（笑）。我玩这个大概有半年时间，当时我还不会自己写工具。

后来我逐渐发现用工具不够好玩，如果可以自己写工具的话就更好了，于是我开始想研究地深入一些。当时看的书都是网络安全这方面的。我从这个时候就开始正式接触病毒了。所以说我不是以编程者的身份来学习计算机的，而是以逆向、攻击、防范这样的切入点进入的。我当时自己去图书馆看书，基本能借的都借回来了。这样大概学了有两三年的时间。我的学业很荒废，也不去管它，但是最后还是幸运地拿到学位毕业了。因为我从初中开始打篮球，是篮球校队的，所以老师也没有特别为难我，他们也希望我可以按照我的兴趣来发展。

随着越学越深入，我发现要是想防范木马，就一定要学编程。比如要写一个防范工具，仿病毒来写，也要了解语言方面的知识。我的兴趣点比较专，所以只学习 C、C++，还有汇编语言。

后来你是怎么把兴趣做成职业进入到安全这个圈子里来的?

因为不是计算机专业的出身，所以我觉得我有必要系统地学习一

下。所以大四的时候，我基本没有回学校，差不多一天都不在，我去参加了一个在安全界很有名的论坛（“看雪论坛”）当时在武汉办的一个培训加竞赛的机构。首先要参加 C 语言考试，我过了。然后就开始了为期一年的学习。期间我还写了一些项目参加比赛。这种比赛就是要自己写一个东西，让别人来破解，看看需要多少时间。当时我的那个“壳”写得还可以，因为是“混淆”过了的。通过这些比赛，我认识了很多行业中的人，再加上老师对我的推荐，所以在我还没有毕业的的时候，就有很多公司 offer 过来了。当时金山比较符合我的预期，因为从小玩他们的游戏长大，觉得金山有一种国家梦想软件企业的感觉，所以对它的印象特别好。

我当时的同学基本都分散在各个安全公司，360、微点，也有人去做外挂。另外我们这个圈子人脉也很广，有人会去做培训。其实我们这（个圈子）和正常的编程圈子不太一样，比如很多事一旦涉密，就很难脱密了。再比如我们关注点也很不一样，我们一般都关注和攻防有关的知识，包括地下的产业链是怎么运作的。

后来就直接来金山了，负责什么工作呢？有什么挫折吗？

当时我们这方面的人才很少，需求很大。再加上很多人没有选择走正途，所以就业压力不是很大。因为有的人掌握了这种攻防技术之后，他就可以选择去做外挂、做病毒，这样来钱会很快。其实当时我们的同学也有一些组团对去做一些那种……我就不说了。像金山这样的大公司最看重的是基础能力和人品，一定要人品过硬。所以他们招人的时候就会看重你有没有那种邪恶的感觉（笑）。如果做安全的话，一旦你了解了安全公司的病毒，你出来做（病毒）就会非常容易，因为你知道他们（安全公司）是怎么运作的。

我最开始做的是病毒分析、后来做的是病毒查杀，再后来做的就

是再往里面一点，驱动相关的东西。工作中还是需要不断学习的，很辛苦。比如突然出现了一个 0Day，我就要拿着老的 WINDOWS 的源码一点一点看，肯定是哪里有问题，自己一点点测、试、琢磨漏洞能被黑客利用的原理，这些过程都很艰难。

后来在金山和可牛合并的时候曾经有一段时间大家的工作状态都有点疲软，处于一个不太积极的状态。但是我认为（工作上）跳来跳去不太利于自己的发展，到了新环境你要花精力适应新团队、适应新工作。而新的团队可能工作节奏和你的不一样。比如说我们有的同事去了腾讯，就遇到了这个问题。后来和 boss 聊了一些，逐渐心态又好起来了。

后来怎么决定不继续做攻防了？

当时病毒方面的业务已经趋向于稳定了，处于维护期，可以做的事情也比较少，有点像一个养老的状态。现在病毒的趋势没有以前那么猛烈了，以前都是深层次的对抗，他们（黑客）都会做一个驱动，想把我们（安全软件）杀掉，会用一些狠的方式。而现在大家都转向简单、快捷、来钱快的方式。比如他们会把你的首页锁了，导点流量，弹个广告什么的。因为和我们（杀毒软件）深层次对抗的话，他们技术划不来，他们的利益链没有那么长的周期周转，他们想要快的。

国外的病毒，因为大家都很富足，偏向于技术的研究，比如鬼影病毒其实就是抄袭外国的 MBR 病毒，也就是引导扇区病毒，当时在国内会被利用也是因为网吧有还原软件，才会有人把这种相对底层的技术引入国内。中国的病毒行业就偏向于浮躁一点的，利益相关的东西比较多，中国人一般不会研究很深然后把它（病毒）用到一个点上。其实像当年爆发的那些熊猫烧香这类病毒，也没

什么技术含量，只不过是一些很简单的拼凑。现在的病毒一般都在用户可接受的范围内，所以杀毒软件的更新也没什么动力。

后来，我就从珠海来到北京，由于保密需要，他们最开始告诉我是做移动相关的开发，而移动这方面我是完全不懂的，但是没办法，于是硬着头皮买了一堆移动的书看。后来发现其实我是来做浏览器的。

来到北京之后，你就加入了猎豹团队了吗?

是的。我从最开始和猎豹浏览器一起长大，从封闭开发，一直到内测，我都在（团队）里面。一开始就是从零做起，我没有做浏览器的经验，连开源的 Chrome 的架构都不了解。只能一点点探索、学习。从内测版到现在还不到一年。一开始研发、产品、测试，也就十几个人，到现在人太多我也不知道有多少，应该至少是翻倍了吧。在猎豹浏览器我负责的是核的部分，比较基础底层的服务，其他人会在我们的基础上做调用。包括双核切换这些，都是我做的。

我原来的工作偏向于对抗，而现在更像是自己和自己对抗。那边可能实时性比较强，但是现在更侧重于自己知识的不断完善，比如我做的基础接口，我就要保证我自己的能力要不断提升，如果自己没有进步的话，就会很困难。如果我能做得超出别人的预期，我就会比较开心。

以前，如果我能帮别人杀掉病毒我会很满足，现在，我的家人都在用我做的浏览器我也很满足。

业余时间都干什么？融入程序员的圈子了吗？

我不太喜欢逛街、买衣服，买化妆品什么的，我很多时间都花在看技术书上了。我的女性朋友们会问我：你下班了为什么不去逛逛街、逛逛淘宝，而要看技术书呢？我和朋友聊天的时候通常我都是被吐槽的一方，她们会和我分享她们遇到的事情，奇难杂症之类的，我就负责点头就好了（笑）。

和同事们在一起的时候，我没有特别把自己当成女生对待。因为在公司，都是女生当作男生用，男生当作畜生用。我不会因为自己是女生所以这也不愿意干，那也不愿意干。

男码农们有时候很猥琐（笑）。他们讲的好多黄色笑话都不是很好笑，但是他们自己又笑得很开心。这种情况我就自己在旁边玩手机。但是我还是很积极参与团队活动的，不可能因为笑话不好笑就不和团队成员交流沟通了。但是有时候感觉还是同类太少，无论以前在珠海的团队里，还是现在猎豹我们这个组里，都只有我一个女生。

你认为女程序员少的原因？

我觉得首先是兴趣问题。我认识这么多女程序员，真正热爱这个行业的其实就只有一个，而且她还没毕业（笑）。很多人都是因为自己学了这个专业，才从事这个职业，只是把编程当作一份工作。而且她们一般的态度都是不要去民企，最好是可以去比较轻松一点的地方，去教书，或者搞科研什么的。所以我认识的女程序员中，热爱这个行业的，还没有出现呢（笑）。兴趣这个大问题就把很多人拦了下来，大部分人（女性）连用个软件，重装个系统都很痛苦。

剩下的有兴趣的人还会因为其他原因打退堂鼓，比如个人时间少、加班辛苦、家庭问题、个人问题。也正是因为前面这些问题，女程序员就已经很少了，无法形成自己的文化。所以这样造就出的男程序员文化就会进一步加剧女程序员数量少这样的现状。

做码农累不累？想做到什么时候？

刚开始的时候加班很多，比如封闭的时候需要全天都在，现在只有要发版本的时候需要加班，平时就不太需要了。时间上来讲，现在是好多了，至少可以自由出入了。但是工作的压力还是很大，因为每天跟进的问题还是在那里，比如效率不行，性能问题什么的。每天都有反馈，要迭代，我的工作有没有进展都是很容易评估的。加班不是必须的，但是我必须要把我的事情搞完，我作为团队的一份子，过不去自己责任心这一关。我现在住的地方离公司比较远。原来在珠海我住的地方离公司非常近，对面就是海。而在北京这边上下班都要赶地铁，路上的时间都比较紧。

关于做码农，我想的是，只要是我身体条件允许我就会一直做。因为听说我们这行会有一些职业病，比如颈椎病、腰疼之类。可能到了这个不得已的时候就要停下来了。做程序员体力一定要好。我在珠海的时候每天都会跑步，到了北京，发现 PM2.5 太高，不适合跑步。

以后我可能逐渐要向项目管理这方面靠拢，并不是从个人兴趣角度出发，而是因为项目有需求要带人。我不可能一直一个人默默地写代码，团队的运作肯定不是靠一个人，我们需要的是一群努力的人，这样的一群人肯定比挖一个很牛的人过来要更有效率。我不会为了管理而去做管理，我肯定一直都会做技术相关的工作。

赵丹（Diana）：
消灭码农才是正经事

赵丹，1973 年生于天津，1996 年毕业于厦门大学法律系，长期从事软件设计开发工作，现任天津海量信息技术有限公司首席架构师。业余时间爱好户外运动，曾攀登过 5000 米雪山，完成过北京马拉松全程。

最早接触编程是什么时候？

我上小学的时候，我爸爸的厂子里有一个计算室，他经常带我去那里玩。计算室里有两台 Apple 2，所以我经常趁人不在的时候摆弄机器，玩玩游戏什么的。所以我从小学 5 年级的时候就开始编程了，我经常用 Apple 2 上自带的 Basic 写点程序。后来他们厂子又来了一台很小的可编程计算器，夏普的 PC1500，上面也可以敲 Basic 程序，旁边还有一个小打印机，可以在小纸卷上打出程序执行结果，画出各种各样的图。更妙的是那个打印机还能打颜色：红的、蓝的、绿的、黑的，能画出漂亮的图。因为这个机器很小，所以我在家的时候也能玩。

后来是怎么成为一位码农的？

我大学学的是法律。毕业以后我在政府机关呆了两年，觉得太没意思了，所以经常在论坛上讨论各种技术问题，后来在论坛上的一位老大要招人，问我愿不愿意来，还好我还有一个高级程序员证，于是我流窜到北京来，在当时很大的一家互联网公司瀛海威找了一份工作。

99 年的时候其实机会还挺好的，后来被一帮炒作概念的人给搞砸了。当时银海威和新浪、网易、搜狐都打算去美国上市。但是瀛海威比其他公司都复杂一点，它起步比较早，另外股东关系比较复杂。后来大股东把创始人和小股东都赶走了，找了一家香港做上市运作的公司来做上市之前的准备。这些香港人把瀛海威的概念包装到自己的公司上，在香港把自己公司的股票炒高了几倍。后来瀛海威自己也不搞技术，也不建设平台，倒是花了很多钱装修，在东方广场租了一层楼，场面上做得很大，没有任何实质的东西。当时我就感觉到这个公司太不靠谱了，于是就离开了。

后来我创业搞了一家小公司，当时我们本来是做电子商务的，但是为了补贴花销我们也接了一些开发的活儿。后来虽然电子商务的公司死掉了，但是我们开发这边还有项目在做。后来我又接了一些奇奇怪怪的项目，曾经想过把一些项目做成开源，扩大影响，但是因为我个人身体原因，精力跟不上了，还住了一段时间医院，所以也不能进行下去了，毕竟还是命重要啊。

于是我又变成了打工一族。这比一个人扛一个项目轻松太多了，给人打工是件很容易的事情，既不用哄客户，也不用催款，什么都不用担心。我现在仍然在一线开发，我总不能要求我们部门人人都要编程，然后自己什么都不干吧。但是我团队里的小孩子们现在都很给力，所以我现在也很省心。

你现在的工作内容是什么?

我是我们公司一个有点科研味道的部门 Leader。我们做的是数据分析的一些相关技术，我们在网上抓文本，在文本里抽取一些结构化数据，然后再把这些数据添到分析化模型里面做分析，然后为客户生成分析报告。

对于现在的项目，我未来打算开源其中的一部分，开源是一个很好的形式。如果我们做的东西有水准，就会吸引有水准的人和我们一起做，这些人分布在世界各地，把这些人都招进来是不可能的。但是他们可以和我们一起工作，可以帮我们提意见，也会贡献很多思想。全世界想做这件事的人应该有很多。

你从什么时候开始觉得码农是个不人道的职业的?

我从中学的时候就觉得编程这件事太没有技术含量了，我就想，怎么会有人做这种傻事呢？这种事不应该由人来做，而应该相办法让它自动完成。当然那时候想法也不很系统，但是这个想法却一直留在我脑子里。现在仍然需要人来做很多重复的工作，而且这些事还很容易出错，这是主要的问题。“消灭码农”已经变成我人生中一个远大的目标了。当然，我有生之年也未必实现的了。

我的优势是会编程，还有资源在网上做实验，另外还有机会接触大量的资料。我现在做的和智能相关的工作主要还是分析语义。如果只有好的想法而没有做实验条件也是出不来成果的，所以依托这些资源，可能未来会有些成果。但是也不一定，因为要在科研方面出成果就要提出新的理论新的模型，但是作为公司来讲，关心的肯定的是要客户感受到效果。这完全就是两件事了。所以还是要等公司不太忙的时候我才真正有时间来做。

现有计算机的计算模型有什么问题吗?

现在的计算机模型可能只适合解决一部分问题，而面对另外一些问题好像就不太痛快。比如现在的计算机适合完成某些事情，像数值计算，而涉及另外一些逻辑不很清楚、模糊的判断和决策的

计算则不太适合。但是现在很多人在搞的统计学习，就让这些问题的处理成为可能。所以如果把计算机的体系结构改一改，它就会变得很善于处理另一类问题。比如 Google 搞的超大规模神经网络，现在可以识别猫脸。这些计算需要一万多台计算机，如果换成神经元芯片的计算机，可能一台就够。

很多计算模型，比如多层次的无监督学习就能识别出各种东西，Google 可以识别猫脸，那它识别花盆也没问题。但是现在仍然只是停留在识别层面上，也就是只能把一堆像素变成视觉形象，而这些只是感觉层面上的第一步。第二步就是需要有一个综合，就是不同物体之间有什么关系，这步就没有做到。第三步就是识别出的关系所构建出的场景间的互动。接下来我们还要进行抽象的思考。比如我可以用语言描述现在咱们所处的场景，这些语言变成了形式化的符号。在描述的过程中需要进行很多抽象思维。如果能做到这个层次的话，智能就差不多了。也就是说，现在使用的很多识别模型不是分层学习，也不是规则引擎，每一步要用到很多分析工具才能把它说清楚。这些东西都需要自动组合起来，每次组合的方式都不一样，各个输出对应的输入都是什么？它们是如何自动组合起来的？这时候需要的是一种驱动机制。也就是说怎么样“一着急”就把它们组合起来了。

你认为突破性的研究成果缺乏的是什么？

现在最需要的是各个领域的人才跨学科的合作。搞神经外科的人会觉得搞心理学的是一帮神棍，说的都不靠谱。搞心理的又说你们（搞神经外科的人）只能解决零件的问题，整体装出来是什么样你们都不知道。心理现象肯定有神经机制来支持，问题是怎么融汇贯通，神经的什么动作会演变成心理上的某些感觉。

研究神经科学、脑科学、心理学，还有研究计算机、传感器的人，如果这些专家能够综合在一起，要实现人工智能其实没有想象中的那么困难。比如人类在胚胎状态从器官最初的几个细胞，如何生长成一个复杂的器官——眼，然后眼睛如何把视觉信号传递到脑中，如何形成场景的印象。这些过程其实都已经有人研究过了。现在最缺乏的应该就是学科的综合。

你说过现在的计算模型是男性思维的产物，你认为女程序员数量少是什么原因?

现代的科学知识体系是男性科学家为主导开创出来的，后人逐渐丰富，变成一个完整的体系。过去基本都是男性来做这件事，女性突破性思维比较少。这主要是因为男性的思维方式比较容易在男性主导的世界获得认同。女性也会有各种各样的想法，男的可能搞不懂女性在说什么，可能会觉得:这个没用，这不是瞎胡闹吗?在以前信息的传播很受限制，思想要是不写下来就没人知道，而男性总是优先获得受教育的机会。而思想被记录下来的成本很高，所以只有主流社会认可的东西才会被记录下来、传播开。所以在这样的社会条件下，女性的思想是很难传播和留存的。而现在就没有这个障碍了。但是要想把这些都纠正过来，就需要很长的过程。女性参与进来之后就会制造适合女性的工具、知识体系，以及思维方法，然后才会有慢慢的改变。

你觉得需要鼓励更多女性加入程序员的行列吗?

