阶级斗争的网络安全手册：编程随想文选

编者的话

在公开实行法西斯专政白色恐怖的社会帝国主义国家中想要避免遭到修正主义铁拳的打击，就必须学会如何【隐匿自己的身份】。**编程随想**在他的博客撰写了一系列有关的文章，编者将其收录于本文选。由于编程随想是一个反动的【网络自由主义者】，所以这册文选只收录他撰写的技术帖而不是政治帖。

这些文章基本都【年代久远】，有些教程在今天已经失效。所以我们最应当学习的是【隐匿身份】的思路和原则，这在编程随想的文章中清晰可见。至于有关VPN、Tor、Veracrypt、虚拟机、Linux等（最新的）教程，还是建议去谷歌搜索。重申一遍：主要学习编程随想总结的【思路】和【原则】

注：本文选中所有超链接均为编程随想博客上的原链接。

为啥朝廷总抓不到俺——十年反党活动的安全经验汇总

文章目录

好几天没上线，可能有读者以为俺出事了。别担心！俺21日还在回复评论，截止发这篇博文，【未】超出14天的期限，属于【正常静默】。

因为这篇博文要【全面地】分享俺十年反党活动的【技术】经验，牵涉到很多零碎的内容，整理起来多费了点精力和时间。

★“朝廷想抓俺而不可得”正说明了——俺的防御措施基本靠谱

熟悉俺博客的读者都晓得——本人已经抹黑党国很多年了。从“第一篇政治博文”一直到“写这篇汇总”，时间跨度已超过“9.5年”，四舍五入就算十年吧 :)

俺记得很多年以前，有人在博客评论区对俺说（以下是大意）：

你小子能在网上得瑟，是因为网监部门还没注意到你；等到哪天网警开始盯上你，你就等死吧。

那时候，很多事情都还没有发生，俺缺少反驳的素材。如今可以理直气壮地反驳了——**朝廷的有关部门早就盯上俺了；遗憾的是：他们拿俺一点办法也没有。**

想看证据的话，请围观前几天的那篇《开博十周年大事记》。俺摘录其中几个要点：

\1. 早在2011年的“中国茉莉花革命”期间，俺就连发多篇具有“煽颠”性质的博文（注：那几篇显然能评上“煽动颠覆国家政权”的大罪）

\2. 到了2016年，朝廷向 Github 发出【政府删除令】，企图干掉俺维护的《太子党关系网络》项目（注：Github 有骨气，此项目至今屹立不倒）

\3. 针对俺 Gmail 邮箱的两次【国家级入侵】（注：分别出现在2011和2017）

\4. 2017年针对俺博客评论区的大规模刷屏（注：Blogspot 评论系统有“验证码机制”，想达到那种刷屏速度，需要好多个职业五毛一起刷）

上述这种种的迹象早已说明——俺是朝廷有关部门的眼中钉。

花了这么多口水，就是想说明一点——俺的防御措施还是基本靠谱滴！

换句话说，俺的防御措施不敢说完美（完美是不可能滴），但至少【没有】明显的漏洞。否则的话（如果有明显漏洞），俺要么被跨省，要么帐号被攻陷，又怎么能在将近十年的时间里“肆意抹黑朝廷，恶毒攻击党和国家领导人”？

★本文的目标读者

开博这么多年来，有一个感慨——（在墙内）很多具备政治素质的人，缺乏信息安全的技能；所以他们无法利用互联网与党国斗争。

虽说墙外已经有很多民运网站，也有很多民运人士开设了社交网络（SNS）帐号。但他们毕竟生活在墙外。天朝的民主化进程，不可能光依靠海外人士，关键还是需要靠咱们这些生活在天朝的民众。所以今天这篇，首先是为了帮助那些【有志于从事反党活动的网民】。

其次，是为了帮助那些捍卫互联网上【言论自由】的人。俺曾经写过一篇《"对抗专制、捍卫自由"的N 种技术力量》，谈到这方面的问题。

当然啦，所有的技术都是【双刃剑】——都可能被滥用。某些在网络上干坏事的家伙，也会从本文中受益。关于这点，俺也很无奈 :(

但是，俺不会因为技术存在被滥用的可能性，就停止对技术的传播和普及。

★本文与"之前写的信息安全教程"的关系

开博这么多年来，俺已经写了很多信息安全相关的【扫盲教程】（参见这篇末尾的：**和本文相关的帖子**）。今天要聊的很多内容，之前的教程都已经有了。那么，为啥俺还要写这篇捏？

因为之前写的那些，都只是针对某个具体的方面或某个具体的软件。而**本文就是为了——把所有这些【串起来】**，以方便那些在"信息安全领域"刚刚入门的同学。

为了避免老读者说俺"炒冷饭"，本文包含了一些过去没聊过的内容。另外，最后一章还附上几个实际案例，作为【反面教材】。

★两个【核心原则】

假如你想要效仿俺——长期利用互联网进行反党活动。如下两个原则需要时刻牢记。

**

原则1：确保你的身份隐匿

原则2：确保你的帐号安全**

下面的讨论，都是围绕这2条来展开。

关于"身份的隐匿"，俺补充说一下：即便你的肉身位于【墙外】，确保身份隐匿依然是必要滴！

★如何选择【网络平台】？

◇首先，【绝对不要】使用【国内】的网络服务

（此处所说的“国内”，指的是：伟光正具备【司法管辖权】的范围，含香港/澳门，不含台湾）

如果你想要在网络上进行敏感的政治活动，这个原则一定要牢记。因为用国内的网络服务进行反党活动，会大大增加你暴露的风险。

以俺自己来举例：

刚开博的时候（2009年初），俺同时也注册了 CSDN 的帐号，并在 CSDN 上架设了一个镜像博客（详情参见《开博十周年大事记》）等到后来，俺越来越放肆地抹黑党国，那个 CSDN 帐号也就用得越来越少了。

虽然俺全程使用 TOR 访问 CSDN（也就是说，CSDN 的服务器无法知道俺的公网 IP）。但是它还是会知道俺的在线活动时间。请注意：“时间线”也会构成某种信息量。关于这方面的详细介绍，请看《如何隐藏你的踪迹，避免跨省追捕》系列教程的第9篇：

《[如何隐藏你的踪迹，避免跨省追捕9]：从【时间角度】谈谈社会工程学的防范》

除了“在线时间信息”。考虑到如今很多网站都重度依赖 JavaScript 脚本（禁用 JS 脚本，网站就没法用）。所以，你如果用了墙内的网络服务，其网站上的 JS 脚本【有可能】收集到你本机的一些系统信息。

如果说“时间信息”和“系统信息”还不足以吓唬你，俺再提一个事情：当你使用墙内的网络服务，你所有的【用户行为】都有可能被有关部门收集和监控。

啥是“用户行为”捏？假如你用的是聊天服务（IM），你的用户行为就是——“你写过和看过的全部内容”；假如你使用的是邮件服务（Email），你的用户行为就是——“你发送和接收的所有邮件”。

请注意：“用户行为”所包含的信息量实在太大了。只要你一不小心，在其中涉及到与你真实身份相关的信息，这个信息就有可能成为日后追溯你身份的线索。（不信的话，请看本文末尾的其中一个反面案例）

还有一个大伙儿容易忽略的【阴招】：

网警如果盯上你（在墙内网站）的帐号，可以直接找到相关的公司，就可以拿到你的【帐号密码】。然后，网警可以直接控制该帐号。

举个例子：当网警控制了你的某个 IM 帐号，就可以用你的身份，去与该帐号的其他联系人聊天（是不是很阴险？）

◇为啥【不要】注册“独立域名”？

俺已经不止一次被问到：为啥博客没有用“独立域名”？老实说，俺一直觉得：只要博客内容足够好，有没有独立域名其实无所谓。

另外，如果从“信息安全”的角度来讲，独立域名还会增加额外的风险。

因为域名是稀缺资源，凡是要注册独立域名，自然涉及到【购买】（也就是【付费】）的问题。不管你是支付现金还是比特币，都会暴露“与你身份相关的信息量”。（通俗地说：增加了身份暴露的风险）

◇为啥【不要】搭建“自己的 Server”？

（注：本小节说的 Server 是广义滴，包括“物理主机、VPS”）

1. 考虑到【付费】的风险

（跟“域名”的情况类似）一旦你要搭建自己的 Server，也要涉及到【付费】的环节。如上一节所说，“付费的环节”会增加身份暴露的风险。

2. 考虑到【安全加固】的专业性

一般来说，用来提供网络服务的 server 通常会安装 Linux；还有极少数装 Server 版 Windows 或某种 UNIX（BSD 是 UNIX 的一种）。不管你的 Server 用哪种操作系统，都需要进行【安全加固】。

本来，“安全加固”已经是个很专业的领域，懂行的人就相对较少。然后，你还要考虑到——**本文讨论的是“反党活动”**。也就是说，你的安全加固，【不仅仅】是防“普通骇客”，还要防【御用骇客】。显然，“御用骇客”要比“普通骇客”牛逼得多。不妨稍微透露一下：由于工作关系，俺曾经跟御用骇客打过交道，知道他们的份量。（更多的细节，俺不便多说）

除非你自己是一个非常资深的信息安全从业人员，并且你对“服务器安全加固”这个细分领域非常熟悉，并且你对 Server 所用的操作系统的安全特点非常熟悉。请扪心自问一下：上述这3条，你都能达到吗?

达不到的话，还是死了这条心——不要自己去装 server。

3. 考虑到【时间和精力】

还有一个原因，也顺便说说——自己维护 server，需要消耗【很多】时间和精力。

请注意：“安全加固”并【不】是一劳永逸滴，而是个【动态过程】。如果你自己维护 server，需要时刻关注各种安全新闻，掌握你使用的操作系统和软件是否曝光新的安全漏洞，了解最新出现的入侵手法，诸如此类。一旦出现安全漏洞，你需要第一时间进行升级/维护。另外，你还要定期进行安全检查/安全审计……

凡此种种，都非常消耗时间与精力！

◇是否用【暗网】？

这个问题分两种情况：

\1. 如果需要你自己搭建服务器，那么答案是“否定”滴。（原因参见前一个小节）

\2. 如果无需你自己搭建服务器，由靠谱的第三方提供服务器，那么答案是“肯定”滴。

说到“暗网”，补充说一点：

很多人过度迷信“暗网”的能耐。俺要提醒一下：“暗网”只是帮你完成【网络层面】的隐匿性。要想彻底地隐身，你要在【多个层面】进行防范。（其它层面的防范，下面章节会聊到）

◇【国外】商业公司之间的权衡

简单说几个参考点：

1. 公司的口碑

不要用那些口碑差（历史上有污点）的平台。这个道理很显然，就不举例了。

2. 在华业务的规模

如果某个商业公司在天朝有庞大的商业利益，那么，一旦有关部门找这家公司索取用户私密信息，这家公司为了顾及在华商业利益，就【有可能】屈服于朝廷。

说到这里，已经猜到有同学会反驳俺——既然这样，为啥还用“微软网盘”分享电子书?

（对这类反驳，请看下一条的解答）

3. 网络服务的种类

不同类型的网络服务，对安全性的要求也不同。

比如说，俺会使用微软的网盘，但俺肯定【不】考虑微软的邮箱。

为啥捏?

因为俺的网盘本来就是用来进行【公开分享】，对保密性没啥要求。而电子邮箱对保密性的要求（相比“网盘”）要高得多。

所以在选择电子邮箱方面，Google 的 Gmail 显然比微软的 Hotmail/Outlook 更好（Google “在华业务”相比其“全球业务”，可忽略不计）

同样的道理也适用于 Github。虽然 Github 已经被微软收购了，但俺无所谓。因为俺的 Github 帐号

也是用来进行【公开分享】滴。又没啥私密性的东西。最坏情况下，也就是帐号被封掉。真有那么一天，俺再转移战场继续战斗嘛。

★如何注册【隔离的】虚拟身份?