程序员这种职业无关乎男女，但是至少不需要干重体力劳动，女生可以胜任。我们的部门女多男少。因为男性的社会压力比较大，要求待遇也比较急迫，目的比较功利。但是涉及研究的话，思维

要开阔一点，不能成天想着这些事，相比之下女生更放得开。

另外，从计算机发展的角度来说，由于男性思维的局限性，他们能想到的大多数已经都有人想到过了，有一些东西可能只有女性才能想到。因为这行女生比较少，所以还没人想到，这也是女程序员的机会。 ■

推荐阅读：消灭程序员需要百年吗?

ThoughtWorks 贾永娜：
挑战 = 成长

贾永娜，Thoughtwoks Senior Developer。08 年毕业于西安交通大学计算机系，获硕士学位，之后加入 Thoughworks 工作至今。期间，主要以 Java 开发为主，包括 Groovy，Scala，同时也参与过以 .Net，Python 等为主的项目。在项目中担任过开发和项目管理等角色。是「个体与交互 - 敏捷实践指南」一书的第一译者

你从什么时候开始学习编程的?

我报专业的时候其实是因为机缘巧合，当时正值舆论鼓吹“互联网泡沫”的时期，计算机是个夕阳专业，很多人都会避开。家人的意见是读医科大学，但是最后误打误撞来到了计算机专业。学着学着竟然发现计算机很适合自己。大一的时候的感觉还是像高中生一样看书、学习。而计算机是一门实践性很强的专业，看书只能是一方面，必须要做东西才有感觉。大二大三的时候学校组织竞赛，做网站，于是和同学组队参加比赛。做的过程中我发现了乐趣，我发现我可以解决别人解决不了的问题，这一点点的成就感就激励着我花更多的时间把它做得更好，我慢慢地发现自己越来越喜欢计算机了。

大学毕业的时候我感觉自己仍然是只知道学校内的东西，其实和工作上可以利用的东西还差的很多。我当时只是看到了冰山一角，缺乏全面的认识，所以我觉得我需要一个过渡期，于是报考了研究生。读了研之后我发现确实如此，读研过程中我接触到了很多项目，也会接触到各种专业方向的同学和老师。学校为我们提供了很好的平台和很多的实习机会，这也使得我们有了一个选择的过程。如果我想未来做什么，我就要去什么公司，去这家公司又需要什么准备。实习的时候才会接触比较真实的软件开发，跟在学校的时候很不一样，如果我本科直接就业的话可能就会比较盲目。

你是如何选择职业生涯的起点的?

经过在 Sun, IBM, ThoughtWorks 实习之后我就有机会接触不同公司的风格，我就可以知道我比较适应哪个公司的企业文化。我感觉自己更适合 TW，所以就来了这里。这里对一个刚毕业的大学生来说也确实是一个很好的平台，因为它不会把你固定在一个专门的部门、专门的岗位上，但是在这里我们可以做选择，公司也会尊重我们自己的兴趣。在项目里的时候，很多资深 senior 程序员会帮助我们成长。当我不知道写代码是怎么回事的时候，有人手把手教我和我自己想肯定是两回事。这种成长的速度是相对较快的。刚开始的时候会有很多培训，还有一个专职导师，在个人成长、职业生涯、个人兴趣上都会给我很多建议，解答我的疑问。

另外，ThoughtWorks 内部关系是很扁平的，没有明显的层级关系，这样我们就不会感觉很压抑，或是为了爬这个“层级”而去做一些不必要的事。所以更多的时候，我们可以用更积极的态度来做自己想做的事。

公司里面的工作环境一般都比较轻松，这里穿着随意，工作的时候有时也会放各种各样的音乐。另外，TW 里女生比业内其他公司要多，在招聘的时候，如果在业务水平相当的情况下，是女生的话就是个优势了。

工作中遇到的最大挑战是什么?

工作上每天都会有挑战，当遇到真正的挑战时，也就是个人成长最快的时候。我现在回顾过去，发现以前的很多事情就是这样的，比如有的项目本身比较难，客户要求多、而技术方面自己又不是很有

把握的时候。客户说他需要一个什么样的技术，我们就会采用他的要求，而这样实现之后如果出现了大的性能问题，虽然最初的技术栈是客户提出的要求，但是只要是我在做这个项目，我就要想办法把它搞定。于是我就思考为什么会变成这样? 如果当初的没有这样做的话是不是就不会有现在的问题? 虽然需要解决的问题很多，但是上线的日期不会因此而变更，我既要解决问题，也要保证按期上线。

曾经有一个项目，最后做到(项目上)很多人都离职了。最开始(项目上)还有 20 人左右，做到最后只剩下两三个人了。现在公司提到这个项目大家还心有余悸，只有“恐怖”可以形容。客户难搞、技术难搞，还要解决项目自身带来的性能问题。我当时也在那个项目上，我决心要把它搞定，所以一直跟到最后项目完成的时候。从开始到最后，这个过程很辛苦。当时没有感觉，但是现在想起来，这应该是我成长最快的时期。这个项目之后我写了很多总结，无论是从软件开发角度，还是从团队建设角度，也有从性能技术角度的。不管别人怎么想，“不好玩”、“很难受”、“工作不应该是这样的”，对于我来说，把它搞定的收获才是我最期待的。如果一有不爽就对自己说算了，我对自己是没法交代的。

刨除学习之外的业余时间多吗?

我出差很多，曾经有一年 7 个多月都在外面，还有两三年的春节都是在外面过的。虽然公司加班时间很少，但是我的私人时间仍然比较少，这是因为上班外的时间还有很多要用来学习。如果一个项目要求我学很多东西才能完成，我认为这是件好事。因为通过接触不同项目、不同技术，我的能力就会得到提高。以后再遇到类似的东西会简单很多，上手会很快，感觉也会很准。我期待未来的业余时

间会随着我的能力的增加而越来越多，事情也应该是这样的。

选择计算机这个行业就是这样，技术更迭很快，所以如果不能保持这种状态，可能很快就会掉队。从而失去一种直觉上的把握。计算机这个行业其实也很适合女性，我认为保持持续学习的状态是一种健康的生活态度。如果我更多地学习，我就能更有效的工作。而不学习的话，就会需要更多的时间来工作。而一个健康的学习状态对于任何人的职业生涯都是很好的。

业余时间都做些什么？现在不是很流行慢生活吗？

是吗？现在有这个潮流吗？听起来像是老人的生活（笑），我爸每天就是这样的。可能我的性格还没有到那种状态，还不适应那种生活。我爸就会无聊地喝茶，喝一天也没什么事。

我们中午吃饭的时候或者晚上回到家我会在 github 上搞一些开源的小项目。公司内部有很多这种自己做的项目，公司里很多人都会自己做一些，成不成就看自己有没有时间投入吧。我和同事们一起写过 CI 的工具，大家也是有一搭没一搭的来写写玩玩。我们都是私下联络一下，然后找几个人一块做。

我认识的女程序员除了写代码之外，也爱逛街。但是每天逛街也是不太可能吧（笑），所以能够每天带来快乐的可能就是写代码了。

你现在还是在一线编程吧？想做码农到什么时候？

我现在一方面是 Tech Lead，一方面是 PM，也包括和客户以及 stakeholder 的沟通，做的工作比较杂。

我肯定会走技术这条路线，但是现在的项目 PM 没有了，Tech Lead 去了别的项目，如果我有能力的话我就要承担他们的责任，这些责任在合适的时候也可能会分给其他组员。现在总在忙项目管理这方面的事，但这不是我想专注的领域。写代码、做技术才是我乐趣和成就感的来源。

我想一直做码农，别的没有考虑过。这个行业就是这样，开发、管理、加工，写代码的和不写代码的。我也做过别的工作，但是还是写代码比较爽。■

读《码农》吐吐槽还能赚银子！

《码农》电子刊如今慢慢悠悠已经出版了 6 期，从一开始的好评如潮，到现在的“怎么这么抽象啊”“赶脚内容没有以前好了？”“一下就翻完了，没什么内容啊”……小编表示压力很大，现在的码农读者太难伺候了，明明是本免费杂志，￥%#%&*%@# ￥*@ ￥#@！

但是，一看到好评，盼盼姐还是会热泪盈眶，热血沸腾，各种正能量啊“这么好质量的杂志还免费，天上掉馅饼呀”“希望一直出！每一版都读了，很值得推荐！！”……

所以《码农》还是会一直做下去，但是，是好是坏，您得给个话儿啊！

活动规则：在吐槽贴中留言，选出任何一期中你最喜欢的文章和最不喜欢的文章，即可获得图灵社区银子 2 两！凡吐槽吐得掷地有声者，加赠银子 3 两（共 5 两）！（怎样使用银子兑换图书）

活动时间：本活动长期有效。

一点心得：

从 Eclipse 转移到 IntelliJ IDEA

作者 / 付学良

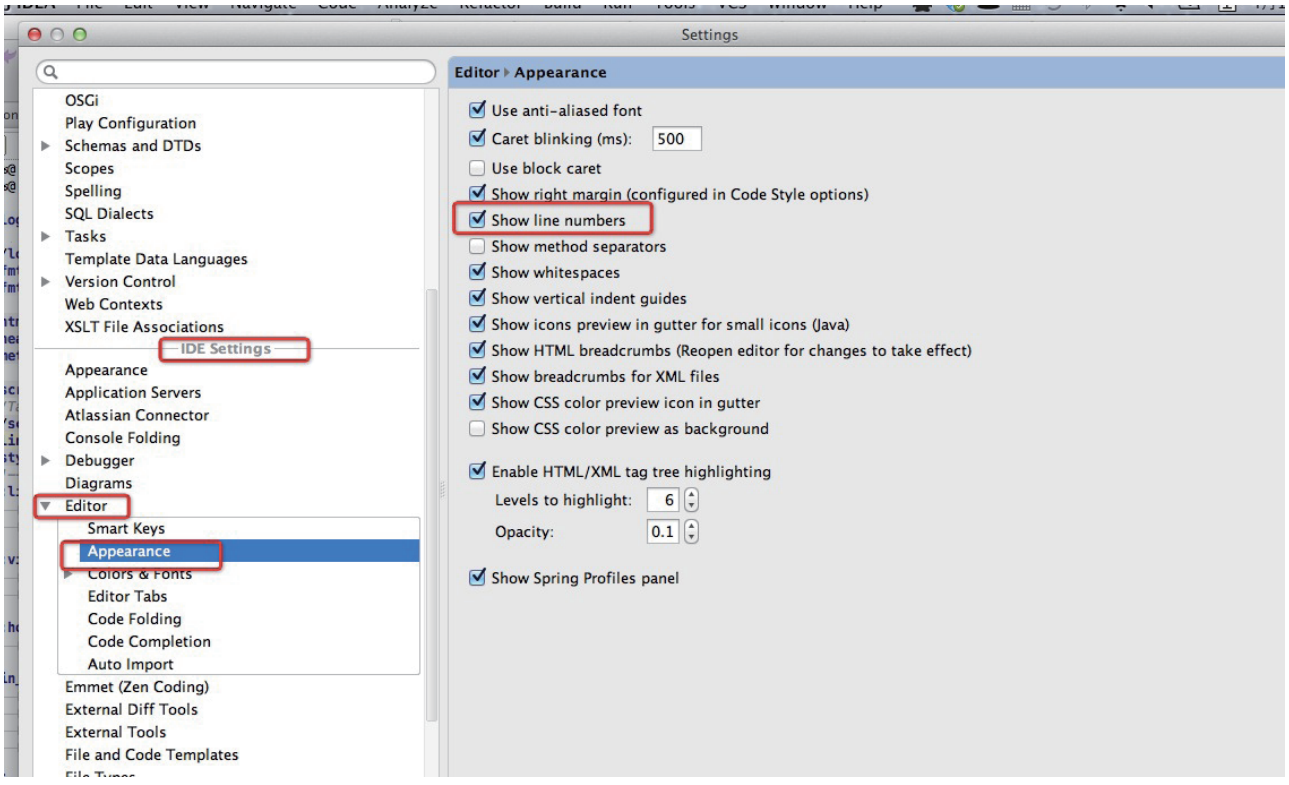
码农一枚，崇尚快乐工作、享受生活。

本人使用 IntelliJ IDEA 其实并不太久，用了这段时间以后，觉得的确很是好用。刚刚从 Eclipse 转过来的很多人开始可能不适应，我就把使用过程中的一些经验和常用功能分享下，当然在看这篇之前推荐你先看完 IntelliJ IDEA 的 20 个代码自动完成的特性，这篇文章对自动完成的特性介绍的非常好，只是快捷键说明部分还可以更完善一些，可以照顾下默认绑定键位不一致的童鞋。本文在说明快捷键的时候是根据 Mac OSX 10.5+ 的默认 keymap，当然也会提供相关的文字说明，如果和我的绑定不一致的可以直接去 keymap 中搜索即可。

基本设置

显示行号

这个以前在 Eclipse 中是个很简单的问题，转移到 IntelliJ 上以后的一些人也问过这个问题。使用同样的方式设置了以后发现只是对当前文件起作用了，打开另外一个文件依然没有行号。就好比你只是在 vim 中执行命令 `set number`，而不是在 .vimrc 中设置一样的效果。这个是需要在 IDE Settings 中设置下的，如图所示勾选即可：

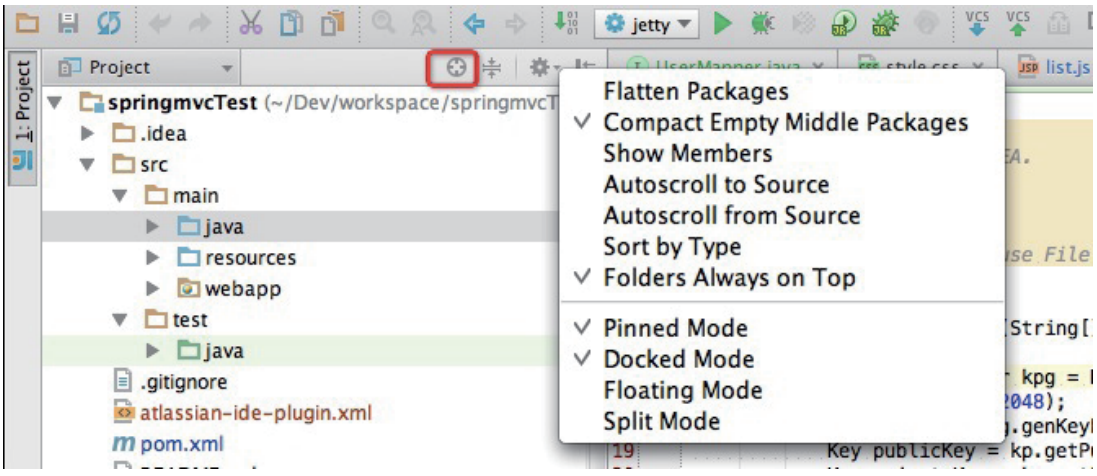


其他设置可参照该设置项勾选即可，例如显示空格等等。

文件和导航关联

我们在写代码的时候经常需要打开某个文件的时候，想在左边的文件树上自动打开关联到位置，这样很方便的找寻本目录下的其他文件或者其他的相关文件，其实这个 IntelliJ 提供了好几种解决方案，第一个就是和 Eclipse 一样的方式，如下：

以前在 Eclipse 中设置也很简单，但是 IntelliJ 里面稍微有点不一样，如果想临时一次的话，打开文件的时候点击下图的小地球（第一眼看的时候很像个小地球仪）图标即可。



如果你想和 Eclipse 的那个一样，让这一切自动的话，那么在上图的浮层中可以看到两个选项 `AutoScroll to Source` 和 `AutoScroll from source`，IntelliJ 很人性化的分了两个设置在里面，你两个都勾选就是基本和 Eclipse 的功能保持了完全一致，只是勾选一个自然也可以。

还有一种方式就是利用 IntelliJ 提供的 **Navigation Bar**，IntelliJ 很多功能操作都可以键盘化，这个就是喜欢键盘化操作的人的必备功能，操作方式就是打开文件的时候，使用快捷键 ⌘+UP (`Jump to Navigation Bar`)，然后就会出现一个下拉，基本就是这个文件所在目录的文件列表，还可以使用**上下左右的键位**来选择其他目录，是不是很酷?

备注：此处建议把 `View > Navigation Bar` 的选项点掉，这样就使用快捷键之后就会在你鼠标的位置出现，感觉会更输入一些，当然这样你就基本失去了使用鼠标的权利，这个需要你根据个人癖好使用即可。

快捷键使用

如果你是苹果电脑用户的话，建议你改成 Mac OSX 10.5+ 的键位绑定，因为 Mac OSX 10.5 这个兼容 Windows 的版本的键位实在是太蹩脚了，用着是真难受，好比你默认习惯 ⌘+w 是关闭标签，在这个键位绑定中竟然是根据符号选择文本的一个快捷键。

这里主要介绍几个可能大家常用的，或者是你可能并不知道 IntelliJ 中有的几个快捷键，特别常用的你直接参照官网的 Keymap 介绍 Getting Started 就可以了。学会在 Keymap 中自己查询需要的快捷键是主要的，不要每次都打开那个没用的百度，具体可以参照本文最后给出的链接。

Outline 调出

以前在 Eclipse 中用的很多的一个快捷键就是 ctrl+o，这样可以调出代码文件的所有成员变量和方法，并且可以支持搜索很是方便，IntelliJ 这种基本的功能自然是有，新手的话还经常问，这个在 IntellJ 中的快捷键是 ⌘+F12 (File Structure)。如果你不是 Mac OSX 10.5+ 键位绑定的话，小括号中的内容就是你可以去 Keymap 中去搜索的内容。

基本操作

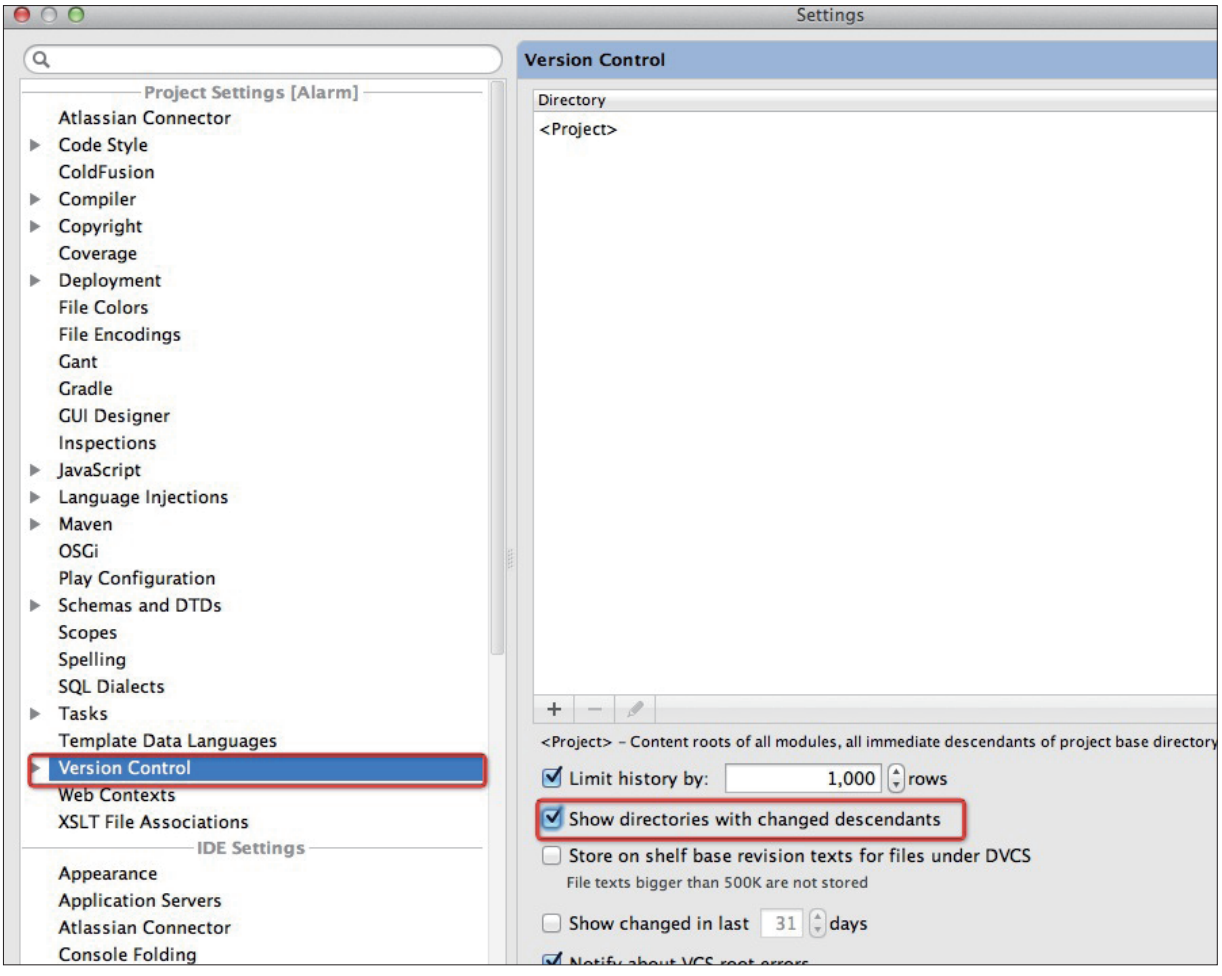
- **复制整行或者块**：在 Eclipse 中使用 ctrl+alt+Up，IntelliJ 默认是 ⌘+D (Duplicate Line or Block)，其实这个和 Eclipse 中的有差距，尤其是多行的时候或者选中内容的时候，不信你可以试试。所以你可以把 Keymap 中的 Duplicate Lines 定

义上一个快捷键，我定义的是 Option+D，这个键位在我的默认绑定中是没任何快捷键的，你可以自己绑定一个试试。

- **根据符号选择文本**：这个键位用的太多了，并且也是非常的好用，举个例子，有这样一行代码 <name>SpringSource Milestones Proxy</name> 你光标在 Milestones 中的某一个位置，你想替换 SpringSource Milestones Proxy 为其他单词，假设想替换为 Test XX，其实这样你只需要 Option+Up (Select Word at Caret) 两下就可以选中你想要的单词了，当你想在开发过程中，你会发现这个键位用的太多了，多到换了其他没有这个键位的编辑器或者 IDE 上，很是不习惯。
- **删除整行**：可以直接 ⌘+delete (Delete Line) 或者利用 IntelliJ 智能的功能，剪切的快捷键，当本行选中文本的时候剪切 `⌘+x 会按照选中的文本实现剪切，但是你本行内无任何选中文本的时候会自动剪切整行，那么自然就类似一个删除本行的快捷键，但是这个其实还是比较蹩脚的，建议使用 ⌘+delete。
- **想直接跳转到接口的实现方法**：⌘+option+ 鼠标单击或者 ⌘+option+B。
- **语句自动完成 Statements completion**: 这个在 IntelliJ IDEA 的 20 个代码自动完成的特性中已经说明，这里还是着重说下，因为的确很好用，也强烈推荐下，还有文中说的智能自动完成 ctrl+shift+Space，当给你的提示太多的时候，这个键绝对会给你惊喜。
- **同样单词之间跳转**：在 Eclipse 中的快捷键是 ctrl+k，这个挺好用的，在 IntelliJ 中也不缺这个功能，但是在 Mac OSX 10.5+ 的键位绑定中默认为空。据说 Windows 的默认绑定中是有绑定的，自己可以查询下，我自己绑定了一个 ⌘+ctrl+k (Find Word at Caret)，没有的需要自己绑定一个，然后还能结合其他的键位实现上一个 ⌘+G (Find Next) 下一个 ⌘+Shift+G (Find Previous) 的功能，很是实用。

版本控制设置

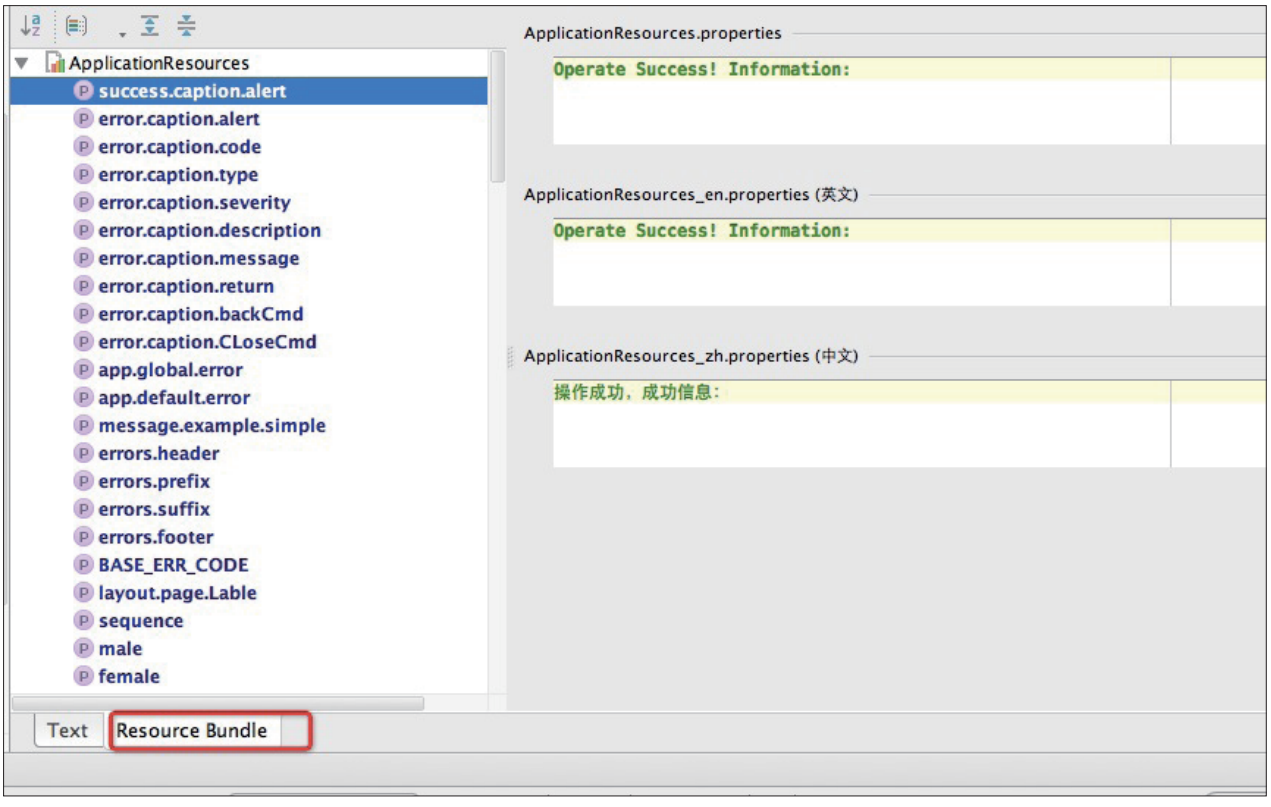
我们开发肯定是有版本控制的，大家以前 Eclipse 的时候在本地文件和版本库不一致的时候，那么文件以及所在的文件夹都会出现一个〉表示，大家能很轻松的看到本地文件修改了哪一些，但是 IntelliJ 中默认是不能这样的。仅仅是给变化的文件在修改的时候提供了颜色上的变化，不包括其所在的各个父级文件夹哦，如果想和 Eclipse 一样的话需要如下图把设置勾选就可以了。



其实看英文解释就可以明白了，这个默认开启以后如果想调整文件夹的显示颜色的时候就直接去 Settings > Editor > Colors & Fonts > File Status 中调整即可了，可以调整成自己喜欢的样子。

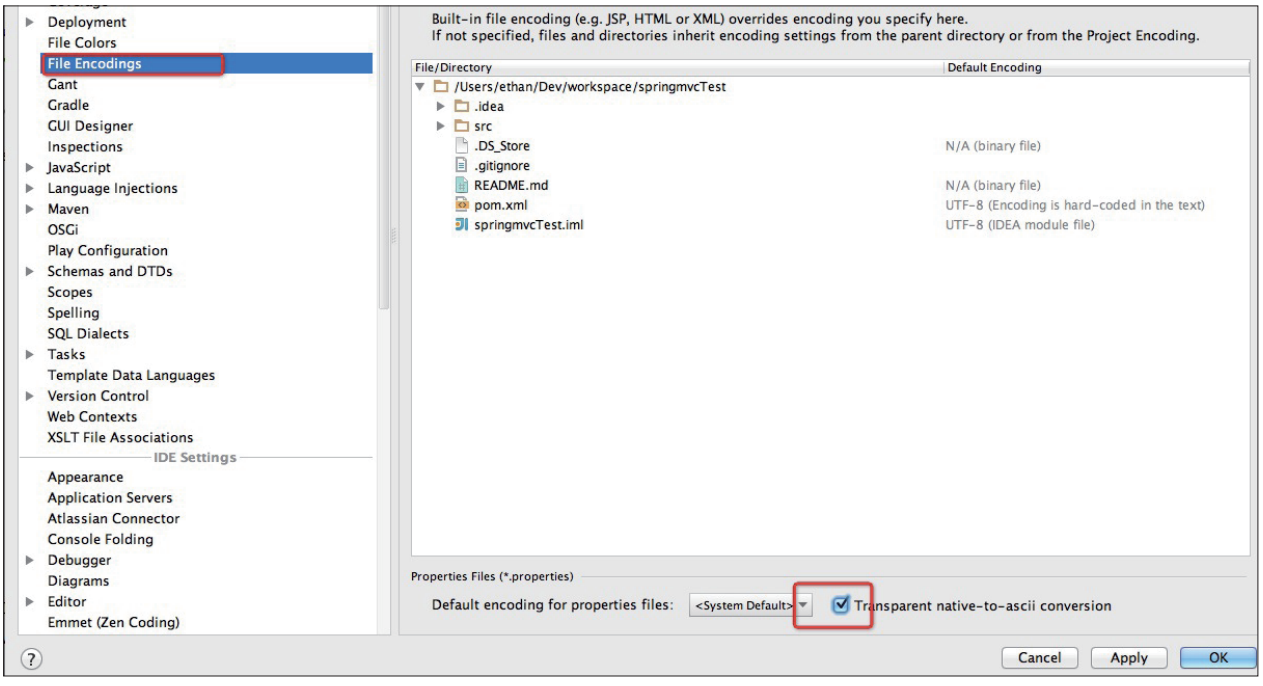
Java 资源文件非英文的情况显示

IntelliJ 在资源文件这方面个人觉得也是完胜 Eclipse 的，在 Resource Bundle 方式下想修改一个属性是相当方便的，可以在一个操作界面上修改所有语言的属性，这一切都是自动完成的，如下图所示:



并且还很贴心的有一个排序和分组的功能，尤其是这个排序。

默认设置下也有一个问题，就是中文默认会显示为 Unicode 码的，其实勾选一个设置(`Settings > File Encodings`)就可以了，如下图:



这样设置以后所有 Unicode 显示的就可以自动转化为其应该显示的语言了。

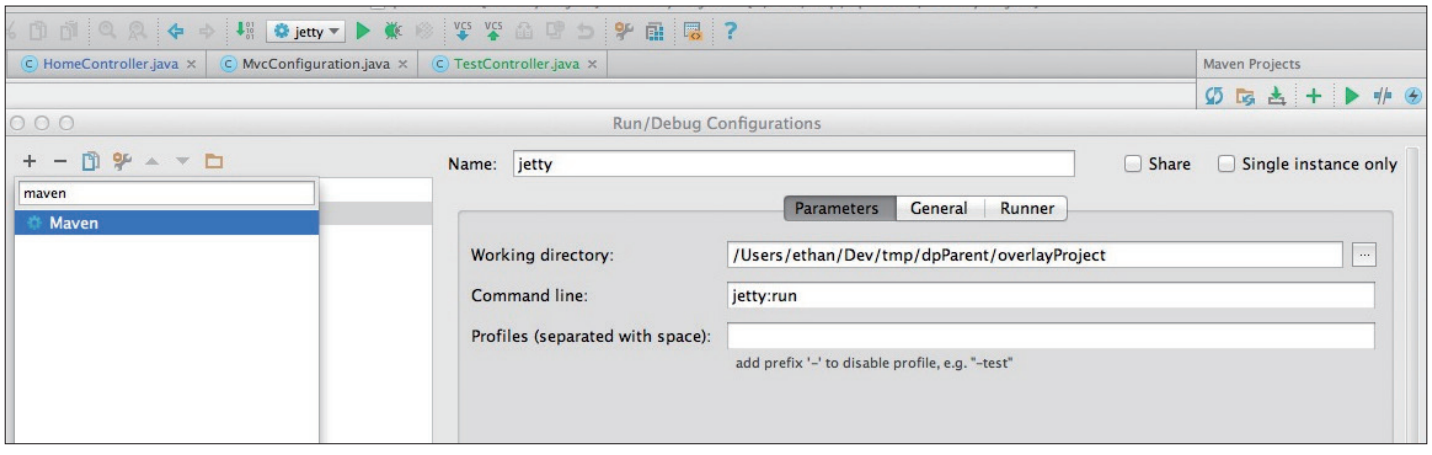
符合 Maven 约定结构的 web 项目如何更好的运行

记得刚开始用 Eclipse 的时候那时候默认是不支持 Java EE 项目的，必须使用一个 MyEclipse 的插件，直到今天好似还有人在用这个玩意，当年的时候每个 Web 项目修改之后都必须 Deploy 一