◇选择一个【全然不同】的网名

你要选择一个【完全不同】的网名。这个网名与你之前使用过的【任何一个】网名都【不】能有相似之处。

说到这儿，顺便分享一下俺起网名的经验。

老读者应该知道：俺刚开博的时候，只是想写点编程方面的博文，算是某种经验分享。当时想过用“喜欢软件开发的极客”，但这个名称太长；改成“喜欢编程的极客”，还是太长；后来又改了几次，才想到如今这个“编程随想”。这个名称比较短，而且还能搭配罗丹那个“思考者的雕像”。

俺的经验是：

网名太长就不好记（从传播学角度而言是不利滴），太短又容易跟其它名称混淆（导致“识别度”降低），所以要找一个长度较短同时还具有一定识别度的名称。

◇注册时，【不要】填写真实的个人信息

比如说，注册电子邮箱时，会让你填写生日，你可不要写真实的哦，瞎掰一个就行啦。

至于像【手机号】、【身份证号】之类的，更加【不】能据实填写。

（注：“短信验证”的问题，下面会聊到，别急）

◇注册的【全过程】都要走【匿名网络】

“注册帐号”是这个敏感虚拟身份的【起点】。如果注册的过程中暴露了身份信息，那么今后再怎么隐匿都【没用】啦!

所以，你要确保——注册的全过程都通过【匿名网络】来进行。这样才可以【彻底避免】“公网 IP 暴露”的风险。换句话说，即使在你注册的过程中，网站服务器记录了你的“访问者 IP”，这个 IP 也【不是】你真实的公网 IP。

◇啥是【匿名网络】?

“匿名网络”，洋文叫做“anonymity network”。顾名思义，是用来帮助你实现【匿名化】的手段之一。通过匿名网络进行各种操作（比如在网上发布言论），可以让【网络层面】的【逆向追溯】变得极端困难。

注：很多人把“匿名网络”与“暗网”混为一谈，其实这是两个不同维度的概念。之所以会有这种混淆，是因为几个知名的工具（Tor、I2P）既是“暗网”，也是“匿名网络”。

◇为啥“只用翻墙代理”还【不够】可靠?

因为翻墙工具的设计初衷是为了“突破 GFW”，而【不是】为了“匿名化”。

如果你非常在意“匿名化”（比如：想在网上发表敏感的政治言论），那么你就必须使用专门的【匿名网络】。

◇如何使用“匿名网络”?

名气最大的匿名网络有两个，分别是 Tor 和 I2P。俺个人推荐 Tor，但如果你想用 I2P，俺也不反对。

关于这两款，俺都写了教程（如下）。

Tor 的教程：

《“如何翻墙”系列：扫盲 Tor Browser 7.5——关于 meek 插件的配置、优化、原理》（这篇讲“Tor Browser 套件”的使用，比较傻瓜化，支持各种桌面系统）

《扫盲 Arm——Tor 的界面前端（替代已死亡的 Vidalia）》（这篇讲 Linux 下如何使用“裸 Tor”，技术门槛比上一篇高）

《“如何翻墙”系列：关于 Tor 的常见问题解答》（这篇是 FAQ）

I2P 的教程：

《“如何翻墙”系列：简单扫盲 I2P 的使用》（这篇是 I2P 入门教程）

◇如何解决【短信验证】的问题?

先声明一下：

由于俺开博比较早（2009年初），那时候移动互联网尚未普及，很多网络帐号【无需】绑定手机。于是就便宜了俺这种懒汉——省去了很多麻烦。

如今，越来越多的帐号需要绑定手机（注册帐号时，需要【短信验证】）。这时候，**你【千万不要】用真实手机进行绑定操作！！！**

很多同学要问了：那咋整啊？大致有两种【通用】的招数可以搞定（如下）：

招数1：虚拟号码

你上网搜索一下：`虚拟号码 短信验证`，应该能找到很多【免费】的“虚拟号码服务”。这种服务可以用来帮你接收验证短信。

用这个招数的时候，有一点请注意——使用“虚拟号码服务”的【全过程】，也要基于【匿名网络】哦！

招数2：【不记名】的手机卡

如今天朝的手机卡，全都已经【实名制】了。要找这种【不记名】的手机卡，需要去【境外】。据说香港就有。（至于还有哪些地方可以买到，欢迎列位看官补充）

当然啦，你没必要为了搞个手机卡，专程到境外跑一趟；可以利用某次境外旅游的时候，顺便买一个。

用这个招数的时候，有几点请注意：

\1. 购买“不记名手机卡”，建议用【现金】（因为现金也具备【不记名】的特性）

\2. 在使用“不记名手机卡”的过程中，要确保你的手机系统是【纯洁】滴（可以考虑去搞一个“功能机/非智能机”来干这事儿）

\3. 通过【境外】的移动网络接收“验证短信”。

\4. 一旦接收完验证短信，这张“不记名手卡”就拔下来，以后也【别】再用了。

\5. 注册的操作过程应该在【PC 端】进行。

\6. 不光是注册过程，以后也【不要】在任何手机上操作“你注册的敏感帐号”（手机的危险性，后续章节还会单独谈）

★【硬件】层面的防范

◇总是启用“开机密码”和“硬盘锁”

一些比较大牌的笔记本电脑，都有这两项功能。当你入手了一台笔记本电脑，首先把这两项开启。

启用了这两项之后，每次你开机（冷启动），都需要先输入两个密码，分别是“开机密码”和“硬盘密码”。可能某些同学会嫌麻烦，俺要强调一下：**想提升安全就不要怕麻烦！**

当然啦，这两个玩意儿的【可靠性】到底有多高，是很难讲滴——因为不同品牌（厂商）的笔记本电脑，这俩密码的实现机制，差别很大。但“启用”总归比“不启用”要好。

另外，正是因为笔记本自带的硬盘锁不一定靠得住，所以你需要在“操作系统级别”进行【全盘加密】（本文后续的章节会聊到这个）。

◇把一些多余且危险的 BIOS 选项禁掉

考虑到不同年代、不同品牌的笔记本，BIOS 选项差异较大。俺在这里只举几个例子。大伙儿要举一反三。

比如说：对于英特尔（Intel）架构，要把 ME（Management Engine）禁掉——这玩意儿有安全风险。另，AMD 架构也有类似的玩意儿，叫 PSP（Platform Security Processor），也要禁掉。（注：有些 BIOS 无法禁用 ME 或 PSP）

比如说：当你装好系统之后，应该到 BIOS 的启动配置界面中，把其它的启动项都禁掉，只保留“硬盘启动”这一项。

比如说：“网络唤醒”的功能就没啥必要，而且有风险。

……

（还有很多，就不逐一列举了。再次提醒大伙儿：举一反三）

★【操作系统】层面的防范

◇如何选择操作系统?

如果你用的操作系统，其本身就有很多安全问题，那当然不行。所以第一步是：选择某种【靠谱的】操作系统。

1. 【不要】使用预装的操作系统

俺【从不】使用笔记本内置的操作系统。**俺的习惯是——只使用自己亲手装出来的系统。**

为啥捏？因为你无法判断预装的系统是否【纯洁】。如果操作系统本身有安全隐患，后面聊的所有安全加固措施都是白搭！

可能有些同学认为俺故意耸人听闻，建议这些同学去搜一下前几年的新闻。

随手举个例子——2015年，联想 PC 因【预装】流氓软件，在美国遭遇集体诉讼，赔了好多银子。

2. 强烈建议【不要】用 Windows

这个已经是老生常谈，具体就不展开了，参见下面这篇：

《吐槽一下 Windows 的安全漏洞——严重性超乎想象》

3. Linux 比 Mac OS 更好

说到这个话题，很多读者以为俺担心 Mac OS 是【闭源】。其实这只是一个方面。甚至都【不是】最主要的方面。

俺重点想说的是——【攻击面的确定性】。Linux 的特点是——【发行版非常多】。不同的发行版，内置软件的种类和版本各不不同，内核（kernel）版本不同，内核的编译参数不同……而且 Linux 还支

持多种 CPU 芯片，既有 x86 系列（Intel/AMD），也有【非】x86 系列（比如 ARM）。

一言以蔽之——Linux 由于发行版之间【巨大的差异】，会导致攻击面【非常不确定】。因此，在不了解你系统配置的情况下，入侵者的"攻击难度 & 攻击成本"会急剧增大。

再来看苹果的桌面系统。因为 Mac OS【没有】"发行版"这个概念。或者换种说法，Mac OS 只有一个发行版（就是苹果官方维护的那个）。所以跟 Linux 一对比，Mac OS 的系统环境（攻击面）就显得非常【确定】了。

另外，还有很多其它因素导致了——Linux 比 Mac OS 更有利于安全加固。具体请看下面这篇：《为什么桌面系统装 Linux 可以做到更好的安全性（相比 Windows & macOS 而言）》

4. 如何选 Linux 发行版?

Linux 的发行版，大概有上百种之多。经常会让新手困惑。

如果你是 Linux 的新手，先看《扫盲 Linux：如何选择发行版》，了解一些基本概念。

"如何选择 Linux 发行版"这个问题，【没有】放之四海皆准的标准答案。不同的场景，不同的使用者，自然会有不同的择。以本文这个话题，俺的建议是：【保守型】、【社区维护】、【口碑好】

你在符合这几个条件的发行版中，挑个你觉得最顺手的。

5. 善于折腾的同学，也可以考虑 BSD 社区

BSD 社区比较有影响力的发行版包括如下几个：

FreeBSD

这是 BSD 社区最知名的一款，也是该社区最多人用滴。

OpenBSD

这款是以【安全性】著称滴。它的社区采用了很多机制（代码审计、最小化权限、最小化安装 ...）来提升系统的安全性。

而且 OpenBSD 社区非常强调【默认安装的安全性】。也就是说，默认装好，不作任何配置，其安全性就已经足够好。根据历史记录，从1997年到俺写本文之时（2019年初），OpenBSD 在默认安装下只曝光了2个【远程】漏洞（时间分别在2002年、2007年）。这种水平，其它操作系统望尘莫及。

值得一提的是：有很多知名的软件（比如：OpenSSH、tmux、LibreSSL、PacketFilter）就源于 OpenBSD 社区。

NetBSD

这款是以【可移植性】著称滴。号称支持的硬件平台超过任何一款 Linux 发行版。

（不过捏，这个优势对个人用户而言，意义不大）

小结

综上所述，用 Linux 或 BSD。本文后续的讨论，也在这两者基础上展开。

◇强烈建议使用【虚拟机】来强化安全

刚才提到的"操作系统防范"，主要是针对你的【物理系统】（也称作"Host OS"）。接下来要谈的是——你【一定要】在 Host OS 之上，用【虚拟化软件】来搭建若干个"虚拟系统"（也称为"Guest OS"或"VM"）。**这种玩法可以大大提升你防御入侵的能力；**在某些特定情况下，还可以避免你暴露公网 IP（本文末尾的某个反面案例会提及这点）

1. 虚拟化软件的选择

如果你对技术方面【不太懂】，优先考虑的虚拟化软件是 VirtualBox（VBox）或 VMware。这两款的知名最大，用的人也最多；你如果碰到问题，比较容易找到相关的文档/教程。

俺当年写的《扫盲操作系统虚拟机》系列教程，主要是也是拿这两款来举例。

至于那些善于折腾的同学，当然还可以考虑别的软件，比如：KVM、Xen、QEMU.....