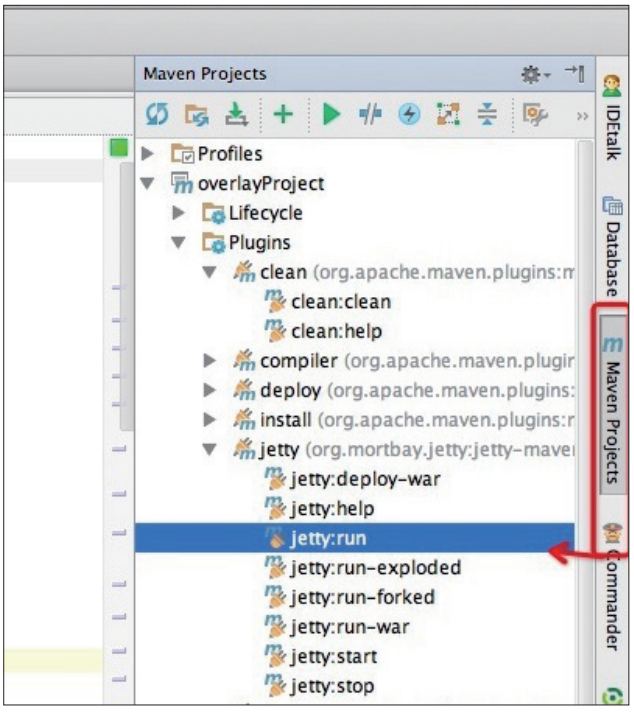
下，修改的东西才会被执行，很是痛苦。后来就自己利用 Tomcat 插件，这个可以保证你修改 jsp 以及静态文件的时候不需要 Deploy 直接就可以被执行，因为 Tomcat 的目录指向其实直接是指向到了你的项目文件，这样的确省心了很多。

在 IntelliJ 中如果你是传统的 Java EE 的结构 (根目录下面有 src，webMoudle 这样的结构)，那么在直接在 Run Configurations 中添加一个 Tomcat 即可，点击 + 号添加即可 (当然需要注意虽然没搜索框，你输入字符会自动给你筛选结果)，这样运行的时候修改 Java 文件的话需要主动 Make Project 也就是 ⌘+F9 (Make Project). 但是如果你是符合 maven 约定的项目结构的话如果继续这样配置就很不理想了，每次修改了 jsp 文件竟然也需要 Make Project 才可以，因为每次 Make Project 以后 IntelliJ 会自动给你 DeployMent，这样好似又回到了几年前那种很麻烦的环境。

这个问题其实很容易解决，就是不要使用这种方式来运行你的 web 项目，既然符合 maven 结构的项目自然是使用 maven 或者 gradle 这种构建工具了，那么你需要利用 maven 的 jetty 的插件来运行项目即可了，如下图配置即可：



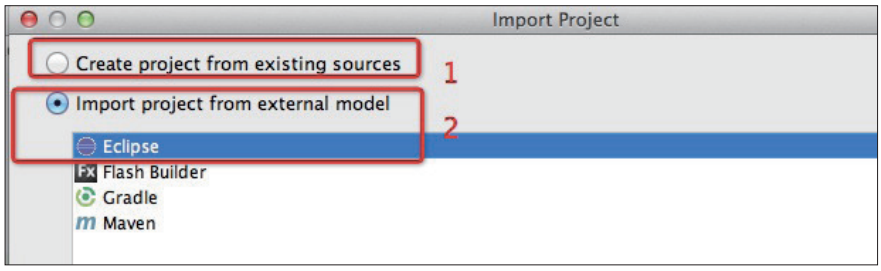
当然你也可以如下图直接点击运行就会自动添加一个：



参看上图其实也没必须非用 jetty，用 tomcat6 或者 tomcat7 的插件运行也可以。

Eclipse 的 web 结构项目如何导入和运行

鉴于有一些用户还没转型 Maven 等来构建项目，当然我也衷心的说一句，构建这块还是赶紧转了吧 . 因为不转的话还单独出来这个章节，并且有的操作还是相当麻烦，下面也会讲到，还是一一道来吧，导入的时候有两个选择然后我分两个部分说明，点击 `File > Import Project…`，然后选择你的项目目录，点击OK，如下图：



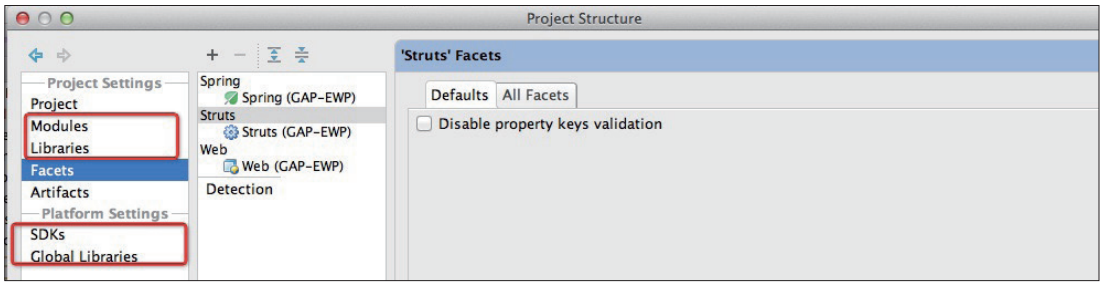
使用新建项目来导入

这个方式图中所示的第一种，导入基本就是一路 next 到完成，但是你第一次也可以费点心思注意下每一步做了啥，例如帮你分析了 jar 有哪一些，如果你使用了 Spring 还给你分析你的所有配置文件。最后完成以后项目就开始构建索引，等一切就绪以后我们就可以完成的基本配置了，由于是 web 项目基本上能够在 Tomcat 等服务器上跑起来就基本算成功了。主要几个点：

- 处理好依赖关系，因为我们没使用 maven，所以你得自己动手丰衣足食了
- 设置 java 文件的编译路径，如果需要的话
- 添加你的 Tomcat，运行成功

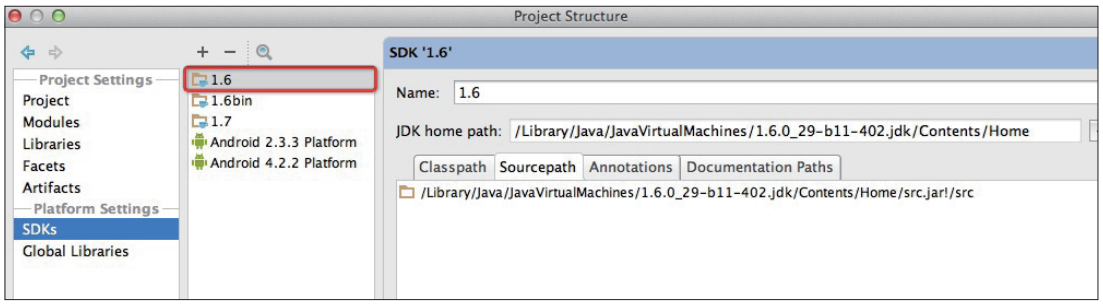
依赖关系

打开 `File > Project Structure`，本节我们主要用到下图几个标红的部分：

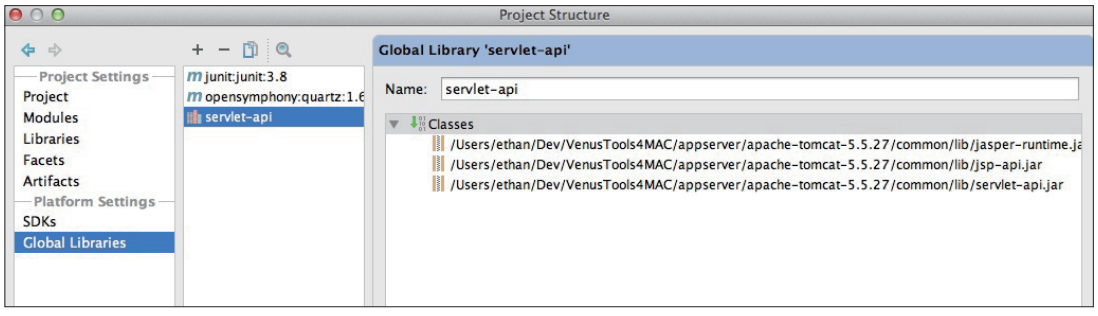


我们先了解下这几个主要部件：

- **SDKs**: 主要存放了我们的 JDK 等，如下图 (我自己下载带 source 的 JDK，这样可以直接查看 JDK 相关的 Java 源代码):



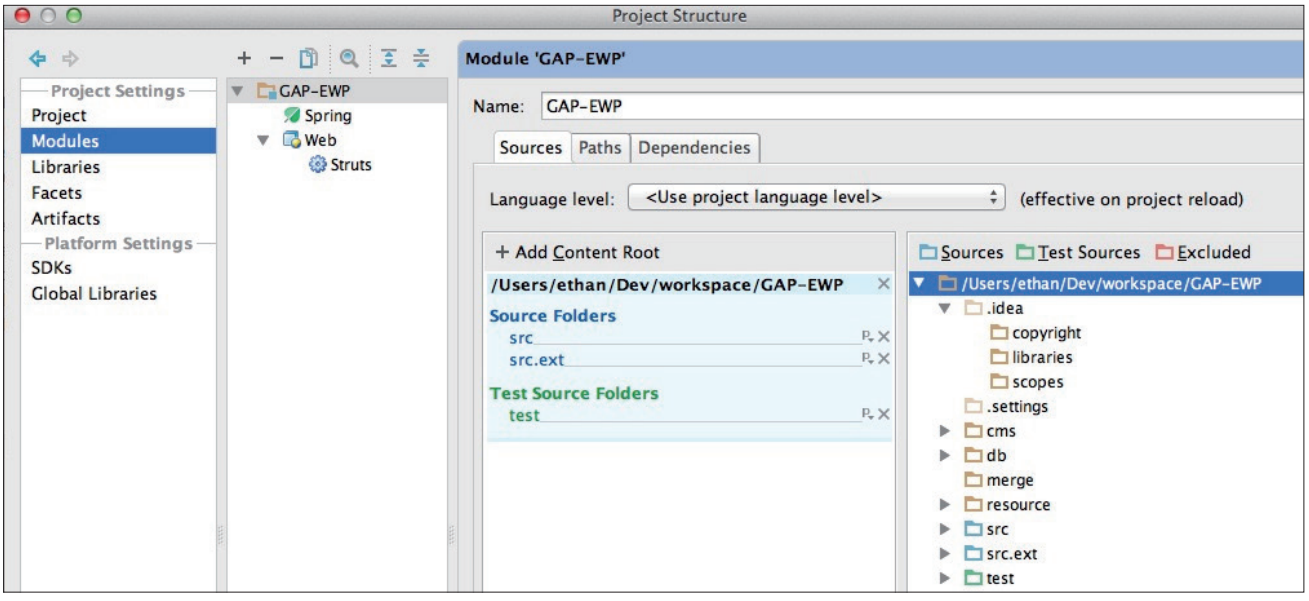
- **Global Libraries**: 主要是公用的 jar 文件，例如 servlet 的几个 jar 可能每个项目都需要，但是 Eclipse 的变量在此变得不好使，所以需要使用这个地方来配置，如下图参照使用即可：



- **Facets**: 在此不做介绍，如有需要后续添加。
- **Libraries**: 这里基本是项目的库，不是全局的，导入的项目一般都会把你 Eclipse 中的。classpath 中的所有的依赖都导过来作为一个 lib 库，然后应用到你的项目的模块中。
- **Modules**: 这个基本包括了项目的模块，通常来说 Eclipse 的项目结构中是单模块的。

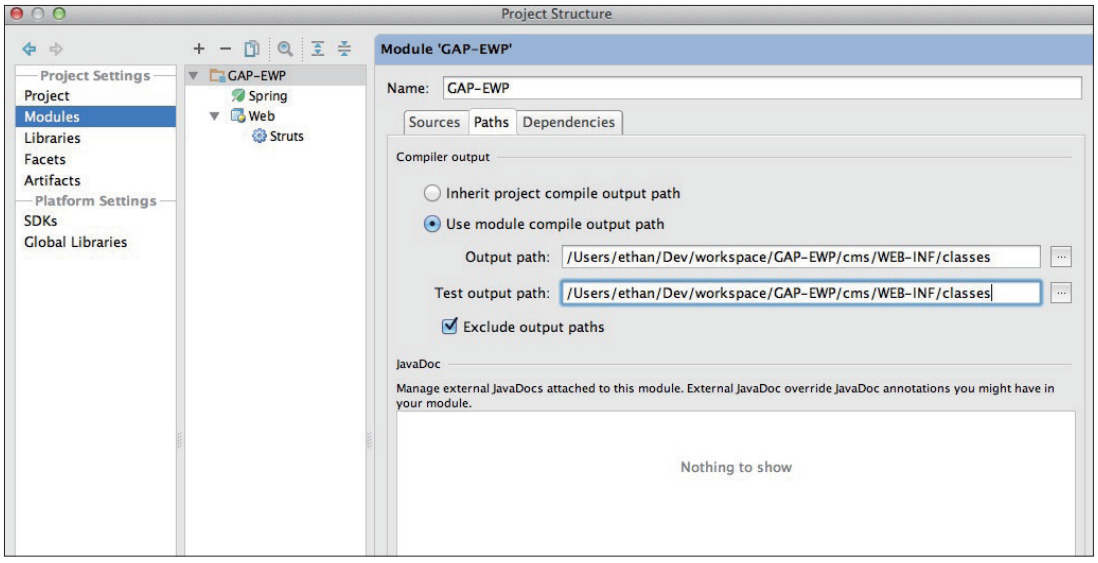
我们需要做的配置主要在 Modules 中，其他的几个自己根据理解配置即可。

1. 我们先查看 **Modules** 中的 **Sources**，如下图：



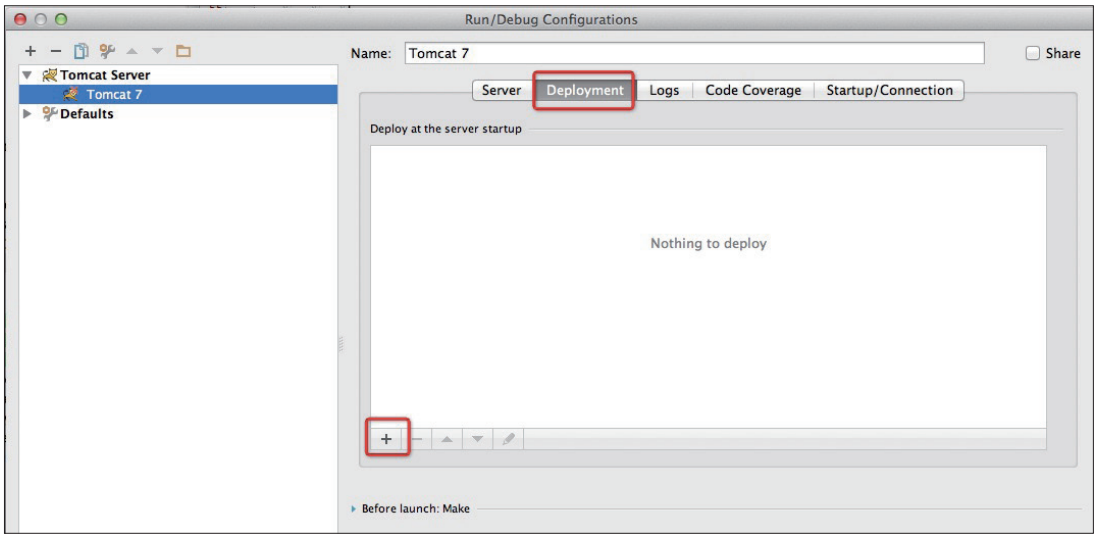
把项目所有的源码都添加即可，图中所示我的项目中其实少了 resource 的文件夹，这样我选中 resource 然后点击 Sources 即可，这样就会在左边的 Source Folders 中添加一个 resource 的文件夹。

2. 然后是 Paths 部分，这个主要是 class 的输出路径，默认是在根目录下的 out 文件夹中，如果项目中对配置文件的解析等有写死路径，必须严格遵循 Java EE 的规范的话，那么需要将此部分修改如下 (如果不存在我假设的情况可忽略本步骤):



3. 然后的 Dependencies，就更好配置了，添加需要的依赖即可。

这些步骤完成以后我们就可以点击`Run > Edit Configurations`，点击左上角的 + 号，然后选择 **Tomcat > Local Server**，这些步骤和 Eclipse 类似，选择相应路径等等，只需要注意一点就可以，就是需要添加一个 DeployMent，如下图：



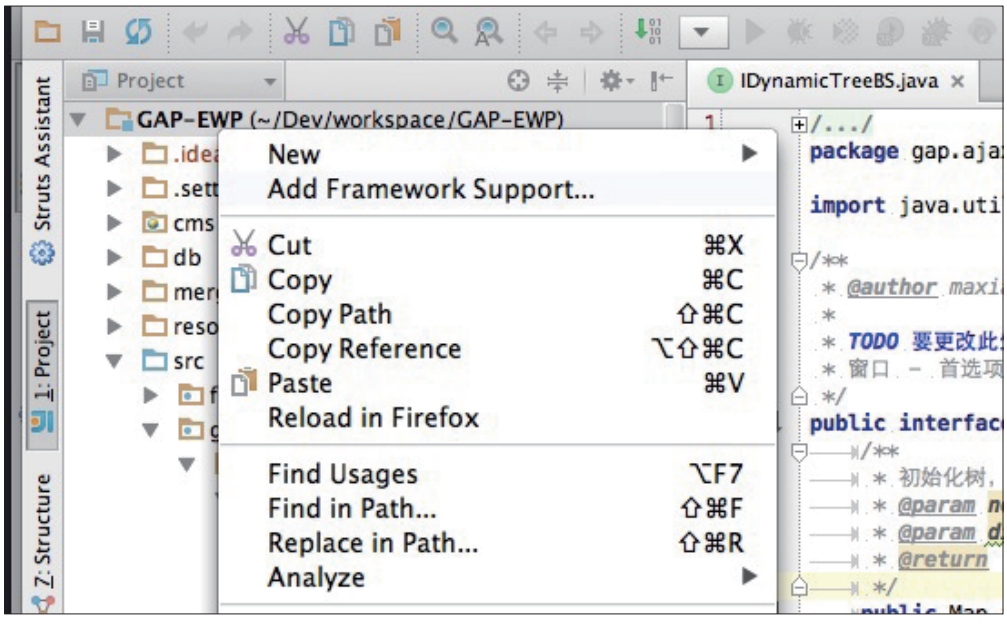
如果是 Windows 的图中的 DeployMent 中的 + 号可能是在右侧，点击之后会弹出对话框选择文件夹，选择 eclipse 默认的 webMoudle 文件夹即可 (MyEclipse 就是 WebRoot 文件夹)，这个一定不能选错，不要选成项目的主文件夹，否则运行就不成功了，再然后 ctrl+R 运行即可。以 debug 方式运行就是 ctrl+D。

使用 IntelliJ 对 Eclipse 项目的支持导入项目

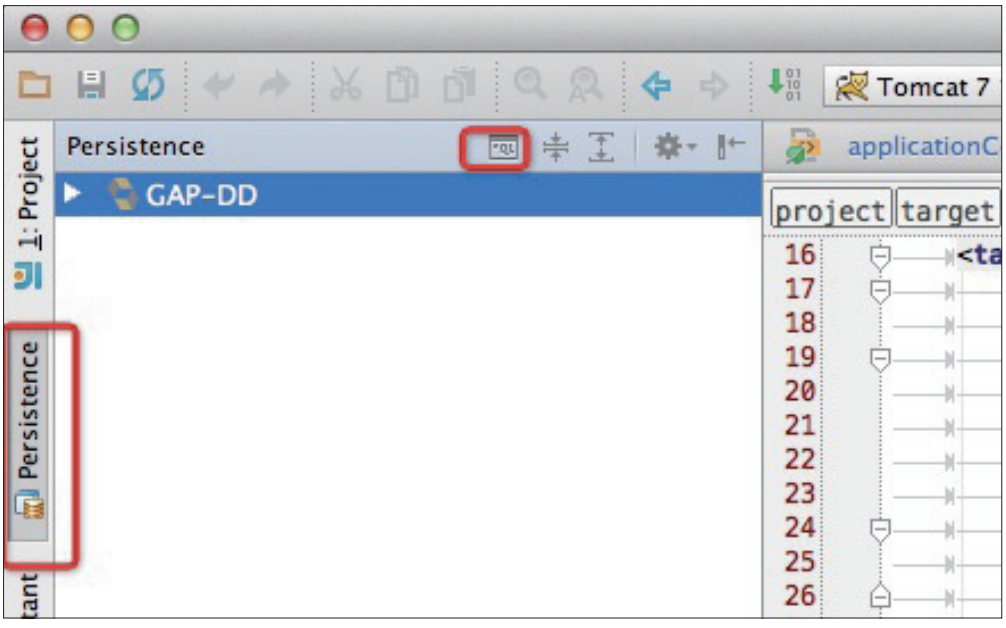
基本参照上个基本都类似，就是步骤比上个步骤要少点，如果出现问题参照上个章节即可。

HQL 查询支持

以前 Eclipse 有个 HQL 的插件可以直接输入 HQL 语法，查询测试结果是否正确，在 IntelliJ 中你也可以，不过不需要任何插件默认提供支持的，如果在导入项目的时候没有自动添加 Hibernate 的支持的话，那需要添加一下 Hibernate 的支持，鼠标选中项目，点击右键如下图：



点击选择 Hibernate 即可，如果没有 Hibernate 应该是项目已经自动添加了 Hibernate 的支持了。然后完成以后就可以如下图操作即可：



选中上面 HQL 图标，在出来的控制台输入你想输入的 HQL 即可了，一样可以自动完成的哦。

注意事项

1. IntelliJ 对于文件是默认随时保存的，基本不需要你 ⌘+s(这个键位默认是保存所有)，所以需要注意不要随手不小心把某个文件改了下，然后直接关闭标签了，其实你的无意的操作已经导致修改了文件，可能导致你在项目运行的时候发现一个很离奇的错误，我暂时没发现可以关闭自动保存的设置，如果你对此非有强迫症的话，vim 的插件可能会帮助到你。

2. IntelliJ 中的文件是实时和本地文件同步的，所以 Eclipse 的刷新功能就不要问了，因为在这根本不需要那个功能。有的编辑器会提醒你是否和本地文件更新，这里也是默认不给任何提醒的，只要你修改了，这里就会立即自动同步更新文件。

3. 在主菜单 File 下面，有个奇怪的 Power Save Mode，这里也说明下，这个顾名思义是省电模式。当你勾选此模式以后，IntelliJ 不会给你完成任何自动完成的功能，例如本来你输入一个字符会自动提示的，勾选以后就没有了，如果出现此问题的话可以考虑是不是自己手贱点过这个按钮噢，这个还是比较适合开会的时候无聊看代码用的，又省电又不影响你看代码。

4. Eclipse 中有个 F2 的功能，可以查看到 Java 文件定义，你在 Spring 的配置文件中可能需要配置这个值，但是在 IntelliJ 中这个是不需要的，因为对于 Spring 这样的支持很是完善，你只需要在 class 的属性中输入 class 的名字然后 ctrl+ 空格 (Basic 这个在 code 的 completion 下面) 自动就完成了全路径，在智能的条件下可能就不需要蹩脚的实现。

5. 项目刚开始导入或者清除 cache 重新打开的时候，下面的状态栏部分会显示 Indexing，这个时候你就等着就可以了，基本你想做的事情什么都不能做，别在这个时候着急点来点去，不过它索引的速度挺快的，别着急。

6. Eclipse 有个 Save Actions 的功能，主要是针对 Java 可以自动格式化以及优化 import 等功能，这个在 IntelliJ 上其实默认是没有的，想在 IntelliJ 中使用这功能可以使用以下两个思路：

    - 利用**宏录制**，宏在开始以后你就格式化下代码，然后优化下 import 等操作，然后结束把宏定义一个 ⌘+s，不过问题就是

不管什么文件都会执行这个宏，不仅仅是 java 文件，我就是利用这个宏主要完成把 tab 转为空格。

- 使用 **Eclipse Code Formatter** 的插件，这个插件基本和 Eclipse 默认的功能类似，但是也有少许区别，用户自己思量是否使用即可。

7. 在 Mac 下 IntelliJ 代码区域是支持手势放大的，这个在给别人演示的时候很是方便。但是 Run、Debug 等视图的时候默认能使想到的就是手动的拖动大小，很是不便。你可以尝试下 ⌘+shift+up/down (Stretch to Top / Bottom)，当然其他的 project 视图等的也都可以利用这个快捷键配置上下左右的方向键来配置使用，方便的很，强烈推荐使用。

最后

第一次发文，如果有什么问题欢迎留言交流，原始文档放在 github 上，各位如果也有一些技巧不妨共同完善下本文档。只需要 Fock 然后 Pull Request 就可以了，如果你觉得有个 github 帐号过于麻烦的话，那我个人觉得你还是转行比较合适。如果不想更新只是想订阅持续更新的内容，只需要 Watch 该项目即可。还可以对项目 Star，这样你就可以在自己的帐号页面快速找到本项目。

附录

Win 快捷键 Win Keymap

Mac OS 快捷键 Mac OS Keymap

华尔街，第一张多米诺骨牌

1987 年初的一天，纳斯达克交易所的一位工作人员出现在世贸中心的电梯间。我们就叫他琼斯吧。他找到合适的电梯，按下按钮。他例行拜访的人是纳斯达克快速增长的客户群中的一位。琼斯知道接下来会见到什么人，华尔街从事股权交易的人都大同小异，一群拥有常春藤名校教育背景 、渴望利润的白人男性。没什么意思。

作者 /Christopher Steiner

Christopher Steiner 是一个为客户提供高折扣的网络杂货店、创业公司 Aisle50 的创始人和 CEO。曾任《福布斯》和 Chicago Tribune 科技版块的记者。

他穿过走廊，来到办公区门前，预感到里面的躁动和兴奋，定了定，打起了精神。交易所和电视上演的抓钱游戏节目并无二致，一个人被塞进玻璃箱里，大把的钱从天而降 。不过有一点还不一样，交易所漫天飞舞的交易单中有一些会赔钱。而敏捷熟练的交易员能够快速交易，并分辨出哪些交易会赚钱，哪些交易会赔钱。

一位接待员招呼了琼斯，然后退到另一个房间去请出主人。她回来的时候，同行的是一位矮个头、满头银丝、衣着整洁的男人。他叫托马斯·彼得菲，一双蓝色的眼睛对琼斯的到来表示欢迎，说话带着口音。

琼斯不会想到，彼得菲后来会成为身家超过 50 亿的富翁，美国最有钱的人之一。他那时还只不过是华尔街的暴发户。但他的交易量却不断上升，利润也不断扩增。琼斯一直很好奇彼得菲这样的人为什么能从市场中持续稳定获利。他雇佣了最聪明的人吗？他有杰出的研究机构吗？他是在冒大险、行大运吗？

琼斯不知道的是，彼得菲根本就不是一名交易商。他是一名计算机程序员。他做交易靠的不是观察交易厅里人们的表情，估测市场的动向或经济趋势怎样影响股票走势。他写代码。他用计算机语言（比如 Fortran、C 语言、Lisp）写出了成千上万条代码，它们构建的算法成就了他的交易所，虽然规模不大，但已是华尔街上最优秀的。他已是华尔街新生代的领袖。

彼得菲带琼斯走进交易厅，琼斯迷惑不已。他看见得越多，就越困惑。其实根本没什么可看的。他预想的是骚动的人群，吵闹的电话声，打印机作业的声音，交易员向纳斯达克交易终端输入交易指令时此起彼伏的叫买叫卖声。可这场景并未出现在他眼前。实际上，他只看见一台纳斯达克交易终端。他清楚彼得菲的交易

量有多大。可是这怎么可能呢？谁是做交易的人呢？

“其他交易场所在哪儿呢？”琼斯问道。“你的交易员呢？”

“那就是，全都在这儿呢。”彼得菲指向房间里唯一的纳斯达克交易终端旁边的 IBM 电脑。“我们的交易都是用它完成的。”交易终端和 IBM 电脑之间有一簇簇的连接线，电脑里装有指示交易品种、交易时间和交易数量的代码。这位纳斯达克职员没有想到，他已经见到了世界上第一台全自动算法交易系统。彼得菲的设备可不只是像过去的交易系统那样提示交易品种，也不仅是简单地弹出需要人来执行的交易单。这台电脑悄悄潜入纳斯达克交易终端，全权决定并执行交易。不需要人的参与。虽然它的对手都是人，都败得一塌涂地。

从纳斯达克终端窃听到的交易数据不断涌来，彼得菲的代码可以利用这些数据分析市场，轻易通过买家出价和卖家售价的价差开出买单和卖单。那时候纳斯达克交易市场价差每股可达 25 美分，那么进行一对 1000 股的交易（比如在 $19.75 价位开出买单，在 $20.00 价位开出卖单），就可以无风险获利 250 美元。

对彼得菲来说，由于他利用机器执行交易，风险成本可以更低。那时候交易员频繁开出挂单，他们面临的最大风险就是在一波市场震荡后，仍然将原来的挂单高高挂起。大多数做市商对市场的反应速度只能和交易者一样，交易者必须不断从电脑屏幕读取新的报价，研读报价信息，重新制定交易计划，撤销老订单，用纳斯达克终端键盘输入新的报价。交易员要是多吃了几口金枪鱼三明治，或是和同事开了会儿玩笑，就有可能被交易市场甩在身后了。彼得菲的电脑可不需要用午餐，它能紧紧咬住交易市场的波动起伏，大大降低风险，这可是人做不到的事情。

彼得菲的交易机构开启了华尔街的新篇章。电脑程序员、工程师和数学家开始了对金融市场长达 20 年的大举进攻，他们的利器就是算法和自动化交易，算法有时无比复杂精密，几乎智能化，可以取代人成为金融市场的决定性力量。

琼斯惊得目瞪口呆。彼得菲把这一切看做是创新的交易方式，琼斯却认为他用临时配置的终端作弊，违反了交易规则。

”你不能这么做，”琼斯说道。

纳斯达克没有交易场所，所有交易都是通过电话或是电脑完成的。电脑网络接收从独立的纳斯达克交易终端的键盘上传来的交易指令。彼得菲整合了应该要连接到交易终端的数据线，将它连接到他的程序员和物理学家团队从零构建的嵌入 IBM 个人电脑主板的一块电路板上。IBM 电脑运行彼得菲自己编写的软件程序。电脑从纳斯达克数据线获取信息，利用算法分析市场，迅速做出交易决定，然后将交易单通过一簇簇连接线传回纳斯达克终端。此前无人知晓的彼得菲已然入侵了纳斯达克。

纳斯达克不会让这奇妙精巧的设计、这个疯子科学家的实验室被市场参与者知晓。其他交易者要是知道了他们是在和 IBM 电脑所运行的算法斗智，而不是和市场上其它凭 直觉下注的赌博者博弈，会作何感想？纳斯达克不想知道答案。

“交易终端和 IBM 电脑的连接必须切断，你得像其他交易者那样，通过键盘一条一条输入交易指令。”琼斯说道。

琼斯离开了。彼得菲站在办公室，想到这也许会是自己事业的终结。纳斯达克给了他一周的时间使其交易符合监督员的 要求。一

想到要拆掉自己的交易机器，他就痛苦不已。他对招聘交易员整天坐在电脑前输入交易指令这想法可没有丝毫兴趣，哪怕是年轻又廉价的交易员。他花了好几年才让自己的交易机构摆脱了人做交易的缺陷，摆脱了人的反复无常。人难免错误频出，懒惰开小差，关键是输入交易指令有延迟。重新使用人力，就很难指望和机器自动化交易获得一样的成效。交易机构的高效会在一夜之间丧失。必须有一个更好的解决方案。

晚上他回到上东区的住宅，睡觉之前，一个方法浮现在脑海。这个方案行之不易，但起码提供了解决问题的可能性。彼得菲想，即使不碰交易终端，他也能从中捕获信息。不用拼接连接线，不用嵌入电路板，什么都不用。但怎样实现呢？他询问了自己的工程师和物理学家，是否能够造出某种设备从屏幕直接读取信息，就像一台照相机那样，然后将这些信息转化成电子字符，传送到等候指令的 IBM 电脑里。答案是能。

但是解决数据来源还只是开了个头，没有一组人坐在纳斯达克终端前，彼得菲要怎样完成交易呢？他不能像之前那样把一根传输线连接在纳斯达克终端上，不行啊，纳斯达克明确规定了交易指令必须得通过键盘输入。彼得菲灵光一闪，有了一个疯狂的念头。但是这真能实现吗？

在接下来的疯狂的一周，彼得菲和他最好的工程师忙着焊接金属，编写代码，焊接数据线。他们在纳斯达克终端屏幕前安了一个大型菲涅耳透镜，放大屏幕字体，又在离透镜一英尺远的地方安装了一部相机。照相机牵出一根数据线连接到旁边的一台电脑上。彼得菲和他的程序员仅用了几天时间就编写了解码照相机传来的可视数据的程序软件。那些数据从特定程序软件流到彼得菲已经做好的算法里，曾经这些算法是通过数据线和纳斯达克终端直接相连的。