因为虚拟化软件的很多功能是相通滴。所以捏，如果你用了别的虚拟化软件，依然可以参考俺上述的系列教程，然后自己举一反三。

2. Guest OS 的选择

关于“Guest OS 的选择”，可以参考“Host OS 的选择”。不过俺要提醒一下：Guest OS 最好与 Host OS【有所差别】。

为啥捏？因为要规避【单点故障】的风险。关于这个话题，可以参考如下博文：

《聊聊【单点故障】——关于“德国空难”和“李光耀”的随想》

3. 设定“安全基线”，并做到【定期回退快照】

关于这个话题，请看俺那个“虚拟机系列教程”的第7篇：

《[扫盲操作系统虚拟机7]：如何用“快照”辅助安全加固、强化隐私保护？》

4. 虚拟系统的【颗粒度】

最起码你得有【两个】Guest OS（VM），一个用于你的日常身份，另一个用于你的敏感虚拟身份。这种做法的“颗粒度”【最大】，也是安全性【最差】滴。

【更好的做法】是——把你敏感的虚拟身份操作的 N 个网络帐号拆分到 N 个 VM 里。以俺为例：有一个 VM 是专门用于“编程随想的 BT Sync”（Resilio Sync）；有一个是专门用于“编程随想的 OneDrive”（微软网盘）；有一个是专门用于“编程随想的 Twitter” ……另外，还有若干个虚拟机用于俺的真实身份。所以，俺的笔记本电脑里有很多虚拟机。

拆分的颗粒度变小之后，即使某个 Guest OS（VM）被入侵，最坏也只是损失一个帐号。

说到“颗粒度”，还有一个需要讨论的问题是：翻墙软件应该装在哪个虚拟机？关于这个问题，在下面讨论【网络】的章节中再细聊。

5. 如何防止【虚拟机穿透】？

在这个小节的最后，俺来聊一下“虚拟机穿透”这事儿。所谓的“穿透”就是指：入侵者先攻占 Guest OS，然后利用“虚拟化软件”本身的漏洞进行“穿透”，渗透到 Host OS 中。

这么干，从技术上讲是可行滴，而且也有安全研究人员演示过这个招数。但这个招数的实现难度非常非常大（需要【同时】具备很多条件，才能做成），一般人其实不用担心这个风险。不过俺在本文开头也说了，本教程是要应付【御用骇客】滴。所以，这种情况的概率虽然小，还是值得考虑滴。

那么，如何防范捏？比较好也比较彻底的做法是【物理隔离】。比如说：在多台【物理主机】上配置不同网络帐号的操作环境。即使某个物理主机被入侵了，其它物理主机上的网络帐号【不】受影响。

最近这些年，笔记本电脑都已经白菜价了。所以，多买几台笔记本电脑来进行物理隔离，钞票的压力应该不大吧？

刚才只是介绍了“物理隔离”的其中一种玩法。其它几种玩法请参见《如何防止黑客入侵》系列教程的第8篇：

《[如何防止黑客入侵8]：物理隔离的几种玩法》

◇确保 Host OS【极简】

使用了“虚拟化软件”之后，你应该把【所有的】日常操作都放到 VM 中进行。普通身份的操作放到“普通 VM”，敏感身份的操作放到“敏感 VM”。

于是捏，你的 Host OS 几乎就不需要啥软件了（除了虚拟化软件和系统自带的软件）。

通过把 Host OS 简化到极致，也就把 Host OS 的攻击面降低到最小。你始终要记住：**Host OS 非常重要！！！ Host OS 如果沦陷，运行在它之上的所有 Guest OS 也将沦陷。**

★【应用软件】层面的防范

◇选择软件的几个原则

1.【不要】使用国产软件

这其中的道理就类似于——不要使用国内的网络服务。

如果你由于某些原因不得不用某个国产软件（比如说：QQ、迅雷……），应该把这个国产软件单独隔离在某个虚拟机（Guest OS）中，【千万不要】装到 Host OS 中，也【不要】安装到那些用于敏感身份的虚拟机。

2. 安装的软件【越少越好】

安装的软件越多，你所暴露出的【攻击面】就越大。

因为每个软件都无法做到尽善尽美，每个都有可能存在潜在的（未曝光的）漏洞。

3. 尽量使用【成熟度比较高】的软件

举个【反例】来说事儿。在浏览器方面 IE 就是个典型的反例。最近这20年，IE 曝光的【高危】安全漏洞（远程执行类、提权类）那真是一坨又一坨，简直惨不忍睹。像 IE 这么烂的浏览器，如果你用它去上网，简直找死。

4. 优先选择【开源】的软件

商业公司必定【逐利】，所以商业公司有作恶（耍流氓）的动机和动力。比如说，用户数据可以转化为利润（变现），所以商业软件（尤其是用户量很大的那些），总是喜欢收集用户隐私。

相比之下，开源社区【没有】盈利的压力。所以，开源软件耍流氓的情况，不敢说完全没有，但肯定远远少于商业软件。

5. 优先选择【发行版官方仓库】所含的软件包

如果你使用 Linux 或 BSD，优先使用发行版官方维护的软件包。

比如说，两个软件，功能差不多，其中一个包含在官方软件仓库中，另一个没有。通常情况下，应该选那个软件仓库已有的。

“官方仓库”相当于某种程度的【背书/担保】。口碑越好的发行版，其官方仓库中的软件，可信度越高。

6.（在安全方面）版本【并非】越新越好

很多同学有个【误区】，以为版本越新越好。**其实不然！**（至少在安全方面，这点并【不】成立）

关于这方面的讨论，可以参见下面这篇博文。虽然这篇博文讨论的是 Firefox，但道理是相通滴！

《基于安全性考虑，如何选择及切换 Firefox 版本？》

另外，前面谈“如何选 Linux 发行版”，俺强调用【保守型】的发行版。道理也在于此。

◇磁盘加密工具的使用

（磁盘加密软件很重要，俺单列一个小节来讨论）

【把你的硬盘加密】——这是对付警方【取证软件】的重要法宝。另一个好处是，万一你的笔记本不小心失窃了，窃贼也无法看到硬盘的内容。

由于磁盘加密软件依赖于具体的操作系统，下面俺以【Linux】来说事儿。用 BSD 的同学请依样画葫芦。

1. 用 dm-crypt（LUKS）全盘加密

装 Linux 系统时，`/boot` 通常会单独分一个区。`/boot` 的加密会比较麻烦。不太熟悉 Linux 的同学，可以把 `/boot`【之外】的其它分区都用 LUKS 加密。如果你想把 `/boot` 也加密，可以到网上搜相关的教程。

（注：因为 `/boot` 分区通常很小（再大也就一百多兆），而且【不】存放个人数据，该分区的保密性要求并不高）

然后你可以在已经用 LUKS 加密的分区上，用 LVM（Logical Volume Manager）创建一系列逻辑分区（也叫“逻辑卷”）。请注意：规划逻辑分区时，要特意留一个空闲的（未用的）。

关于 LUKS（也叫“dm-crypt”）的使用，请阅读如下教程：

《扫盲 dm-crypt——多功能 Linux 磁盘加密工具（兼容 TrueCrypt 和 VeraCrypt）》

关于 LVM 的使用，请阅读如下教程：

《扫盲 Linux 逻辑卷管理（LVM）——兼谈 RAID 以及“磁盘加密工具的整合”》

2. 用 TrueCrypt / VeraCrypt 在“空闲逻辑分区”创建【敏感加密盘】

先再次唠叨一下：虽然 TrueCrypt（以下简称 TC）这个开源项目已死，但其替代品 VeraCrypt（以下简称 VC）完全兼容 TC 的功能和加密盘格式。所以，这两个软件大体上可以通用滴。

俺在前一个步骤提到：预留空闲的逻辑分区。到了这一步，你选择这个空闲的逻辑分区，用 TC/VC 在这个分区上创建加密盘，用来存放【特别敏感】的数据（跟你的敏感虚拟身份相关的数据）。为了叙述方便，该加密盘称之为“敏感加密盘”。

TC/VC 的加密盘【格式】有一个优点，是其它磁盘加密格式所不具备滴。那就是 TC/VC 的加密盘【没有】特定的文件头，也【没有】任何其它特征。换句话说，给你一段看似随机的数据，你【完全无法】通过数据本身来判断其是否 TC/VC 的加密盘数据。

这个优点很重要。因为某个未格式化的分区，其数据看上去是随机的；把这个分区做成 TC/VC 的加密盘之后，数据依然看上去像是随机的。这样就【不易】引起怀疑；即便引起了怀疑，你也可以抵赖，一口咬定该分区就是闲置未用滴。

3. 详细的磁盘布局

在如下博文中，俺介绍了详细的磁盘布局方案，并有大量配图。

《扫盲 Linux 逻辑卷管理（LVM）——兼谈 RAID 以及“磁盘加密工具的整合”》

4. “敏感加密盘”的【配置】原则

由于这个加密盘特别重要，建议使用如下措施来强化其安全性：

4.1. 认证因子要包含【key file】

也就是说，要么只用“key file”，要么是“密码 + key file”。一旦你的认证因子中包含了“key file”，暴力破解就变得【不可行】。

“key file”是啥玩意儿捏？通俗地说就是：用某个【内容随机生成】的文件作为加密盘的“钥匙”（其效果类似“密码”）。但是“key file”比“密码”更优秀之处在于——由于 key file 的内容是【随机】生成滴，你自己也不知道其内容（而且你也不可能把它的内容背下来）。因此，一旦你【彻底】销毁了这个 key file 之后，连你自己也【不可能】再打开加密盘。所以，“key file”机制不但可以对付【暴力破解】，还可以用来对付警方的【酷刑逼供】。

请注意：“key file”要用【二进制】文件，文件至少64字节或更大（以确保【熵值足够大】）。TC/VC 自身都提供了“生成 key file”的功能，以确保生成的“key file”是【高度随机】滴。

4.2. 【多重】加密

TC/VC 支持多重加密，每一重都使用不同的加密算法。

4.3. 设置【隐藏卷】

“隐藏卷”也叫“内层卷”。有了它，你就可以享受“Plausible Deniability”带来的好处啦 :)

5. “敏感加密盘”的【使用】原则

由于这个加密盘实在太重要了，俺建议遵循如下使用原则：

5.1. 何时挂载“敏感加密盘”?

由于“敏感加密盘”非常敏感重要，没事就别挂载它。

只有当你确实需要操作那些敏感身份的帐号，才开启/挂载（mount）“敏感加密盘”

5.2. 关机 VS 待机 VS 休眠

当你要【长时间】离开自己的电脑——应该【关机】（shutdown）；而【不要】“休眠”（hibernation）或“待机”（suspend，stand by）

（注：如果你不太熟悉 TrueCrypt 或 VeraCrypt，对本小节提到的很多名词和建议，会觉得纳闷。请参考如下几篇教程）

《TrueCrypt 使用经验》（系列）

《扫盲 VeraCrypt——跨平台的 TrueCrypt 替代品》

在本文发出后没几天，俺又写了一篇，专门补充【磁盘加密】相关的细节，尤其是——【如何对付警

方】。
《如何用“磁盘加密”对抗警方的【取证软件】和【刑讯逼供】，兼谈数据删除技巧》

★【网络】层面的防范

◇网络方面的基础知识

如果你的网络技能不够扎实，建议先【看完】如下这个长篇教程：
《计算机网络通讯的【系统性】扫盲——从“基本概念”到“OSI 模型”》

◇严格设置 OS 自带的防火墙（Host OS 和 Guest OS 都要设）

无论是 Linux 还是 BSD 都内置了操作系统级的防火墙（Linux 社区有：iptables 和 nftables；BSD 社区有：PF、NPF、IPFW）

你应该养成一个好习惯，一装好系统就开启防火墙。

设置防火墙要遵循【最小权限原则】（凡是不需要的，都是禁止的）。

比如说，你要配置一台个人用的 PC，并且【不】需要远程访问。那就应该把防火墙设置为“禁止对外监听端口”。

（同样的原则也适用于 Guest OS 自带防火墙的配置）

◇Guest OS 的网卡模式

前面提到了：要使用【虚拟化软件】来强化安全性。所以，你还需要对 Guest OS 设置“虚拟网卡模式”。

俺的建议是：

1. 网关 VM【别用】bridge 模式，应该用 NAT 模式（NAT 可以起到类似防火墙的效果）
2. （在极少数情况下）如果你需要【跨物理主机】共享“网关 VM”的翻墙流量，想让网关 VM 对其它物理主机暴露监听端口，可以在 NAT 模式下添加端口映射（洋文叫“port forwarding”）。如果俺没记错的话，VirtualBox 和 VMware 都支持 NAT 模式下的端口映射。
3. “操作上网帐号的 VM”用 host-only 模式（VBox 下还可以考虑 internal 模式，比 host-only 更严格）通过这种方式，【彻底隔绝】该 VM 中的【任何软件】的网络直连，强迫他们都经过“网关 VM”联网。