现在 IBM 电脑有了一根新的连接线，它不是接入到纳斯达克终端机箱，而是连接到了悬在终端键盘上空的那一堆密密麻麻的金属棒、金属塞和手柄上。如果说照相机和读屏设备略显奇怪，那么系统的这一部分就是怪异得超乎寻常了。它让人想起工业大革命之初复杂精细的机械设备。这个装置是一个从零组装的自动化打字机。手柄断断续续敲打着键盘，执行从电脑传来的交易指令，不到 30 秒就有几十个交易单输入终端。

纳斯达克说了，交易单必须得输入终端，可也没有规定谁来完成输入。彼得菲的团队花了六天时间创造了一种交易和指令输入的半机器人。表面上看，他遵守了法规；但却实实在在地违反了法规的本质。彼得菲可不在意。华尔街不就是一个法律擦边球、变通方案，和秘密交易的世界吗，总是青睐那些最有创造力的骗子。

纳斯达克监督员一周后如约而至。彼得菲在电梯间见到他，带他穿过走廊，来到交易厅。门哐当一响，一周以前鸦雀无声的交易厅，现在是一派喧闹繁忙，这才是交易厅该有的样子嘛。彼得菲带来人穿过大门，骄傲地指向自己的创作。这位纳斯达克职员仿佛置身于儒勒·凡尔纳笔下的科幻小说所描述的场景。

“这是什么？”琼斯问道。

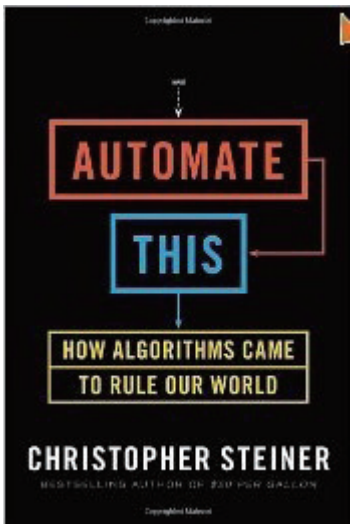
彼得菲解释道他的交易机器如纳斯达克要求的那样运作——键盘输入，一次一单。就在这时候，交易市场活跃起来了，机器也跟着忙碌起来。彼得菲的程序交易速度如此之快，输入设备就像一把全自动机关枪不停开火。交易单不断涌入，噼里啪啦打在键盘上，噪音如此之大，谈话声都听不到了。机器每次停下来，仿佛要安静下来之后，转瞬之间再次启动，比上次更加气势汹汹地弹出更多单子。这整个交易所是华尔街的聪明人 对规则的又一次令人叹为观止的绕行。

“他可不太喜欢这机器啊，”彼得菲回想道。

彼得菲感觉这会儿也没什么大不了的了，他提出在这奇异装置前弄个洋娃娃模型，让她来敲键盘。这是个玩笑，不过彼得菲也愿意这样做。但是琼斯一直紧绷着脸。琼斯摇摇头，彼得菲扮了个鬼脸。他造出了世界上运行速度最快的交易机器，这交易机器可能会面临的被拆卸的命运，他也预料到了。这位纳斯达克职员木了几分钟，然后一言不发地走出了彼得菲的办公室。彼得菲做了最坏的打算：一张纳斯达克传来的禁止他发明的科技参与交易的禁令。但是琼斯没有再回来，彼得菲担心的那个电话也没有打来。他的交易所完好无损。彼得菲几年前不到 10 万美金起家，1987 年赚到了 2500 万美金。

1987 年，彼得菲在华尔街还只是个无足轻重的小人物，但他却是新生代交易商的领袖之一。他们擅长编写复杂的代码、焊接半导体芯片，并运用数学知识探索迷宫般的市场。彼得菲所做的事情理论上简单易懂，操作起来却复杂繁琐：他吸取了最睿智的交易员的智慧，用一系列的算法表达他们的思想。彼得菲的程序包含了一个高明的交易员在做交易决策时所要考虑到的全部因素。不同之处在于电脑运行算法、查阅价格、执行交易所用的时间远远少于人来操作。

使用软件、代码，和敏捷的电脑来击败市场的人不止彼得菲一个。但他的发明，不管是敲打键盘的金属塞还是窃取数据的传输线，都引发了一场革命。今天，60% 的交易由电脑在无人实时监管或很少监管的情况下自动执行。彼得菲在交易界演绎的故事是独一无二的。他不是有先见之明雇佣程序员来扩充自己统治领域的金融大亨，也不是自学编程以期在金融市场获得优势的华尔街玩家。他与众不同的地方在于他是个程序员，一个优秀的程序员。早在

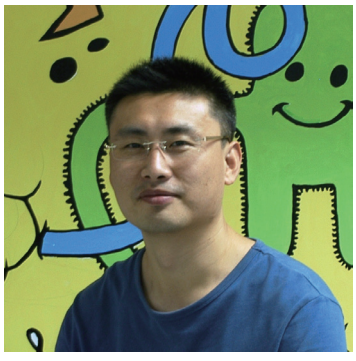


算法渗透了包括金融业的很多社会领域，本书作者将带我们领略这一美丽的算法新世界。算法软件可以让我们的生活更轻松，但如果完全把我们生活外包给计算机，就有可能产生不良的、甚至可怕的后果。我们让算法控制医院、交通、食品供给，究竟能达到什么样的效用？如果我们继续让算法取代人类本能的决策，会产生什么样的后果？本文摘自《算法改变世界》的第 1 章。

他理解股票期权的原理，理解为何不同公司的股票会走势一致之前，他就是个程序员。

华尔街交易所，在一个对彼得菲而言陌生的领域，他用自己的编程技能、数学知识，和编写复杂代码的能力构建出的分层算法破坏了原有的交易秩序和规则。在过去的 15 年间，这种破坏性黑客范式风行于世界各地：一个聪明的电脑代码和算法工程师对某一新的领域产生了兴趣，培养自身在该领域的专长，然后应用计算机科学让代码片段模仿人类先行者的操作。这些代码战胜了无数公司，破坏了行业标准，击败了行业老旧势力，从而颠覆了整个行业。构建算法模仿，超越，并最终取代人类的能力是 21 世纪最重要的能力。随着具有这种能力的人数量激增，工作会越来越少以致最终消失，生活会发生巨变，各行各业都会重建规则。这已经发生了，这种趋势还会延续。这一趋势也如历史上的其它趋势一样追逐利益。这也就是为什么它始于华尔街，很大程度上这要归功于这位匈牙利移民。■

一本畅销书的诞生



作者 / 吴海星

2001 年毕业于南京理工大学，现任北京北控文化体育有限公司技术总监。熟悉 Java 及 Scala 语言，熟悉 web 应用系统开发，熟悉 IC 卡应用系统建设，目前主要对商业智能方面的内容感兴趣。《量化：大数据时代的企业管理》译者。

图灵 ID: 海兴

声　明　本文是《精益创业实战》的读书笔记，虽然以怎么做一本畅销书为题，而且作者 Ash Maurya 在书中所讲的过程和思想可能会对出版一本畅销书有帮助，但是否可行，我不得而知。

畅销书是怎么写出来的?

书是一种产品，制作过程简单，生产历史悠久，成品种类繁多。但一本书会不会畅销，完全是一个谜。

哈利波特

1990 年新年前夜，只有 25 岁的罗琳失去了 45 岁的母亲，她说，母亲的去世在她的心里留下了一片空白，也给了她创作从小就是孤儿的波特这个人物的灵感。

1997 年 6 月，一家小型的中立出版社出版了哈利波特系列的第一部:《哈利 · 波特与魔法石》。随后几个月，这本书大受好评。该系列前三集都获得了 9-11 岁年龄组“雀巢聪明豆儿童图书奖”。

君主论

尼可罗 · 马基亚维利(1469 年 5 月 3 日 ~1527 年 6 月 21 日)，是意大利著名政治家，他的著作《尼可罗 · 马基亚维利君主论》影响了后世许多政治家，他的理论也被曲解为马基亚维利主义，即为达到目的不择手段，强权至上主义。希特勒、墨索里尼都信仰马基亚维利主义。

马基维亚利写作《君主论》时一贫如洗，隐居乡间，据他给朋友的一封信中描述: “傍晚时分，我回到家中的书桌旁，在门口我脱掉沾满灰土的农民的衣服，换上我贵族的宫廷服，我又回到古老的宫廷，遇见过去见过的人们，他们热情地欢迎我，为我提供单人的食物，我和他们交谈，询问他们每次行动的理由，他们宽厚地回答我。在这四个钟头内，我没有感到疲倦，忘掉所有的烦恼，贫穷没有使我沮丧，死亡也没能使我恐惧，我和所有这些大人物在一起。因为但丁曾经说过: 从学习产生的知识将永存，而其他的事不会有结果。

揭开谜底

这两本书差异巨大，但也有共同点。这两本书都很成功，但是，虽然作者觉得自己的作品会受欢迎，但是不是真的能如己所愿，心里也没底。他们甚至可能都没想过自己写的书是否会畅销，只管把自己关起来，把想法落到纸面上。

它们跟世面上大多数产品的诞生过程没什么两样，事先没什么设计，后期主要靠运气(尽管作者的思想创意是决定性因素，但只是一个未经证实的决定性因素)。

怎么做一本畅销书

Running lean 的作者 Ash Maurya 很精明。虽然他写的书肯定不如上面那两本畅销，但他肯定比那两位精明。

他是一位创业者，很忙，本来也没想过要写一本书。Ash 最初只是在发一些博客，跟人探讨精益创业的想法和问题。有读者建议他把博客编纂成书，所以，他决定，再等等。

因为他深知 **时间**是最稀缺的资源，应该在每单位时间内尽最大可能学习(与客户有关的一切)。用更小、更快的迭代测试愿景。*

后来提出这个要求的人越来越多，他才决定用精益创业的方式来做个尝试。下面就是他做 Running lean 的过程。

值得解决的问题

他先花了一天时间 **搭了个 Demo**。把这本书的标题、目录和封面都放到了一个网页上。然后把怂恿他写书的家伙都叫来，问他们，“如果我写这么一本书，你们会不会买？”

这个问题主要是针对目录，因为对一本书而言，**风险最大的部分**就是对目录的细化，填上内容。甚至价格都没关系，他相信公道自在人心。目录确定了，**解决方案就定下来了**。

尽管他得到了积极的反馈，但他知道，只有那几个人的肯定，并不足以说明这是个 **值得解决的问题**，所以他决定留着那个网页，告诉人们他准备开始写这本书，让博客上那些读者帮他传播这一消息(**渠道测试**)，等到他收到了 1000 封 email(**潜在客户**)的时候，

他觉得写这本书对于他而言是个 **值得解决的问题**，最起码他能把成本赚回来。

《精益创业实战》的中文译者张玳先生正在用这个过程编写《宫本茂传说》一书，所以你可以亲自见证这个过程的再次应用。他在第一篇文章中要了社区的 50 个推荐。坦白地说，我推荐的时候并没想着将来要买一本 :D。

定性检验

写一整本书是个工程浩大的工作，Ash 决定先从自己的博客上扒拉一些东西下来弄成第一章。但他还是觉得内容太多，做出来的不是那种 可以从客户那里得到反馈的东西（**最低限度的产品**）。

所以他把目录做成了幻灯片，要免费开讲。最开始有 30 个人报名听课，但他只找到了能容纳 10 个人的教室。这是上天的安排，让他可以再讲两次课（**小批量迭代**）。

在最初的成功之后，他不仅开了更多次课，还开始收费了（**收钱是检验产品的第一标准**）。每次课程中，他都会对幻灯片的内容进行微调，效果越来越好，价格也水涨船高。

经过了一个夏天，他觉得从客户那里得到的反馈已经足够多了，

他对解决方案已经有了充分的认识，所以决定开始写书。他联系了那些看过页面的人，告诉已经不耐烦的他们，这本书会像软件一样，以迭代的方式编写和发布。如果他们肯预定的话，就能每两周收到两章 PDF 格式的内容。

有多一半人同意了，他们就是早期用户，只关心内容，不关心形式。所以内容仍然是产品中风险最大的部分，得测!

在这每两周的迭代周期得到的客户反馈弥足珍贵，有很多优化改善工作就是在这以阶段完成的。采用这种办法，这本书写得又快又好。

定量证实

直到书的内容全部完成，Ash 才找人设计了封面，检查子标题，研究印刷 /ebook 之类的问题，并且搭了个营销网站(在正确的时间做正确的事)。

跟做产品一样，吸引外部资源的理想时机是在产品 / 市场相匹配之后。并且在那之后，这本书还在根据读者的访谈及课程中的反馈不断调整(**跟客户之间建立起持续的反馈回路**)。

结束了吗?

一本书，就像软件一样，从来不会结束，只会发布。

在这本书之外，Ash 还

- 在他的博客上继续分享他学到的东西

- 有相关的专栏写作
- 课程还在继续

创业跟上面提到的过程一样，关键是要找到值得解决的问题。因此学会使用“精益画板”和“用户周期”。

精益画板是一个业务模型验证工具。可以帮助创业者记录业务模型，评估进度，跟内部及外部的利益相关者沟通学习情况。

用户周期是一种客户生命周期管理软件，帮公司把用户转化成推荐型用户。推荐型用户是回头客，还是口碑传递者，是给你钱的人。■

八 ++

职场有影帝出没，屌丝们请当心！

引子

职场有影帝出没，请当心！广大屌丝请注意危险，谨慎前往。

人生苦短，必须感性；职场如戏，要靠演技。不少公司正变成秀场，影帝层出不穷，屌丝们的辛苦努力一不小心就成了影帝的嫁衣。影帝在人前风光灿烂，而在秀场中吃盒饭的屌丝们却只有回家躺在床上翻看《论演员的自我修养》。

作者 / 张林

一个软件研发的老兵，享实战、乐思考、爱分享，同时还是位有梦想的吃货。着迷于帮助和领导软件研发取得成功，从而接触了敏捷精益。虽然敏捷标签还在，已经开始少谈敏捷、精益等时髦概念，回归软件、人和系统这些更加本质的东西。愿意为国内的软件研发社区奉献一己之力，社区组织者，多个大会分享者，博客微博共用名 @ 大卫张 33。

正文

拿着卖白菜的钱，操着卖白粉的心，一副永远没睡醒的邋遢样，难道这就是传说中的屌丝。拿着卖白粉的钱，操着卖白菜的心，一副永远光鲜亮丽的模样，难道这就是传说中的影帝。屌丝和影帝同在职场，他们又有哪些不得不说的那些故事呢?

第一节：生为屌丝我悲剧

生而不幸，没遇上李姓老爸，从此成为屌丝，悲剧了。虽然我们也见过不少屌丝励志奋斗并成为高帅富的例子，然而大多数人的命运依然是生是屌丝终身是屌丝。这种命运不仅是因为矮挫穷，还在于无法改变屌丝心态。

屌丝心态之首是惧怕冲突。身为屌丝，或者自惭形秽不敢与人冲突，或者装的云淡风轻避免与人冲突，甚至故作清高不屑冲突。这些归根到底都是不理解冲突的正向作用，也没有足够的冲突管理技能进行把控。惧怕冲突的屌丝在面对外面的世界时总是无力引导事务的发展方向，自身的意见难以表达，总有被强奸的感觉，所以总是在抱怨。

屌丝心态之二是明君幻想。这种幻想美好的地方在于，假设真的存在所谓的明君，屌丝什么都不用做就能得到救赎。可理想很丰满，现实很骨感。真正的现实往往是“我的意中人是个盖世英雄，有一天他会踩着七色的云彩来娶我，我猜中了前头，可是我猜不着这结局 ……”。

屌丝心态之三是被害者心态。因为心中的盖世英雄一直没有出现，也因为屌丝心中的冲突一直得不到解决，为了保护自己，屌丝只

好把自己封闭起来。这世界太冰冷，太残酷，太不友好，屌丝在碰到任何新兴事物时总有很强的怀疑态度和叛逆精神，在很多场合下以受害者和打酱油的状态出现。

屌丝心态之四是唯一角度。世界上只有一种正确的做事方式，就是屌丝自己的做事方式。面对其他的任何方式，屌丝都会从怀疑的眼光开始，甚至会先下否定的结论，再试图为否定的意见找到佐证。症状严重的屌丝甚至可能把他身边的大部分人都看做影帝，因为他只能看到自己的努力，却否定了所有其他人的努力。

据说混 IT 圈的屌丝特别多，也据说 IT 圈怀才不遇的人特多，看来这一切并非巧合。

第二节：人生何处无影帝

人生何处无影帝。走捷径是人的天性，每个人都希望付出更少的努力，同时得到更多。这种本能在缺乏约束的情况下会不断放大。在屌丝眼中，不用付出太多努力，就能风光灿烂的职业不少，例如公务员，例如管理者，例如……。更可怕的是，屌丝身边也有这样的人，屌丝需要天天和他们打交道，看着那些指个手画个脚就能舒舒服服地侵占屌丝大部分功劳的人，屌丝们很不平衡。

当影帝好啊。他可以画饼，但不付出；他可以要求，却不承诺；他可以承诺，可不兑现；他可以看到问题，当然不承担；他还会争功，必不落人后。

屌丝的心声，我有一个愿望，影帝不干，我也不干。可惜屌丝没演技，一旦不干，就像漆黑里的萤火虫那样耀眼。屌丝吃亏后发誓说，我有一个梦想，我要当影帝。

第三节：从屌丝到影帝有多远?