详细的“原理说明”和“配置教程”，参见如下两篇：
《[如何隐藏你的踪迹，避免跨省追捕6]：用虚拟机隐匿公网 IP（原理介绍）》
《[如何隐藏你的踪迹，避免跨省追捕7]：用虚拟机隐匿公网 IP（配置图解）》

◇避免用无线网络（比如：wifi）

为啥要避免用无线捏？一言以蔽之，（相比物理网线）无线网络会显著增加你的攻击面。

比如说：那些安全防范等级较高的公司或机构，其【核心网络】肯定是物理布线，而不会走 wifi 之类的无线网络。

◇设置家用路由器（如果有的话）

哪怕是比较普通的家用路由器，也提供了一些基本的安全设置（比如：防火墙、MAC 地址绑定……）

你应该把这些设置都用起来，还是刚才提到的老话——配置时参照【最小权限原则】。

另外，如果你采用了前面提到的【物理隔离】方案，那么你就会有 N 台物理主机。这种情况下，你要在家用路由器上进行一些配置，使得这 N 台物理主机相互【不可见】。

◇操作敏感帐号，要【全程走】匿名网络（Tor 或 I2P）

（关于这点，前面聊"注册帐号"时已经提过。为了加深你的印象，俺再次唠叨）

◇Tor 或 I2P 要加【前置代理】

前面俺已经聊过了：Tor 和 I2P 是两个最有影响力的匿名网络。因为俺个人推荐 Tor，所以下面拿 Tor 来说事儿。想用 I2P 的同学，请依样画葫芦。

从许多年以前，Tor 在天朝就无法独立联网了。因为 GFW 把 Tor 视作心腹大患，封杀了所有的 Tor 中继（Relay）。

所以，要想在墙内用 Tor，需要让 Tor 借助某个【前置代理】（这个前置代理，通常是某个【可用的】翻墙软件）。后来 Tor 官方推出的 Tor Browser 内置了一个 meek 插件，可以辅助 Tor 在墙内联网。这个 meek 插件也算是（某种意义上的）前置代理。

本来，让 Tor 走前置代理是为了突破 GFW 对 Tor 的封锁。但这么干产生了若干【额外的好处】——让你的网络传输更加健壮。啥意思捏？下面俺解释一下。

好处1——ISP 【无法】知道你在用 Tor

无论你在家上网还是在公司上网，最终你的网络流量都要经过 ISP。也就是说，ISP 完全有可能监控你的流量。

当你采用"Tor over 前置翻墙软件"，ISP 监控你的流量，看到的是"前置翻墙软件"的流量。由于翻墙软件的流量都是【加密】滴，所以 ISP 无法解密，也就无法知道你在用 Tor。

在全球的网络用户中，Tor 用户的比例依然很【低】；在天朝，这个比例会【更低】（墙内网民对隐私还是不够重视）。由于 Tor 是设计用来【隐匿网络踪迹】滴。如果让 ISP 看到你在用 Tor，终归不是啥好事儿。

所以，即便你在【墙外】上网，此时 Tor 可以独立联网，你还是应该给 Tor 配一个【加密的】前置代理。

好处2——双保险

当你采用"Tor over 翻墙软件"，你的"真实上网流量"的外面其实包裹了【两层】，第一层是 Tor，第二层是"前置翻墙软件"。由于包裹了两层，类似于某种【双保险】。

换句话说，如果有人要在网络层面截获你的"真实上网流量"，必须先破解最外层（前置翻墙软件的加密），然后再破解次外层（Tor 的加密），才能看到你的真实上网流量。由于 Tor 本身是【强加密】，而翻墙软件的加密也不会太弱。所以，同时破解这俩层加密的可能性，小到可以忽略不计。

◇【不同】身份的帐号，要使用【不同】（各自独立）的匿名网络环境

假设你让"真实身份"的帐号和"敏感身份"的帐号使用同一个 Tor/I2P 环境，有可能导致这两个帐号在【同一时间】使用了【相同出口节点】。

如果这种情况长时间持续出现，就会使得这2个帐号产生某种【相关性】，从而让人怀疑这2个帐号背后是同一个人。

更详细的说明，参见下面这篇博文中【公网地址】导致的关联性这个章节。
《[如何隐藏你的踪迹，避免跨省追捕10]：从【身份隔离】谈谈社会工程学的防范》

◇“翻墙软件”与“你的网络帐号”【隔离】

刚才提到了：用某些翻墙工具作 Tor 的【前置代理】。那么，这些翻墙软件要放在何处捏？
俺的建议是——把翻墙软件放到【另一个】虚拟机，以进一步降低网络帐号的风险。为啥捏？因为你无法知道：翻墙软件本身是否会耍流氓。
在下面的这篇博文中，俺详细介绍了几种部署方式。其中包括“Tor 的【前置】部署”和“Tor 的【后置】部署”。
《[如何隐藏你的踪迹，避免跨省追捕8]：如何搭配“多重代理”和“多虚拟机”》
对于本文的目标读者，如果你通过浏览器（Web 方式）操作网络帐号，此时你的上网软件（浏览器）是可信滴，而翻墙工具不一定可信。所以你应该采用“Tor 的【后置】部署”。

★【Web】层面的防范

◇如何选择浏览器？——俺推荐 Firefox

说到“选浏览器”这个话题，其实也就是在 Chrome/Chromium 或 Firefox 这两家二选一。因为前面说了，你上网的系统应该是 Linux 或 BSD。像 IE、Edge 之流，就甭考虑啦。
俺个人的建议是——Firefox
俺知道读者中有很多 Google 的粉丝，也有很多 Chrome/Chromium 的粉丝。对俺倾向 Firefox 会感到不理解。建议这些同学看如下博文的分析：
《弃用 Chrome 改用 Firefox 的几点理由——关于 Chrome 69 隐私丑闻的随想》

◇如何选择 Firefox 的【版本】？

关于 Firefox 版本的问题，列几个要点：
\1. 【千万别用】中国版
\2. 一定要用国际版中的 ESR（长期支持版本），别用 Release 版，【更不能】用 Beta 或 Nightly 版。
\3. 在某个【恰当的时间点】切换 ESR，【不要】一发布新的 ESR 就切换。

如果你不太了解 Firefox 的版本体系，未必明白上述这几个是啥意思。请参考如下博文：
《基于安全性考虑，如何选择及切换 Firefox 版本？》

◇关于 Firefox 的【插件和扩展】

先说一下，“插件”（plugin）和“扩展”（extension）是两种不同的东西。在“这篇博文”中，有一个小节专门谈：插件和扩展的区别。
对于操作重要帐号的浏览器——第三方“插件”一个都【不装】；第三方“扩展”要【尽量少】，最多只装几个安全相关的，而且要选那种口碑足够好的。

◇如何【加固】Firefox?

对于不太懂技术的同学，建议直接用【Tor Browser】套件。这个套件是 Tor 社区在 Firefox 的 ESR 版本基础上，又进一步强化了安全性。而且还绑定了 Tor。

至于那些喜欢折腾的同学，可以自己用 `user.js` 对 Firefox 进行很多的定制。主要原则就是——把 Firefox 的【攻击面】降到尽可能小。

如何【深度】定制 Firefox？参见如下教程：

《扫盲 Firefox 定制——从“user.js”到“omni.ja”》

（注：上述教程只是教你如何配置 Firefox。“如何对 Firefox 进行【安全加固】”属于另一个话题。考虑到这个话题太小众，暂时还没动手写）

◇操作敏感帐号，确保【全程】HTTPS

要做到这点，有个前提——敏感帐号对应的网站要提供【全站 HTTPS】。

考虑到如今 HTTPS 已经很普及啦。知名的网络服务，基本都支持【全站点 HTTPS】。有些网络服务做得更贴心——即使你用【明文】的 HTTP 协议访问，它也会把你重定向到【加密】的 HTTPS。

有些同学会问：万一碰到某个网络服务，不支持 HTTPS，咋办?

俺的建议是：如果某个网站到现在（2019）都还【没】实现“全站 HTTPS”，那这个网站也够烂的，不用也罢。

为啥要强调【全程 HTTPS】捏?

前面提到了“全程走匿名网络”，但是匿名网络中的节点都是由世界各地的志愿者维护的，不排除其中会有恶意节点（蜜罐节点）。如果你访问网站的流量是加密的 HTTPS 流量，即使是恶意节点，也【无法】看到你的上网内容（网页、图片、视频、等），更加不可能去篡改。

◇确保浏览器【专用】

为了说明【专用】是啥意思，举个例子。

博客的读者都知道：俺有个推特帐号，是专门用来发布“博文更新的通知”。

在俺的电脑上，有一个专门的 VM（Guest OS）用来操作这个推特帐号（刚才聊虚拟机“颗粒度”的时候，已经提到这点）。这个 VM 里面的 Firefox，除了访问 Twitter 的网页，【绝对不】访问其它任何网站。

确保浏览器【专用】，可以预防大部分的 Web 攻击。就算不幸被入侵了，（只要没出现“虚拟机穿透”）受影响的范围也只局限在这个 VM 内。至于如何【彻底】防范“虚拟机穿透”，本文前面某章节已经聊过。

★【社会工程学】层面的防范

对于技术高手而言，“社会工程学”的防范【最难】。因为“社会工程学”探讨的是【非】技术领域的话题。

这方面的防范，靠的不是你的技术，而是你的【心理素质】。比如说：是否足够理性，是否足够细心，是否足够耐心，是否足够冷静 ……

（注：如果你之前没听说过“社会工程学”这个概念，可以先看下面的系列博文）

《扫盲社会工程学》

◇关于【偷窥】

（俺特意把这个放在第一条，因为谈到社会工程，很多人只想到对网上其他人的防范，而忽略了【身边人】）

当你操作敏感的虚拟身份时，要确保【不】被周围的人看到。如果是在公共场合（包括公司里），还需要警惕周边的摄像头。

再次拿自个儿举例：

俺有时候会在【上班时间】回复读者评论，那是因为俺作为公司的高管，有独立办公室 :)

如果俺是在开会或者与别人讨论问题，肯定不会运行"编程随想"相关的 VM（甚至连存放这些 VM 的【敏感加密盘】都不开启）。

既然说到"偷窥"，再顺便强调一个常识——输入重要密码记得遮挡键盘（尤其是在公共场合）。比如说：用笔记本的同学，（输密码时）把屏幕合拢到与键盘成30度角。

◇关于【信任】

当你使用敏感的身份与别人沟通（哪怕【私密】的沟通），【永远不要】提及自己的真实身份。

就算你能相信对方，你又如何确保沟通双方的软件环境是可信的？你又如何确保沟通双方的物理环境是严密的？……（这样的反问句，俺可以写一大堆）

基于同样的道理，即使是与俺进行邮件沟通，你也【不要】暴露自己的身份信息。

◇关于【即时通讯】（IM）

聊天工具（IM）会暴露出比较多信息量。所以"编程随想"这个身份从未使用 IM 与读者沟通，最多只用邮件。（为了安全起见，俺如今连邮件也用得少了，主要在【博客评论区】与读者沟通）。

如果你确实想用 IM，那就只用【文本】形式，千万【别用】"音频 或 视频"（多媒体形式的 IM，暴露的信息量太大了）。

另外要提醒一下：【不要】过度迷信"端到端加密"。

某些同学【天真地以为】：采用了"端到端加密"之后，聊天内容就只有两人知道。**其实不然！**比如说：其中一人的 PC/手机中了木马，聊天内容就有可能外泄。这还只是一种可能性，还有其它很多种可能性。

◇关于【私密沟通】

前一个小节提到：聊天工具（IM）会暴露出比较多信息量。现在来解释一下。

先定义一下【私密沟通】的范畴——指的是那些【不】公开的一对一沟通。【至少】包括：两人聊天、非群发的邮件、社交网络的"私信"、等等。

私密沟通的【危险性】在于——这种沟通方式会让你放松警惕（这是由心理学层面决定滴）。

对照一下现实生活。当你处在一个【多人】的场合，你说话就会比较谨慎和自律。而在那种一对一私下沟通的场合，你说话的警惕性就会下降。这种情况下，你就更容易暴露出更多个人信息。