屌丝可以生而为之，但影帝可不是那么好当的。这个世界最远的距离不是我和你的距离，而是我们在一起，我是屌丝，你是影帝。

首先，影帝当然需要彪悍的演技。影帝需要多重角色扮演，有时是大爷，有时装孙子，时而软磨硬泡，时而大义凌然，需练就一番精彩演技，还需在各种场景下快速变脸。影帝还需要很好的卖相，以便快速获取信任。

其次，影帝需要催眠自己。影帝要坚信自己做的一切都是对的，甚至为了自己的行为找到各种高尚的理由。一旦不能面对，心防被破，就容易走火入魔，功力大退。心定则意定，意定则慧生，心安理得是影帝必备的状态。

再次，影帝需要能言善辩。面对各种类型的人，各种工作角色，需要针对这些方方面面组织起有效的语言，才有可能在其中左右逢源。谎言需要更多的谎言来弥补，影帝需要聪明的头脑和强大的语言组织，从而能够逢凶化吉、化险为夷。

再次，影帝还需巧妙地抬高自己并贬低别人。刚入门的影帝往往会用谎言直接打击对手，效果直接，缺点是容易被戳穿。中等影帝会巧妙地利用语言技巧，描述对别人不利的事实，隐藏对自己不利的信息，从而让人高看自己低看别人。而资深影帝什么也不用干，往那里一站，就有那个气场，所有功劳归于影帝。

最后，影帝必须要有一颗足够坚强的心。谎言需要更多的谎言来弥补，再美的谎言也有被戳破的时候，影帝需要有一颗足够坚强的心来面对这种打击，才不会丧失继续当影帝的自信，常常可以挺住并渡过难关，甚至可以换个地方，凭借演技东山再起。

第四节：当屌丝遇上影帝

这个世界最远的距离不是成功和失败的距离，而是我站在你面前，你却分不出我是屌丝还是影帝。在我从业的第一个软件研发公司中，我就遇到了一位了不得的影帝。他是博士，能言善辩，阳光帅气，让人如沐春风，给我留下了很好的第一印象，不仅让我相信公司前程远大，而且还给我指点了技术方向。在这高大的印象中，我原谅了他在小事上一次次的爽约，在公司迟迟没有营收和多个项目失败后依然认为公司前程远大。他更厉害的是，还泡到了老板的女儿。直到我们做的项目失败，直到数次失败后，我才承认了现实。为什么我那么晚才承认事实呢？多年以后，我才知道光晕效应这个专业名词。从这里我学到了一个非常重要的道理，那些让你难受的不一定是影帝，而让你开心的未必不是。

如果你把每个人都当做影帝，你就完全没法工作了，终日生活在怀疑中。如果你认为每个人都不是影帝，你就悲剧了，时常受到伤害。关于应对，下面是些小小建议。

应对 1：克服自己的屌丝心态，我是屌丝我何惧。要记住，是影帝需要屌丝，而屌丝完全不需要影帝。因此，完全不要惧怕与影帝发生冲突，不要把希望太多寄托在清官身上，调整自己的心态，做更加积极的自己。

应对 2：光照不到的地方才会有阴影。为了掩盖自己自卑脆弱的心灵，屌丝往往是清高而自负的，不会去关注与老板、合作者之前的关系，不会把自己的工作变得透明。这往往为影帝提供了可乘之机，影帝擅长发掘和填补这些空白，让自己变成不用干活却非常重要的人。

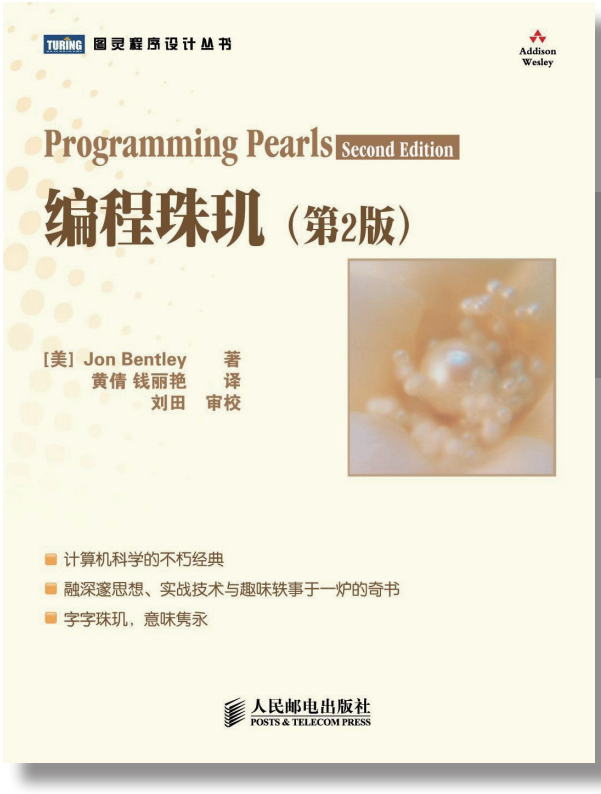
应对 3：“贪官奸，清官要更奸”。屌丝也可以有演技，不管你是想未来成为影帝，还是避免被影帝伤害，适当的演技都必不可少。

应对 4： 要用脚投票，不要动辄用脚投票。碰到影帝，实在搞不定，就用脚投票吧，在上面的例子中我就是这么干的。但是要记住，影帝哪里都有，遇到影帝就搬家，很快你就无处可去了。

后记

来分享你作为屌丝和影帝不得不说的那些故事吧！你曾经遇到过什么样的影帝，故事是如何开场，如何结局，你有什么感受呢？在应对影帝方面，你有什么好方法可以介绍给大家呢？ ■

书　榜



编程珠玑（第 2 版）

作者：Jon Bentley

译者：钱丽艳，黄倩

书号：978-7-115-17928-9

图灵社区推荐：2

本书被认为是历史上最伟大的计算机科学著作之一。它融深邃思想、实战技术与趣味轶事于一炉，带你真正领略计算机科学之美。

本书是计算机科学方面的经典名著。书的内容围绕程序设计人员面对的一系列实际问题展开。作者 Jon Bentley 以其独有的洞察力和创造力，引导读者理解这些问题并学会解决方法，而这些正是程序员实际编程生涯中至关重要的。本书的特色是通过一些精心设计的有趣而又颇具指导意义的程序，对实用程序设计技巧及基本设计原则进行了透彻而睿智的描述，为复杂的编程问题提供了清晰而完备的解决思路。本书对各个层次的程序员都具有很高的阅读价值。

2 大数据：互联网大规模数据挖掘与分布式处理

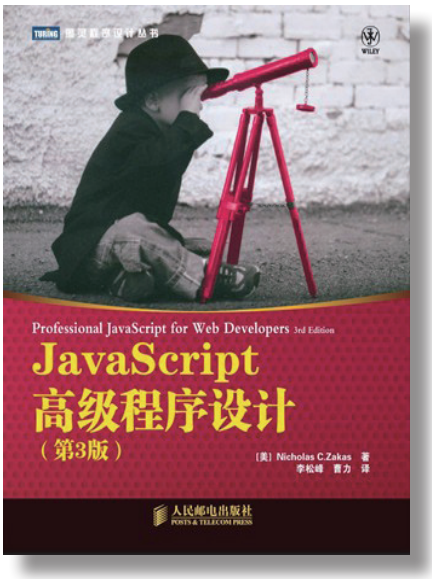
作者：Anand Rajaraman, Jeffrey D. Ullman

译者：王斌

书号：978-7-115-29131-8

图灵社区推荐：21

本书源自作者在斯坦福大学教授多年的“Web 挖掘”课程材料，主要关注大数据环境下数据挖掘的实际算法。书中分析了海量数据集数据挖掘常用的算法，介绍了目前 Web 应用的许多重要话题。读者更可以从网上获取相关拓展材料。



3 JavaScript 高级程序设计（第 3 版）

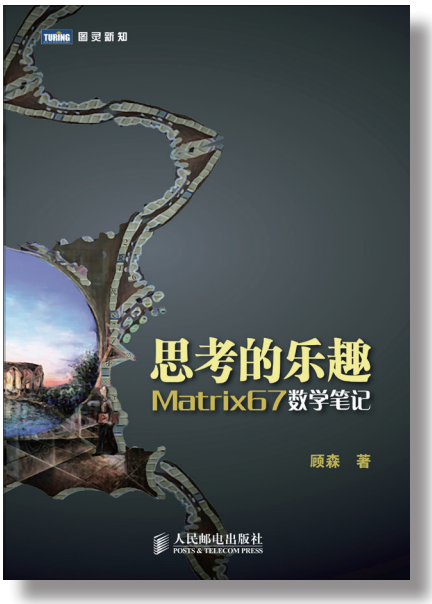
作者：Nicholas C.Zakas

译者：李松峰，曹力

书号：978-7-115-27579-0

图灵社区推荐：27

本书是 JavaScript 超级畅销书的最新版。ECMAScript 5 和 HTML5 在标准之争中双双胜出，使大量专有实现和客户端扩展正式进入规范，同时也为 JavaScript 增添了很多适应未来发展的新特性。



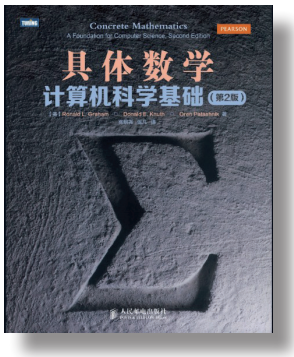
4 思考的乐趣：Matrix67 数学笔记

作者：顾森

书号：978-7-115-27586-8

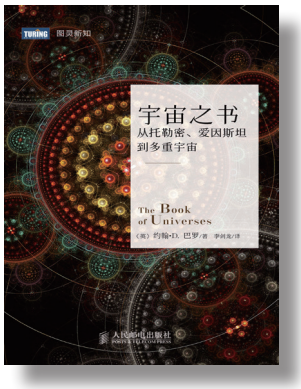
图灵社区推荐：23

本书是一个疯狂数学爱好者的数学笔记，面向所有喜爱数学的读者。本书包括 5 部分内容，即生活中的数学、数学之美、几何的大厦、精妙的证明、思维的尺度，涉及 48 篇精彩的文章。即使你不喜欢数学，也会为本书的精彩所倾倒。本书文章是独立的。一篇文章一个话题，文章与文章之间基本不会做参考，读者可以随意跳着看。



5 具体数学：计算机科学基础（第 2 版）

作者：Ronald L.Graham, Donald E.Knuth, Oren Patashnik
译者：张明尧，张凡
书号：978-7-115-30810-8
图灵社区推荐：29



6 宇宙之书：从托勒密、爱因斯坦到多重宇宙

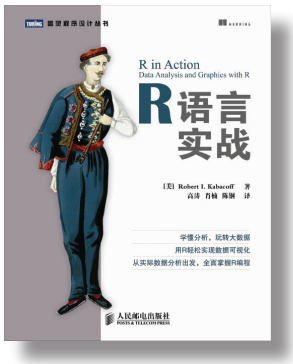
作者：John D. Barrow
译者：李剑龙
书号：978-7-115-30972-3
图灵社区推荐：8

7 腾云：云计算和大数据时代网络技术揭秘

作者：徐立冰
书号：978-7-115-31150-4
图灵社区推荐：8

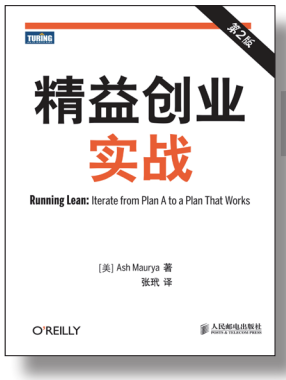
8 iOS 6 编程实战

作者：Rob Napier，Mugunth Kumar
译者：陈晓亮，武海峰，邓强，周庆成
书号：978-7-115-31218-1
图灵社区推荐：15



9 R 语言实战

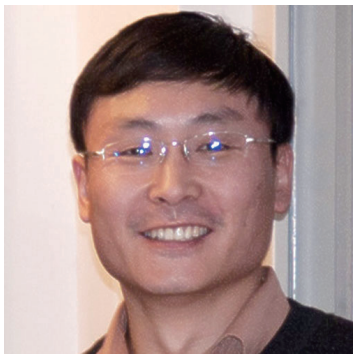
作者：Robert I. Kabacoff
译者：高涛，肖楠，陈钢
书号：978-7-115-29990-1
图灵社区推荐：15



10 精益创业实战（第 2 版）

作者：Ash Maurya
译者：张玳
书号：978-7-115-30540-4
图灵社区推荐：10

D is for Digital 成名手记



作者 / 李松峰

图灵 QA 部主任。2006 年起投身翻译，出版过译著 20 余部，包括《JavaScript 高级程序设计》、《简约至上》等畅销书。2008 年进入出版业，从事技术图书编辑和审稿工作。2012 年下半年创立“A List Apart 中文版”站点（alistapart.cn），旨在向中文读者译介这一国际顶级 Web 设计与开发杂志。他经常参加技术社区活动，曾在各种技术大会上做演讲嘉宾，好评如潮。

2013 年 4 月 12 日，图灵社区发起了“有奖征书名: D is for Digital”活动。截止活动 2013 年 4 月 19 日活动结束，共有 65 位社区会员参与，征得 104 个书名。非常感谢各位会员的鼎力支持和热情参与。

评选结果

根据活动规则，经过图灵编辑两轮投票，评出了前三名 4 个，分别奖励社区银子 200 两、100 两、50 两。获奖书名及会员 ID 如下，恭喜恭喜:

- 第一名: **数字王国漫游指南** @ newiuce
- 第二名: **当谈论计算机时我们在谈些什么** @ liubai
- 第三名: **数字时代——你不可不知的计算机和通信知识** @ Juven
- 第三名: **数字世界——我们生活中的计算机** @高建武

评选过程

本次活动本着公开、公平、公正的原则，由图灵编辑经两轮投票，最终选出前三名。以下是本次活动征得的 104 个备选书名:

```
▼Array[104]
  ▼[0 … 99]
      0: "D时代漫游指南"
      1: "数字王国漫游指南"
      2: "D时代 -- 你不可不知的计算机和通信知识"
      3: "数字时代--你不可不知的计算机和通信知识"
      4: "D大调第一数字协奏曲"
      5: "数码D国"
      6: "D时代字典:从总统到平民，一个见多识广的公民应当知道的有关计算机时代的一切"
      7: "计算机之技"
      8: "D造数字世界的基本准则"
      9: "D造数字世界指南"
      10: "数字的世界"
      11: "数字生活"
      12: "信息时代生存指南"
      13: "数字时代生存指南"
      14: "信息技术漫谈"
      15: "头文字D"
      16: "数字时代普及读本"
      17: "计算机的秘密"
      18: "我们世界中的计算机"
      19: "数字世界--我们生活中的计算机"
      20: "D大调第一数码协奏曲"
      21: "你不知道的世界"
      22: "遨游数字世界"
      23: "畅游数字世界"
      24: "达人眼中的数字世界"
      25: "为数字时代而生"
      26: "D调数据的世界"
      27: "D丝入门手册"
      28: "写给大家看的计算机书"
      29: "计算机世界ING"
      30: "数字之术"
      31: "数码之术"
      32: "数字时代的常识"
      33: "数字之道"
      34: "数字与技术"
      35: "数字与科技"
      36: "信息时代之数字"
      37: "D客帝国"
      38: "大话电脑"
      39: "电脑之道"
```