◇关于【社交网络】（SNS）

敏感虚拟身份使用的 SNS 帐号，要与你真实身份使用的 SNS 帐号【没有交集】。

比如说：俺的真实身份有一个 Twitter 帐号，但这个 Twitter 帐号肯定不会 follow 编程随想的 Twitter。

◇关于【密码】（password）

有一个大伙儿很容易忽略的盲点，是【密码】。

不同的帐号，密码也【不能】有相似之处。为啥捏？

因为你无法确定网站在存储密码的时候，是否符合安全规范。如果网站对密码的存储不够规范，然后网站的数据库还被入侵了（往往是这种不规范的网站，更容易被入侵），导致用户的【原始密码】曝光。（在剔除掉那些极简的傻瓜密码之后）那些密码【高度相似】的帐号，就有可能被关联起来，从而导致身份暴露。

（注：存储密码，规范的做法是——用足够【强】的散列算法，并配合【随机】撒盐，然后存储散列值。虽然只是短短一句话，可惜大部分程序员并不理解其背后的深意）

关于如何构造复杂密码，参见如下教程：

《[如何防止黑客入侵3]：如何构造安全的口令/密码》

◇关于【个人信息】

不论是写博客还是用 SNS（社交网络）与别人沟通，你所说的话，总是会不经意地暴露出一些个人的身份信息。

比如俺博客聊了这么多信息安全的话题，有些话题还比较“阳春白雪”（只有懂行的人才写得出）。因此，读者就能猜出，俺是在这个圈子里混的——这就是某种“个人信息”。

所以，除非你完全不说话，否则，总会有这样那样的信息流露出来。当你暴露的信息足够多之后，某些“有心人”就会根据这些信息，逐步缩小范围，逐步拼凑出你的完整脸谱。

那么，该咋办捏？

说到这儿，俺要借用《红楼梦》里面的名言——**【假作真时真亦假】**。也就是说，你要故意暴露【假信息】。通过这些【假信息】来干扰对方的视线。“假信息”关键在【质】而不在“量”。啥意思捏？就是说，“假信息”的数量并不需要太多，但一定要让人信以为真。

由于存在“假信息”的【干扰】，当“有心人”企图根据你暴露的信息来缩小搜索范围，你就有可能【漏网】——漏到包围圈之外 :)

◇关于【时间信息】

关于这个维度的讨论，之前已经专门写过一篇博文（如下）。

《[如何隐藏你的踪迹，避免跨省追捕9]：从【时间角度】谈谈社会工程学的防范》

正是因为这方面的考虑，所以俺要让自己的“上线时间”尽量【随机化】，不能有固定的模式。

◇关于【行文风格】

每个人的遣词造句都有其独特之处，这种独特性就像是语言层面的“指纹”。

举个例子：

J.K. Rowling 曾经用化名出了一本推理小说《布谷鸟的呼唤》（The Cuckoo's Calling）。某公司通过专门的软件对文字风格进行分析，发现此书与《哈利·波特》的行文风格高度一致，从而曝光了作者的真实身份。

所以，如果你的“虚拟身份”与“真实身份”都在互联网上留下【足够多】文字，别人【有可能】从“文字风格”发现两者的相关性。文字越多，被发现的可能性越大。（注：据热心读者反馈，可用专门的软件批量修改文章的文字风格。这种软件俺没用过，感兴趣的同学可以尝试一下）

俺比较幸运之处在于——本博客是俺第一个博客。在2009年之前，俺一直是网上的【潜水者】（从来不冒泡）。另外，俺在公司里也不会写长篇大论的文档。所以，在“行文风格”方面，俺的风险会比较低。

考虑到俺博客有不少程序员读者，顺便提醒【源代码风格】的“指纹”。其原理是类似滴。

前些年，俺大幅度改造博客的评论区界面，加了很多定制的 JS 脚本，当时就有热心读者提醒俺这个风险。今天顺便也解释一下。

作为一个老程序员，俺在公司里写了很多代码，但都是 C/C++、Java、Python（从俺写的编程博文，也能猜出这点）。而且俺在公司里写的都是【后端代码】（服务器端）。而博客评论区的改造属于【前端 JS】。因为前端与后端的差异太大，且编程语言也不同。因此，俺在这方面的风险也很小。

◇（其它）

社会工程学涉及的方方面面太多，肯定有些是俺漏了说的。欢迎列位看官到博客评论区继续补充。

★对【手机】的防范

关于“手机”的话题比较特殊，因为手机同时涉及前面提到几个层次，所以俺单列一章来讨论。

◇手机的风险

关于手机的隐私风险，这些年来，俺已经重复唠叨很多次啦。今天再来一次。

当你想用手机操作你的网络帐号，这已经隐含了一个前提——此手机必然是【智能机】。“智能机”的安全风险【至少】包括如下：

1. 【硬件探测器】太丰富，能收集的信息太多

手机包含的硬件探测器太多，至少包括：摄像头、麦克风、GPS、陀螺仪 ……

在这种情况下，如果手机中的某个软件（app）是恶意的，并且获得了足够的权限，那么这个 app 就可以监控你日常生活的方方面面。

比如说：通过“GPS 定位”或“基站定位”可以了解你日常活动范围，可以知道你用哪种交通工具（根据移动速度）……

2. 两大手机操作系统（Android & iOS）都不是【完全开源】滴

iOS 是闭源，这个众所周知了。

很多人【误以为】Android 是开源，其实它只有【一部分】是开源滴。如果要说得再详细一点，那就是——

Android 系统包括两部分：AOSP（Android Open Source Project）和 GMS（Google Mobile Services）。其中的 GMS【不】开源。

而且自从 Android 占据市场主导地位之后，Google 逐渐把 AOSP 中的模块转移到 GMS 中（注：Google 这么干，再次体现出商业公司的德性）。

3. 固件是【闭源】滴

请注意：固件处在操作系统的【下层】。固件如果不可信，比操作系统还麻烦。

4. 手机上无法实现【操作系统虚拟机】

到目前为止，手机上还无法实现“操作系统虚拟机”，也就是类似于 VMware 或 VirtualBox 之类的玩意儿。

而“操作系统虚拟机”是非常重要的安全防御手段（前面章节已经聊过）。

5. 手机上的【全盘加密】不够严密

虽然如今的 Android 和 iOS 都已经有了“全盘加密”，但它们机制和功能，对俺这类高危人士而言是【远远不够】滴。

为了长话短说，简单举个例子——

至今还没听说有哪个手机系统的全盘加密支持【key file】，但成熟的桌面加密软件（TrueCrypt/VeraCrypt）都有这个功能。

前面俺说过了——“key file”这种机制可以用来对付警方的【酷刑逼供】——只要你在被捕前销毁“key file”，之后连你自己都打不开加密盘（【酷刑】又有啥用捏？）而手机缺乏这个机制，也就意味着如果你被捕，警方（尤其是天朝警方）总是可以想办法迫使你解锁手机。

不光缺少“key file”功能，手机的磁盘加密还缺少其它很多重要的功能，比如“Plausible Deniability”，比如“自定义加密算法组合”，比如“自定义密钥迭代次数”……而这些功能对提高“加密盘的抗破解能力”，是非常重要滴！

6. 常用的手机软件（App），大部分都是来自商业公司

在《如何保护隐私》系列教程的第一篇，俺就特地强调了“商业公司”与“非盈利组织”的差异。很多人应该听说过“流量变现”，同样的道理，用户数据也可以变现。作为商业公司，“收集用户数据”自然成为他们的一大癖好。

7. 用户群很大的那几个 App，都很流氓

这个道理，俺也聊过多次了。像“微信/支付宝/百度/京东”这些 App，装机量都是以【亿】计。这么大的安装量，朝廷的有关部门，难道会不动心吗？假如有关部门找到这几家公司的老板，要他们稍微配合一下，在 app 里面玩点猫腻，像菊花疼、马淫、李阉红、刘强奸这些老板，**他们有胆量拒绝朝廷提的要求吗？**答案显然是【否定】滴！

因此，国内装机量特别大的 app，不耍流氓几乎不可能！

还有一个比较讽刺的是——所有这些公司（不管是老板还是公关部门），都会信誓旦旦地说：从来不耍流氓。但是大伙儿别忘了——这是在天朝，这是一个“诚信还不如狗屎”的国度。诸如此类的诅咒发誓，你当笑话听听就行啦，切莫当真。

◇结论

由于手机存在如此多的风险点。所以——

\1. 要【完全禁止】手机参与操作敏感的网络身份

\2. 如果某个网络服务只提供手机 App，而不提供“Web 界面”或“桌面客户端”，那么你就应该【弃用】这个网络服务

\3. 你在操作敏感的网络身份时，最好把手机放到别处（别忘了手机上的流氓软件有可能偷偷对你进行拍照/摄像哦）

★对几个【反面案例】的分析

为了进一步加深大伙儿的印象，俺给大伙儿准备了几个反面教材。

◇案例1：Freedom Hosting 网页挂马事件

Freedom Hosting 是暗网上提供托管服务的平台（在暗网的圈子里小有名气）。其站长被 FBI 抓了之后，FBI 接管了网站服务器，然后在页面中嵌入了某个恶意脚本。这个恶意脚本可以利用 Firefox 17.0 ESR 版本的某个漏洞。

当年的 Tor Browser 用的就是这个 ESR 版本的 Firefox。因此，当某个 Tor Browser 用户访问了这个挂马的页面，该脚本就会利用 Firefox 17.0 的安全漏洞，然后【绕过代理】，直接向某个 FBI 控制的服务器发送 HTTP 请求。

由于是【绕过代理】进行直连，所以 FBI 只要检查该服务器收到的 HTTP 请求，就可以知道这些中招的 Tor Browser 用户的【真实】公网 IP。

有些同学以为俺说这个案例，是想谈“修补漏洞”。可惜不是！因为任何浏览器都【不可能】保证零漏洞，所以光靠修补浏览器漏洞来对付这类威胁，不够保险。

更保险的做法是【系统级网络隔离】。如果上述这些 Tor 用户看过俺的教程，懂得用【虚拟机隔离】来隐匿公网 IP，那 FBI 的招数就失灵了——因为在【隔离】的虚拟机中，恶意脚本【对外直连】的

HTTP 请求会【失败】（发不出去）。

所以，这个案例的教训是——你要杜绝所有【不经代理】的网络直连行为。为了做到这点，要把【所有】敏感的上网行为都放到【虚拟机】中，以确保【所有】流量都经过你设定的“网关 VM”。

◇案例2：顶级黑客 Jeremy Hammond 被捕

此人是大名鼎鼎的 LulzSec 骨干成员，网名 yohoho。从其辉煌战绩可以看出，他显然是技术高手。而且他也一向谨慎，LulzSec 的其他成员并不知道他的真身。

后来，LulzSec 的某个成员（网名 Sabu）被 FBI 逮捕，并转为卧底。所以 FBI 拿到了 yohoho 与 Sabu 之间的所有聊天记录。

在与 Sabu 聊天时，yohoho 无意间提到自己参加了对“共和党全国代表大会”的抗议示威，并被警方拘留。这个信息量已经足够高，足以把范围缩到很小。警方开始怀疑 Hammond，并监控他家的网络流量。观察多日后发现：他家 Tor 流量出现的时间段，与 yohoho 上线的时间点高度吻合。

于是 FBI 申请了“强行搜查令”，破门而入……

此案例的第1个教训是——不要暴露【信息量太高】的真实个人信息。

此案例的第2个教训是——**Tor 前面再放个【加密】前置代理**（这招俺唠叨了很多年啦）。如果 Hammond 遵守这个原则。那么，FBI 监控他家的流量，就无法判断他是否在使用 Tor（因为 Tor 流量被包裹在前置代理的加密流量之内）。

俺博客上，和本文相关的帖子（需翻墙）：

《计算机网络通讯的【系统性】扫盲——从“基本概念”到“OSI 模型”》
《如何保护隐私》（系列）
《如何防止黑客入侵》（系列）
《如何隐藏你的踪迹，避免跨省追捕》（系列）
《扫盲社会工程学》（系列）
《扫盲操作系统虚拟机》（系列）
《如何用“磁盘加密”对抗警方的【取证软件】和【刑讯逼供】，兼谈数据删除技巧》
《为什么桌面系统装 Linux 可以做到更好的安全性（相比 Windows & macOS 而言）》
《吐槽一下 Windows 的安全漏洞——严重性超乎想象》
《扫盲 Linux：新手如何搞定 Linux 操作系统》
《扫盲 Linux：如何选择发行版》
《扫盲 Tails——专门强化隐匿性的 Linux 发行版》
《文件加密的扫盲介绍》
《扫盲 Linux 逻辑卷管理（LVM）——兼谈 RAID 以及“磁盘加密工具的整合”》
《TrueCrypt 使用经验》（系列）
《扫盲 VeraCrypt——跨平台的 TrueCrypt 替代品》
《扫盲 dm-crypt——多功能 Linux 磁盘加密工具（兼容 TrueCrypt 和 VeraCrypt）》
《扫盲文件完整性校验——关于散列值和数字签名》
《文件备份技巧：组合“虚拟加密盘”与“网盘”》
《“如何翻墙”系列：关于 Tor 的常见问题解答》
《“如何翻墙”系列：扫盲 Tor Browser 7.5——关于 meek 插件的配置、优化、原理》
《扫盲 Arm——Tor 的界面前端（替代已死亡的 Vidalia）》
《“如何翻墙”系列：简单扫盲 I2P 的使用》
《多台电脑如何【共享】翻墙通道——兼谈【端口转发】的几种方法》
《如何让【不支持】代理的网络软件，通过代理进行联网（不同平台的 N 种方法）》

[如何隐藏你的踪迹，避免跨省追捕0]：为啥要写此文？

文章目录

★引子

在早年的安全圈内，也曾有一篇名叫“如何隐藏你的踪迹”的帖子，洋文原名叫“How to cover your tracks”（原始链接、网页存档）。那可是国外老牌黑客组织（The Hacker's Choice）在上个世纪写的。不过那篇帖子有很大篇幅是在教你：如何在入侵系统之后，不留下痕迹。而今天俺要聊的内容，基本与入侵【无关】。

本系列的【主题】

有很多因素会导致真实身份的暴露。

俺在这个系列中会介绍：如何从【各个层面】防止真实身份的暴露？

本系列的【受众】

这个系列是写给那些——通过互联网进行各种【政治敏感活动】的网友，包括但不限于：异议人士、民运人士、维权人士……

不过捏，凡事都有两面性。某些通过网络干坏事的家伙，或许也能从本系列博文中获得启发。技术总是双刃剑，这也是没办法滴 :(

本系列的【意义】

请参见博文《“对抗专制、捍卫自由”的 N 种技术力量》

★为啥要写此文？

从最近2年的趋势来看，互联网越来越成为揭露社会阴暗面、批评党的一个利器。因此，党国为了维护统治阶级的利益，也会想方设法扼杀对党不利的言论（具体如何扼杀，可以参见“党和互联网的较量”）。在党国采取的各种措施中，就包括“跨省追捕”这一招。（“跨省追捕”一词来源于2009年轰动一时的“王帅发帖事件”）

俺发觉很多网友非常缺乏这方面的技术常识，这可真是大大便宜了党国的爪牙。另外，自从开博客之后，已经有很多个读者给俺写邮件，询问这方面的知识。如此看来，专门开一个系列，扫盲这方面的技能，还是很有必要滴。

★免责声明

为了免遭别人拍砖，省却不必要的口水战，在介绍正式内容前，先来个免责声明：

免责声明一：

本系列仅介绍各种保护隐匿性的技术手段及注意事项，并【不】意味着俺鼓励大伙儿去干坏事。

免责声明二：

本系列介绍的招数，可以大大降低暴露的可能性，但是并【不能确保】百分百不暴露。（要知道，绝对的安全是不存在滴）如果你用了这些招数，还是暴露了，那可别怨俺 :)

★本系列的目录

为了方便阅读，把本系列帖子的目录整理如下（需翻墙）：

俺博客上，和本文相关的帖子（需翻墙）：

《计算机网络通讯的【系统性】扫盲——从"基本概念"到"OSI 模型"》
《如何保护隐私》（系列）
《如何防止黑客入侵》（系列）
《为啥朝廷总抓不到俺——十年反党活动的安全经验汇总》

[如何隐藏你的踪迹，避免跨省追捕1]：网络方面的防范

文章目录

前一个帖子介绍了本系列的动机。本帖开始讲正题。

在网络方面，要想隐藏自己，重点是要保护好自己的真实IP和通讯内容。下面且听俺细细道来。

★网络基础知识

首先，本文主要谈【网络】方面的防范；其次，本系列的后续很多篇，都会大量涉及网络的相关知识。

除非你是网络方面的行家，否则的话，俺建议你先【看完】如下这个长篇教程：
《计算机网络通讯的【系统性】扫盲——从"基本概念"到"OSI 模型"》

★【公网 IP 地址】的防范

大部分暴露行踪的人，都是因为暴露了自己的【公网 IP】。所以俺首先来聊一下公网 IP 的保护。

◇啥是“IP 地址”?

要了解“公网 IP”这个概念，你首先得明白何为“IP 地址”?
对这个概念的解释，参见开头的章节（★网络基础知识）中引用的那篇扫盲教程。

◇“上网行为”与“公网 IP”的关系

如果在家里上网，需要通过 ISP（互联网服务提供商）才能接入互联网。不论你用的是宽带（比如：ASDL、有线通、FTTB ……）还是老式的电话 Modem 拨号，ISP 都会分配给你一个公网 IP 地址（以下简称“公网 IP”）。对于家庭宽带上网，这个 IP 地址很可能是【动态】滴。
如果你在公司/政府机关/企事业单位（以下简称“组织机构”）里上网，其实也差不多。这些组织机构也需要通过 ISP 才能接入互联网。ISP 同样会给每一个组织机构分配公网 IP。与“个人用户上网”不同的，每个组织机构分配的 IP 地址可能不止一个，而且组织机构分配的 IP 地址通常是【固定】滴。

◇“公网 IP”如何暴露你的行踪?

请看下面的例子：
假设你是一个愤青，经常在家里上网发帖。某天，你头脑一发热，在某些网站（尤其是党国控制的网站）发表了恶毒攻击党、毁谤国家的言论。
虽然网站不知道你是谁，但是他们可能会保存如下的【用户访问日志】：某个公网IP、在某个时间点、发表了某个言论。因此，你的这个抨击党国的行为，就被记录在案了。
然后，如果你的言论引起了“有关部门”的注意（“有关部门”是传说中的神秘部门），它们会要求该网站提供上述的【用户访问日志】。通过该记录可以知道你发表言论时，所使用的公网 IP 及具体的时间点。
再然后，有关部门根据这个【公网 IP】，可以判断对应的是哪个 ISP（中国电信、中国网通、中国移动、广电 ...）。然后它们会找到这个 ISP，调阅那个时间段的【IP 地址分配记录】。通过这个记录，就可以知道：在那个时间段，这个公网 IP 分配给了哪一个终端用户。
对 ISP 来说，一旦定位到了终端用户，基本上也就知道你的家庭住址了。然后，有关部门就会直接去你家敲门。俺估计，当年不幸被跨省追捕的王帅同学，就是这么暴露滴 :(

刚才说了“家庭上网”的情况，假如你在组织机构（公司、政府部门）里面上网，其实也差不多。有关部门照样能通过网站的【用户访问日志】，定位到你所在的组织机构的公网 IP。然后就到你上班的地方排查。这时候，距离你的彻底暴露，估计也就是一步之遥了。

◇防范措施之“代理上网”

要防止因为“公网 IP”而暴露，简单有效的办法，就是通过【代理】来上网。
代理又分两种方式：“Web 代理 & 代理软件”。（注：俺这里所说的“代理软件”是广义滴，包含 VPN，具体介绍可以参见之前的帖子“如何翻墙？”）
当你通过代理的方式访问某些国内网站并发表敏感言论，网站服务器记录的是“代理的公网 IP”，而【不是】你自身的公网 IP。而且，同一个时间段，使用这个代理的人可能很多，因此就难以区分出：到底是谁通过代理发表了不良言论。
再补充一下：如果你使用的是翻墙代理，其代理服务器往往位于境外，而且往往是朝廷无法控制的（比如那些具有法轮功背景的翻墙代理）。这就为网监人员追踪你的真实身份，设置了很大的困难/障碍。

◇防范措施之“公共场所上网”

在家里使用代理上网，可以防止暴露自家老巢的公网 IP。但是，你还可以有另外的法子——干脆不在家上网。你可以拿着自个儿的笔记本电脑，到某些带有 WiFi 热点的公共场所上网，或者到网吧上网。在这些公共场合的网络中，完成某些敏感的操作。如使用这个招数，要牢记【打一枪换一个地方】的原则，以防万一。

提醒一下：即使在公共场所上网，最好也使用代理，以增强隐匿的效果。在本系列后面的帖子里，俺会单独写一篇，介绍公共场所上网的注意事项。

★【传输内容】的防范

前面费了很多口水谈“公网 IP”。但是，光隐藏“公网 IP”是远远不够滴！你还需注意——【网络传输内容】可能会让你曝光。

◇“传输内容”为啥会暴露你的行踪?

请看下面的例子：

如今，一些比较大的公司或政府机关，可能会在内部网络部署“网络行为审计系统”。这是啥玩意儿捏？通俗地说，就是进行【网络监控】的设备。它会实时监视网络上传输的数据内容，并进行分析，然后把分析结果保存下来。通过这种设备，公司的老板们可以了解到哪些员工利用工作时间，干一些不良的活动（比如：看有颜色的网站、打网游、炒股、看视频……）。

假设你经常上某国内论坛去谩骂伟光正，或者你经常浏览国外的反动网站，没准也会被单位里的审计系统记录在案。到时候你们的网管去查阅审计系统的历史记录，你的小动作就全暴露啦。

◇如何防范?

为了防止你上网的行为被监控，传输的内容被截获，要记住如下两点：

第1点：能用加密协议，就【别用】明文协议。

比如：很多网站同时支持 HTTP & HTTPS。你一定要用【HTTPS】协议。因为 HTTPS 是【加密】滴，有助于防止被传输的内容被监视。

第2点：能用加密代理，就【别用】明文代理。

无论你用“代理软件”还是“Web 代理”，一定要确保：你使用的代理是【加密】滴！

不加密的“明文代理”非常不安全——既容易被入侵，也容易被监控。

★结尾

今天就先说到这儿。作为本系列的第一篇，本文只是聊【最基本】的技能，而且只是针对【网络】方面。

今天这篇相当于“热身” :) 在本系列后面的帖子，俺还会聊很多更高级的话题。

本系列的下一个帖子，俺会介绍个人软件方面的防范。

[如何隐藏你的踪迹，避免跨省追捕2]：个人软件的防范

文章目录

俺在前面的帖子介绍了和网络有关的招数。接下来，再来说说和个人软件有关的防范知识。

★为啥要防范个人软件?