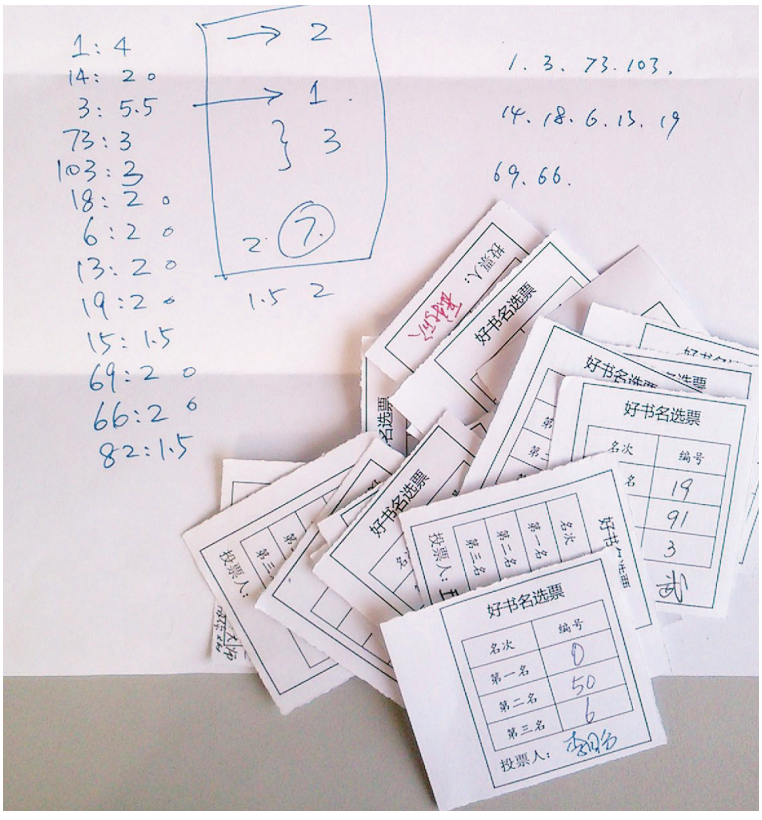
```
40: "D的意志：大数字时代"
41: "四维数码时空"
42: "D手册：数字时代红宝书"
43: "数字时代超级指南"
44: "数字时代权威指南"
45: "数码技术一点通"
46: "数字之谛"
47: "数字化生活之道"
48: "数字世界之道"
49: "数字（Digital）生活术（D）"
50: "D is for Digital：计算机通信时代的生活常识"
51: "解读数字王国：一本带你穿越到数字王国，解读你所知道的计算机"
52: "道 可道"
53: "我是搞IT的：不可不知的计算机和互联网知识"
54: "数字时代常识"
55: "数字时代的秘密"
56: "数字时代"
57: "字节时代"
58: "游戏计算机-快乐不简单"
59: "D大点事儿"
60: "计算机简读"
61: "计算机科学基础知识葵花宝典"
62: "D：为数字而生"
63: "揭秘数字与通信"
64: "揭秘数字与技术"
65: "數位時代：隱匿在我們身邊的必備知識"
66: "计算本源Digital"
67: "计算机本源之Digital"
68: "计算本源之Digital"
69: "数字瘸--信息达人必知必会"
70: "数字地--信息达人必备"
71: "数字D--计算机与通信达人指南"
72: "数字时代-每个人都应该知道的计算机原理"
73: "当谈论计算机时我们在谈些什么"
74: "数字时代漫谈：互联网基础/普及读本"
75: "数字时代漫谈：不可不知的互联网世界"
76: "D's 数据"
77: "D次数据"
78: "D造之数据"
79: "D空飞行-Da时代的领航人"
80: "数码须知"
81: "计算机视界——关于计算机，你需要知道的事"
82: "电脑之道:大师教你的电脑基础"
83: "数字系统札记"
84: "数字系统漫步"
85: "Digital新思维"
86: "漫谈Digital"
87: "Digital：谈谈数字系统方方面面"
88: "Digital：你需要知道的一些常识"
```

```
  89: "数字系统到底是什么"
  90: "计算系统泛论"
  91: "生活中的计算系统"
  92: "计算机里有什么：你应该去了解的计算机与网络"
  93: "从数字说起：你应该知道的计算机和通信知识"
  94: "数字系统之源"
  95: "数字系统基础"
  96: "数字的微观世界"
  97: "计算机的微观世界"
  98: "D大调的计算机常识"
  99: "D大调的计算机交响曲"
▼ [100 … 103]
  100: "数的日常"
  101: "计算机日常"
  102: "典雅数字"
  103: "追根究D：计算机科学基础"
  length: 104
```

第一轮投票选出得 2 票以上的书名 11 个（括号内为得票数，0.5 票是有编辑只选了主书名或副书名）：

1. 数字王国漫游指南(4)
2. 数字时代——你不可不知的计算机和通信知识(5.5)
3. 当谈论计算机时我们在谈些什么(3)
4. 追根究 D: 计算机科学基础 (3)
5. 信息技术漫谈(2)
6. 我们世界中的计算机(2)
7. D 时代字典：从总统到平民，一个见多识广的公民应当知道的有关计算机时代的一切(2)
8. 数字时代生存指南(2)
9. 数字世界 -- 我们生活中的计算机(2)
10. 数字滴 -- 信息达人必知必会(2)
11. 计算本源 Digital (2)

以下是第一轮选票和统计结果**实拍图**:



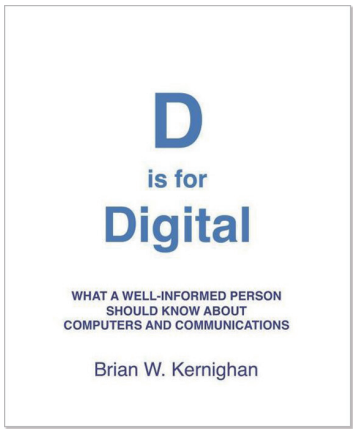
为体现公平，第二轮计票采用了加权计算方法:

- 某编号被选为第一名得 3 票
- 被选为第二名得 2 票
- 被选为第三名得 1 票。

以下是第二轮备选书名得票情况(括号内为加权后的得票数，加粗的书名为最终得奖的书名):

1. **数字王国漫游指南** (30)
2. **数字时代——你不可不知的计算机和通信知识** (17)
3. **当谈论计算机时我们在谈些什么** (18)
4. 追根究 D: 计算机科学基础 (16)
5. 信息技术漫谈 (5)
6. 我们世界中的计算机 (6)
7. D 时代字典：从总统到平民，一个见多识广的公民应当知道的有关计算机时代的一切 (16)
8. 数字时代生存指南 (5)
9. **数字世界——我们生活中的计算机** (17)
10. 数字滴——信息达人必知必会 (6)
11. 计算本源 Digital (6)

以下是第二轮选票**实拍图**:



《世界是数字的》中没有高深莫测的专业术语，但它全面解释了当今计算和通信领域的工作方式，包括硬件、软件、互联网、通信和数据安全，而且讨论了新技术带来的社会、政治和法律问题，让你理解这些问题以及在解决问题时作出的折中。这本书结构紧凑、深入浅出，即便没有技术背景也能由此了解这个数字世界的全貌和真相。

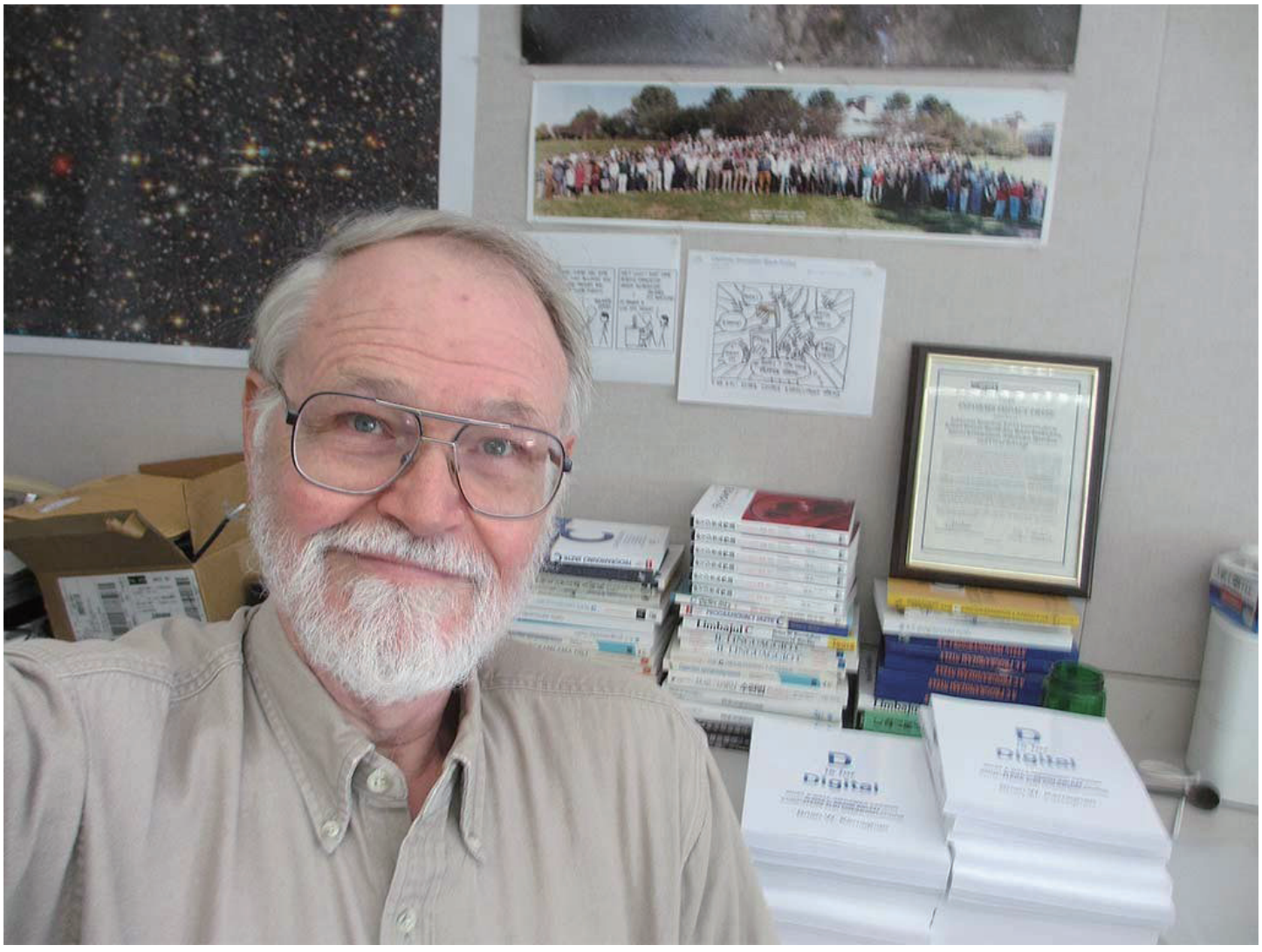
感谢图灵编辑参与投票，感谢社区会员积极参与。Over :-)

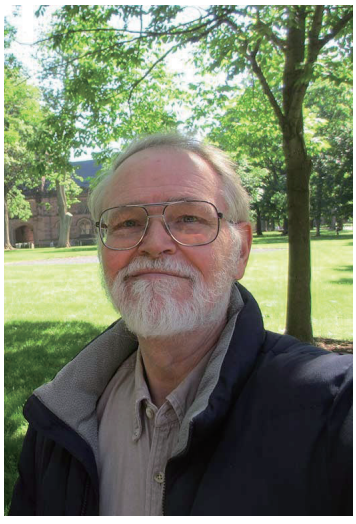
后记：

D is for Digital 最终定名为《世界是数字的》。虽然这本书最后并没有选择任何一个备选作为最终的书名，但是社区上朋友们的创造力和想象力着实让各位编辑们大开了眼界，也为这本书的最终定名提供了无限的灵感。我们热切希望更多的爱书之人加入图灵社区，提勘误、评封面、想书名、吐槽……来自读者的参与就是我们前进的动力，也是我们勇气的来源。■

Brian W. Kernighan：

D is for Digital 中文版序





如今，计算机、通信系统，以及由它们支撑的数字产品已经无所不在了！笔记本电脑、手机和互联网，这些都是显而易见的。更多的则是我们平常看不见的，比如那些栖身于电子设备、汽车、火车、飞机、电力系统、医疗设备中的计算机。偶尔，透过某些提示，你会得知世界上有无数系统正在悄悄收集、分享你的个人信息，这些信息甚至会被用在违背你意愿的地方。

《世界是数字的》简明扼要但又深入全面地解释了计算机和通信系统背后的秘密，旨在让没有技术背景的读者更好地理解自己生活的这个数字世界。这本书解释了如今计算和通信的运作方式，包括硬件、软件、互联网，还有万维网，同时还探讨了新技术引发的社会、政治和法律问题，让你明白现实当中的一些难题和迫不得已的折中。

我相信，了解这些技术常识对于任何人都非常重要，无论你是什么背景。非常高兴这本书由李松峰和徐建刚翻译成中文出版，希望中国读者能够喜欢。■

欢迎加入
图灵社区

最前沿的IT类电子书发售平台

电子出版的时代已经来临。在许多出版界同行还在犹豫彷徨的时候，图灵社区已经采取实际行动拥抱这个出版业巨变。作为国内第一家发售电子图书的IT类出版商，图灵社区目前为读者提供两种DRM-free的阅读体验：在线阅读和PDF。

相比纸质书，电子书具有许多明显的优势。它不仅发布快，更新容易，而且尽可能采用了彩色图片（即使有的书纸质版是黑白印刷的）。读者还可以方便地进行搜索、剪贴、复制和打印。

图灵社区进一步把传统出版流程与电子书出版业务紧密结合，目前已实现作译者网上交稿、编辑网上审稿、按章发布的电子出版模式。这种新的出版模式，我们称之为“敏捷出版”，它可以让读者以较快的速度了解到国外最新技术图书的内容，弥补以往翻译版技术书“出版即过时”的缺憾。同时，敏捷出版使得作、译、编、读的交流更为方便，可以提前消灭书稿中的错误，最大程度地保证图书出版的质量。

现在购买电子书，读者将获赠书款20%的社区银子，可用于兑换纸质样书。

最方便的开放出版平台

图灵社区向读者开放在线写作功能，协助你实现自出版和开源出版的梦想。利用“合集”功能，你就能联合二三好友共同创作一部技术参考书，以免费或收费的形式提供给读者。（收费形式须经过图灵社区立项评审。）这极大地降低了出版的门槛。只要你有写作的意愿，图灵社区就能帮助你实现这个梦想。成熟的书稿，有机会入选出版计划，同时出版纸质书。

图灵社区引进出版的外文图书，都将在立项后马上在社区公布。如果你有意翻译哪本图书，欢迎你来社区申请。只要你通过试译的考验，即可签约成为图灵的译者。当然，要想成功地完成一本书的翻译工作，是需要有坚强的毅力的。

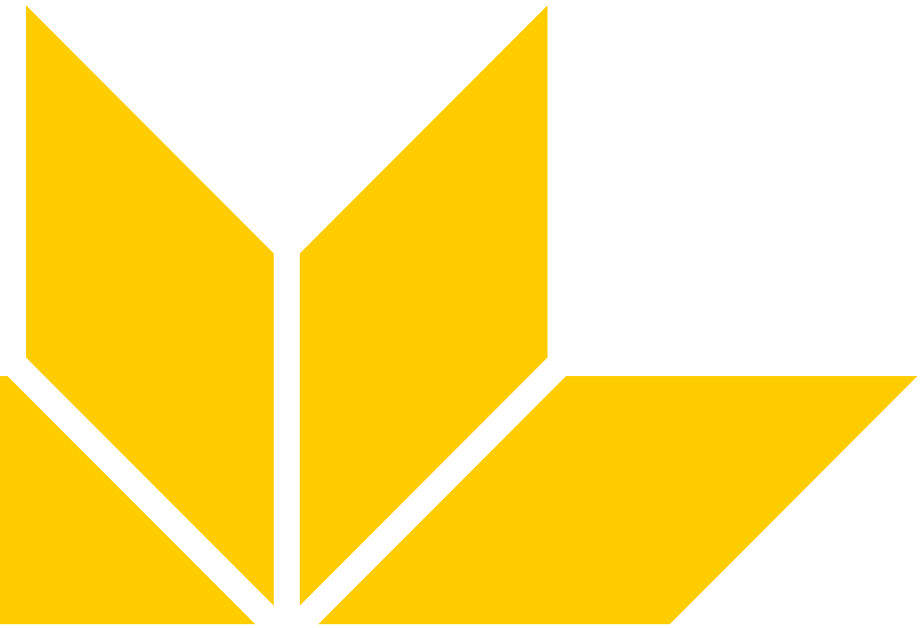
最直接的读者交流平台

在图灵社区，你可以十分方便地写作文章、提交勘误、发表评论，以各种方式与作译者、编辑人员和其他读者进行交流互动。提交勘误还能够获赠社区银子。

你可以积极参与社区经常开展的访谈、乐译、评选等多种活动，赢取积分和银子，积累个人声望。

图灵社区 出品

出版人：武卫东
编辑：李盼
顾问：杨帆、谢工、李松峰
设计：大胖



地址：北京市朝阳区北苑路13号院领地OFFICE C座603室
电话：010-51095181
微博：weibo.com/ituring
Email: ebook@turingbook.com