2009年那个臭名昭著的“绿霸事件”，大伙儿还记得吧？（不知道的同学，可以看“这篇”）

据说这玩意儿可以把你上过哪些网站、甚至通过键盘打了哪些字都记录下来，然后发送到“绿霸”所在公司的服务器上。如果你的电脑上装了它，基本上你的一举一动，都在党国的监视之下（这就是传说中的 老大哥在看着你）。这时候，你再搞什么“磁盘加密，上网代理”，也是白搭。

所以捏，你首先必须彻底杜绝这种带有后门性质、木马性质，且受控于党国的软件。那么除了“绿霸”，还有哪些软件【可能会】暴露你的隐私和行踪捏？俺粗略整理了几种类型，介绍如下。

★即时聊天（IM）软件

说到聊天工具，自然就会提及 QQ——毕竟它占据了国内即时通讯市场的大头。所以俺就具体说一下，党国是如何利用腾讯公司来监控/打压不和谐的声音。

◇监控聊天内容

首先，党国会利用 QQ 监控你的聊天内容（不论是“一对一聊天”还是“群聊”）。比如，党国60大寿前后，很多 QQ 群被封——因为群里面的讨论的内容涉及了某些敏感词。另外，还可以到网上搜一下标题为腾讯 QQ 如何监视你的聊天记录的帖子，看看别人的遭遇。另外还有很多揭露 QQ 的文章（请翻墙看“这里”、“这里”）。

◇利用聊天历史记录取证/抓人

有些维权人士、异议人士、民运人士明显缺乏对党的斗争经验，居然敢用QQ相互联络。结果是：不光自己暴露了，还牵连了别人。这样的例子太多了，俺就拿前几天（4月20日）刚宣判的“严晓玲、范燕琼案件”来说一下。

话说福建的年轻女子严晓玲神秘而离奇死亡。有些网友看不下去，在网上为她喊冤。结果捏，统统被福州警方捉拿归案——罪名都是“涉嫌诬告”，其中三人被判有期徒刑。在办案过程中，咱们的公检法部门，充分利用 QQ 进行顺藤摸瓜，以迅雷不及掩耳盗铃之势，把这些维权者一网打尽。请看被捕网友郭宝峰的原话：

> 各位，够胆就继续用 QQ 吧，监狱等着你们去闯。警方当时不但通过 QQ 把我们几个一网打尽，而且一个警察曾经拿着厚厚的一叠我和游兄的聊天记录让我签字，我完全没有想到我和他说过如此多的话而且多数记录和本案没有直接关系。我拒绝签字，那位警察就说：“你不签也没关系，我们有证据。”

这下知道 QQ 的利害了吧?

◇利用 QQ 搜罗电脑中的文件

今年（2010）1月份，坊间又有新的传言，说 QQ 还会扫描你硬盘上的某些文件（具体请看：“这里”、“这里（墙外）”、“这里”）。如此一来，QQ的危险性又增加了一层。不排除今后有关部门利用QQ庞大的装机量，收集 QQ 用户电脑中的敏感文件。这虽然有点耸人听闻，但俺一向不惮以最坏的恶意来揣测咱们伟大的党。

顺便说一下，有2个 QQ 的替代品：其一是：腾讯提供的 Web 版的 QQ 工具（在“这里”）；其二是：开源的Pidgin（原先支持 QQ 协议，后来疼讯升级了 QQ 版本后可能无法支持了）。用这2个替代品，可以避免 QQ 客户端软件扫描你的硬盘文件，但还是【无法】避免你的聊天内容被监控。

◇还有哪些 IM 可能被监控?

既然党国的有关部门会利用 QQ 进行监控，它自然也不会放过其它几款【国内】的聊天工具——毕竟它们的老窝还在天朝之内，有关部门要它干啥，都得乖乖听话。因此，如下的几款 IM，都【有可能】处于党国有关部门的严密监视之下。

移动的飞信
淘宝的旺旺（后改名阿里旺旺）
新浪的 UC（前身是朗玛 UC）
网易的泡泡
百度的 HI

另外，Skype 虽然是国外进口软件，但它的【Tom 版】，后门可是大大滴哦！所以，如果你想用 Skype，一定要用国际版（在“这里”），【千万别用 Tom版】！

◇如何选择【国外的】IM?

在本系列第4篇，俺会专门谈【通讯工具】这个话题，到时候再谈“IM 如何选型”。

★杀毒软件

不光国内的 IM 工具不保险，国产的杀毒软件也要多加小心。

◇国产杀毒软件的危险

在2010年初，瑞星和奇虎360给大伙儿上演了一出【狗咬狗】的闹剧（瑞星咬奇虎的在“这里”，奇虎咬瑞星的在“这里”）。大伙儿在看热闹之余，应该也察觉出某些杀毒软件厂商的险恶用心——【在杀毒软件中暗藏后门】。到底有多少国产杀毒软件暗藏后门，俺不敢随便乱说。但是既然360开了个头，不排除有其它厂商会效仿；另外，也不排除某些后门，就是党国的有关部门要求增加的。

咱不妨设想这样一个场景——
你用了某款国产杀毒软件，而该软件恰好又安放了受党国控制的后门。由于它要“查毒”，当然可以名正言顺地对你硬盘中的每一个文件进行扫描；然后，在扫描的过程中，顺便收集一下你电脑中的敏感文件；再然后，利用“在线升级”的机会，顺便把收集到的信息传回厂商的服务器上；最后，党国的有关部门发觉了你的异动，到你家来敲门。

◇应对措施

要规避上述风险，也挺简单——就是尽量【不用国产】的杀毒软件，改用国外滴。 目前朝廷的触角还不够长，还没有伸到国外杀毒厂商那里。

★浏览器

再来说说上网必用的工具——浏览器。

◇浏览器的选型

最近这些年，搞 Web 浏览器似乎成为时尚潮流，稍有实力的公司都去跟风。因此，Apple 推出了 Safari、Google 推出了 Chrome、腾讯搞了个 TT 浏览器、奇虎搞了个360安全浏览器……真是好不热闹。

面对这么多花哨的浏览器，大伙可得留神。像腾讯、奇虎这些【国内】公司推出的浏览器，同样有可能暴露你的行踪。具体的道理与杀毒软件类似，俺就不再多啰嗦了。

俺通常推荐【国外】知名的浏览器。目前比较知名的有 Firefox、Chromium 家族、IE（注：Edge 算“Chromium 家族”）。这三类如何选捏？从安全角度（防骇客入侵）考虑，首先要排除掉 IE（具体原因请看“这篇博文”）；从隐私方面考虑，Firefox 比 Chrome 要好（具体分析请看“这篇博文”）。

◇上网痕迹

两大主流的浏览器（Firefox、Chrome）都支持【隐私浏览模式】。如果你通过浏览器进行某些敏感的操作，建议在“隐私浏览模式”下进行。这样，当你干完事情后，只要关闭了浏览器，你的浏览历史（包括浏览器 cookie）就【不】会保存下来。

提醒一下，**“隐私模式”对浏览器【插件】是无效滴！**比如隐私模式可以控制“浏览器 cookie”，但是【无法】控制 Flash 的 Cookie！也就是说，即使你用了隐私模式，Flash 的 Cookie 还是会保存在硬盘上。

（注：“插件”与“扩展”是两码事儿，别搞混喽。关于两者的差异，请看“这篇博文”中的这个小节“◇插件和扩展的区别”）

对于浏览器【插件】留下的痕迹，该咋办捏？有如下两个招数——

办法1

你的浏览器【不要】装任何插件（比如：Flash 插件、Java 插件、PDF 插件、媒体播放器插件 ...）

办法2

如果你由于某种原因，不得不安装某些插件，还可以利用“虚拟机的快照功能”。你先设置好一个干净的虚拟机快照，然后在该虚拟机中上网。上网结束后，回退到快照，那么你在虚拟机中的任何痕迹都会被抹去。

没用过虚拟机的同学，请看《扫盲操作系统虚拟机》系列教程。

◇简单删除 VS 彻底删除

还有一个需要提醒的是：浏览器的缓存通常是存储在硬盘的某个目录中。浏览器清除缓存的时候，只是【简单删除】这些缓存文件（**“简单删除”完全不同于“彻底删除”**）。

对于“简单删除”，还是有可能用专门的工具软件恢复出来的（比如：反删除工具、警方的取证软件）。更保险的做法是，你需要采用一些【磁盘加密】的措施（加密浏览器存放历史信息的目录），以防止“浏览历史”被恢复出来。

关于“磁盘加密 & 彻底删除”，在本系列的下一篇《操作系统的防范》有相关介绍。

◇警惕【CA 证书】引发的风险

浏览器的 CA 证书，一直是被忽视的薄弱环节。直到2010年初，CNNIC 通过招摇撞骗，成为根证书颁发机构，这个薄弱环节才引起少数网友的重视。如果你对 CA 证书缺乏了解，请看看《数字证书及 CA 的扫盲介绍》，然后再看《CNNIC 证书的危害及清除方法》。看完之后，应该就明白 CNNIC 证书的危害性了。

有必要提醒一下：

流氓的 CA 机构，当然不止 CNNIC 这一家。比如后来又冒出一个"沃通/WoSign"，也很流氓（参见"这篇博文"）。

★输入法

（得益于热心网友在评论中的提醒，俺再追加"输入法"这一节）

◇输入法的危险性

其实，早期传统的输入法都是单机软件，没有太大的隐私风险。

但是随着这几年网络的发展，Web 2.0 的普及，连输入法软件也开始上网了（还美其名曰"云输入法"，可见 IT 业多么喜欢炒概念）。很多新推出的输入法，可以把用户的"个性化词库"同步到输入法厂商的服务器上。有了这个功能，无论你使用哪台电脑，只要该电脑可以联网，你就可以体验到自己的个性化词库。

但是，这样也就带来了一个潜在的隐私问题。因为输入法软件非常了解你经常输入哪些词组，而且把你经常输入的词组保存到你的个性化词库，然后再把词库同步到服务器上。如果你用的输入法是国产软件，那么，党国一样可以逼迫输入法的软件厂商把每一个用户的个性化词库公开给有关部门。然后有关部门就可以通过你的个性化词库，知道你平时经常输入哪些东东。

◇如何防范?

如果你觉得输入法的"在线同步词库"功能很爽，让你很 High，让你无法割舍，俺还是奉劝那句老话：别用国产滴，用进口滴。

如果你觉得"在线同步词库"只是个花哨的功能，无所谓，俺建议你还是用【单机模式】的输入法比较保险。这时候无论国产/进口，差别应该不大。

★总结

前面哇啦哇啦说了许多，大伙儿应该看出点门道了吧？——但凡【国产的】、带有【网络功能的】应用程序，只要【用的人多】了，都可能被有关部门盯上。所以，不要怨俺崇洋媚外，实在是党国的爪牙无孔不入啊！

可能会有人质疑说：老美的软件，也可能植入了美帝安全局（NSA）的后门啊！

但是俺想反驳说：即便国外的那些软件，都带有美国国安局的后门，也不用怕——毕竟美国佬对咱们【没有】司法管辖权 :)

关于"个人软件"的话题，就先聊到这儿。下一个帖子，俺来介绍一下"操作系统的防范"。

[如何隐藏你的踪迹，避免跨省追捕3]：操作系统的防范

文章目录

上一个帖子，咱们聊了“如何避免个人软件泄露你的行踪”，今天的主题是“操作系统相关的防范”。而操作系统相关的防范，归根结底，就是保护你操作系统中的【各种数据】不被泄露。

★先来个八卦

考虑到本帖有点长，先拿一个八卦旧闻来给各位同学提提神，顺便也让大伙儿了解了解：保密性是何等滴重要。

想必列位看官都还记得，当年陈冠希同学的艳照门丑闻吧？（就算你不记得艳照门丑闻，总该还记得那些艳照吧）陈同学之所以身败名裂，就是因为太不注重敏感数据的保密性啦。

首先，他没有把重要的数据（也就是那些艳照）加密存放；其次，在电脑拿去送修的时候，也没有进行相关的处理（至少也应该先把硬盘留下来）。最后的结果就是——既搞臭了一堆女明星，也便宜了广大男网民。

★电脑中的数据，如何让你暴露？

通过上述例子，列位看官应该体会到【保密性】的重要了吧？回到咱今天的话题，“数据的保密性”和“隐藏踪迹”有啥关系捏？且听俺细细道来。

◇电脑中的虚拟身份

当你用【虚拟身份】上网时，不可避免的，会有一些相关的信息保存在电脑上。比如：

> 很多用户为了省事，会让浏览器记住自己登录的网站的用户名/密码；
> 有些网站，会把你的登录名保存到 cookie 中；
> 你可能会把聊天工具设置成自动登录；
> 你的聊天历史可能会保存在本地硬盘上；
> .....

凡此种种，都可能在你的电脑中，留下和你的虚拟身份相关的信息。

◇电脑中的真实身份

另外，你除了用虚拟身份上网，还可能会用电脑干一些个人的事情，甚至用【真实身份】登录一些 Web 网站。因此，有些和你真实身份相关的数据，也会留在电脑中。比如：

> 邮件客户端（Outlook、Foxmail ...）的通讯簿；
> 你保存的一些个人的照片；
> 你使用的网银信息（如果你用这台电脑访问“网银”）；
> 你公司的一些文件（如果这是你的工作用机）；
> 有些网站，会把你的登录名保存到 cookie 中；
> 你的手机号（如果你在某些 IM 工具中绑定了手机号）
> .....

◇两种身份的关联

假设你是一名“地下工作者”，隐藏得很好，正在与党作斗争。结果有一天，由于某种原因，你电脑上的数据，落入他人之手。那么，拿到数据的人，可能会发现——网上的“某XX”原来就是现实生活中的“某叉叉”。这时候，你的踪迹也就彻底暴露鸟 :-(

★数据泄露的几种途径

那么，在什么情况下，别人会拿到你电脑中的数据捏？
俺总结了一下，大致有如下几情况：

◇电脑被入侵

首先要考虑的风险就是——你的电脑被入侵，并且很不幸地被植入了木马。在这种情况下，木马可能会盗取你电脑中的很多数据。（如果你在信息安全方面一窍不通，不知何为“木马”，请看维基百科的“这个页面”）

千万不要以为中木马是小事。如果是普通网友中招，可能确实是小事——植入木马的，可能只是一名普通的黑客/骇客。但如果你是一个小有名气的维权人士、异议人士、民运人士，那党国的走狗很可能会想尽办法让你中招（植入木马），然后利用你电脑中的木马监视你的一举一动。

另外再提醒一下：

最近几年（大约2010年之后），六扇门的相关部门（尤其是公安的【技侦部门】）已经开始采用你意想不到的技术来投放高级的木马，具体详情参见《如何对付公安部门的“网络临侦”？——“黑暗幽灵（DCM）木马”之随想》

◇电脑被没收

有时候，当党国的爪牙开始怀疑你的身份，它们可能会突然没收你的电脑，拿回去分析（行话叫做“信息取证”）。通过分析你电脑中的数据，就可以了解你在网上的各种虚拟身份。

如果你本身已经是一个【公开身份】的异议人士、维权人士、民运人士，那电脑被没收的概率就更大了（之前已经有很多类似案例）。然后，朝廷的走狗们可以分析你电脑中的信息，从而了解你与哪些人过从甚密、干过哪些对党不利的事情。

◇电脑公用

假设你的电脑不是你一个人专用，而是与别人合用，那也得小心。比如：你把电脑借给别人或者你使用公共场所（网吧、学校机房）里面的电脑；又或者家人合用一台电脑。

在一台【多人共用】的电脑上，你的个人隐私很容易暴露。

◇电脑遗失

这年头，台式机越来越少，笔记本电脑越来越普及。而且，电脑的“小型化趋势”越来越明显——比如“上网本、平板电脑”等。电脑小了，便于携带，但同时也增加了【丢失的概率】。

一旦你的电脑丢失，捡到的人又不愿意做活雷锋，那电脑上的数据就有可能暴露。

★如何防范?

经过前面漫长的铺垫，终于要说到本文的重点部分了 :)

◇防止电脑被入侵

关于“防范入侵”这个话题，内容可是相当的繁杂，三言两语是肯定讲不清楚滴。为了避免本贴过长，俺另外开个一个“如何防止黑客入侵”的系列，普及一下黑客入侵及木马的防范。

除了操作系统被入侵（被植入木马），其它几种情况（电脑被没收、电脑丢失、电脑共用）导致的风险，都可以用后续几个招数来化解。

俺再多啰嗦一下：一旦你的操作系统被入侵并被植入木马（尤其是很厉害的木马），后续的这些招数是【帮不了】你滴！这时候最保险的做法，只能是【重装系统】。

◇数据加密

首先，你要把一些重要的、敏感的数据，以【加密】方式保存。具体的加密方式，可以考虑如下几种。

1. 加密文件系统（EFS）

加密文件系统是比较方便的一种方法。你可以针对文件系统中的某几个文件或某几个目录，设置为加密存储。平常使用的时候，你完全感觉不到（用 IT 的行话，就是“对使用者透明”）。但如果别人拿走了你的硬盘，是无法读取出 EFS 里面被加密的文件滴；甚至同一个操作系统的其它用户，也是无法读取出被加密文件的内容。

EFS 有赖于特定文件系统的支持。如果你使用 Windows 系统（Win9x 不算，至少要 Win2000），必须得用 NTFS 格式的分区才行（不能用 FAT16、FAT32）；如果你使用 Linux，使用默认支持的 ext3 或 ext4 文件系统即可。关于 EFS 的更多介绍，请看“这个维基百科页面”。

（考虑到大部分读者用的是 Windows 系统，再多说两句）

Windows 传统的 EFS，有若干缺点——

其一、如果包含有 EFS 文件的系统重装了，你就再也无法打开这些 EFS 文件了；

其二、不便于在移动设备（如 USB Key、移动硬盘）上使用 EFS；

其三，如果要解决前两个缺点，需要导出/导入相关的密钥，但是步骤较繁锁。

想必微软也意识到这些缺点，从 Vista 开始（包括其后的 Win7、Win8、Win10），推出了 BitLocker 这款工具，可以解决上述缺点。详细的功能介绍，可以看“这个维基百科页面”。

2. 【专业的】磁盘加密工具

除了 EFS，还可以使用专门的加密软件来达到加密数据的效果。目前的文件加密软件五花八门，俺推荐一个相当牛逼的加密盘工具——TrueCrypt。关于该软件的介绍，可以看俺写的系列扫盲教程（在“这里”）。

补充说明：

本文写于2010年；到了2014年的时候，TrueCrypt 突然停止开发，于是俺又介绍了它的两款替代品，分别是 VeraCrypt 和 dm-crypt（LUKS）。这两款工具都可以完美兼容“TrueCrypt 的加密盘【格式】”。

换句话说：虽然 TC 这个【软件】已经死了，但是 TC 使用的【加密盘格式】，将长期存在。

3. 硬盘密码/硬盘口令

有些笔记本电脑（一般是“商用型笔记本”），提供了“硬盘密码”的功能。一旦设置了这个玩意儿，在开机时必须先输入该密码，才可以使用。设置了“硬盘密码”之后，即使把硬盘取来下，挂载到另外的电脑，也还是【无法】读取该硬盘的数据。

请注意：“硬盘密码”与“BIOS 开机密码”是两码事儿，别搞混喽！“BIOS 开机密码”并【不能】硬盘的数据。

有必要提醒一下诸位：不同的电脑厂商，其“硬盘密码”的实现机制不同，因此其强度（抗破解能力）也就不同。有鉴于此，俺建议把硬盘口令作为一种辅助手段，而【不要当作唯一手段】。

◇学会【彻底删除】数据

除了要懂得“数据加密技术”，还要懂得“数据销毁技术”。

很多傻瓜用户【误以为】：只要把文件搞到回收站，就万事大吉了；还有一些不那么傻瓜的用户【误以为】：把回收站清空，就没事了。这些都是极其幼稚的想法。

如果你只是普通地删除一个文件（比如：Windows 资源管理器的删除功能、命令行的删除命令），那么该文件的内容，还是继续保留在硬盘上。别人用专门的【反删除工具】，还是可以恢复出来滴；警方使用专门的“取证软件”，也可以恢复出来。

那么如何才能彻底删除文件捏？

1. 用专门的【删除/擦除工具】

目前已经有很多专门的软件，可以帮你彻底删除一个文件。这类软件在删除文件之前，会用某些特别的方式，对文件的内容进行【覆盖】，然后再删除文件。这样就可以避免文件内容被恢复出来。

2. 用【低级格式化】

很多时候，当你想把硬盘的所有数据都干掉，你可能会选择格式化硬盘（或格式化分区）。但是俺要提醒一下：“快速格式化”和“完全格式化”都无法保证覆盖整个分区的所有扇区，同样存在数据被恢复的风险。

比“完全格式化”更保险的是——“低级格式化”（简称“低格”）。

3. 【更高级】的数据删除技巧

在如下博文中，俺介绍了一些【更高级】的数据删除技巧。如果你对安全性的要求很高，如下这篇必须看。

《如何用“磁盘加密”对抗警方的【取证软件】和【刑讯逼供】，兼谈数据删除技巧》

◇尽量【别用】移动设备

俺【不】建议用移动设备（尤其是智能手机、平板之类）进行某些敏感操作。主要的原因有如下几个——

\1. （前面已经提及了）这类便携式设备，很容易丢失；
\2. 这类设备往往【不】支持“操作系统虚拟机”，不利于安全加固。
\3. 这类设备往往【不】支持专业的全盘加密软件（类似 TrueCrypt/VeraCrypt 那种）。

补充说明：

虽然如今的 Android & iOS 都已经支持“全盘加密”，但它们提供的“全盘加密”还是【比不上】专业的磁盘加密工具。

比如说：专业的磁盘加密工具，可以实现“隐藏操作系统”；而所有的移动设备都【没法】这么玩。

比如说：专业的磁盘加密工具，可以在【加密系统分区】的时候，采用“key file”作为认证因素；而所有的移动设备都【没法】这么玩。

◇尽量做到【专机专用】

如果经济条件许可，最好是专机专用——专门用一台电脑来操作你敏感的虚拟身份。在这台电脑上，【不要】有任何能关联到你【真实身份】的东西。

俺重点强调如下几种有风险的情况：

\1. 不要在任何网站（尤其是社交网站）中输入你的真实姓名、手机号、身份证号……

\2. 不要在任何软件（尤其是 IM 软件）中输入你的真实姓名、手机号、身份证号……

\3. 不要存储涉及个人信息的文件（比如个人照片、通讯簿）

\4. 不要存储任何与你的工作有关的文件

……

如果你由于条件所限，无法做到“物理电脑的专机专用”，最起码也得做到——【虚拟机】的专机专用。也就是说：你应该准备一个【专用的】虚拟机（VM），这个 VM 专门用于你的某个【敏感的】虚拟身份。在这个 VM 中【不】操作任何其它虚拟身份（当然更加【不】能操作“真实身份”）。

没用过虚拟机的同学，请看《扫盲操作系统虚拟机》系列教程。

★总结

费了好大劲，终于说清楚操作系统层面，跟数据泄露相关的防范措施。由于篇幅所限，没来得及聊“防黑客”的话题。与之相关的讨论，俺放到另一个系列《如何防止黑客入侵？》。

在本系列的下一篇，咱们来聊一聊“通讯工具的防范”。

[如何隐藏你的踪迹，避免跨省追捕4]：通讯工具的防范

文章目录

前几天，国内民主人士朱虞夫被党国关了将近一年之后，终于开庭审理。起诉的罪名是：煽动颠覆国家政权；证据是：他的 Skype 聊天记录中有一首诗——《是时候了》。看到此新闻之后，很多网友纷纷担心 Skype 的安全性问题。所以，俺今天重点来聊一聊通讯工具的防范。

★通讯工具包含哪些?

本文提及的通讯工具，主要包括：电子邮件（以下简称 Email）、聊天工具（以下简称 IM）、手机短信（以下简称 SMS）。

★通讯工具的危险性

上述这几类通讯工具，有如下的共同点：

◇群发性质

这几类工具都可以用来群发或转发。一旦你转发或群发的内容，包含有敏感的言论（哪怕仅仅是抱怨或者影射），党国都可以说你是“**煽动**颠覆国家政权”（这个罪名的亮点在于“煽动”二字）。

◇历史记录

无论是 Email、IM、SMS，都需要经过服务器中转，理论上都可以在服务器中保留历史记录。此外，某些 IM 客户端（比如：Skype）和某些 Email 客户端（比如：Outlook）还会在你的电脑上保存历史记录。

如果你的历史言论中，有过对党国不敬的只言片语，都可能成为对你不利的罪证。

★如何【选择】通讯工具?

要确保通讯工具的安全，首先要确保你用的工具是可靠的。如果你选择的工具本身就不安全，那其它一切防范措施都白搭。

◇SMS（短信）

SMS 是无论如何都不能相信滴!

毕竟天朝的3大电信运营商（中国移动、中国联通、中国电信）都在党的掌控之下。党国早就在这些运营商的短信网关中，设置了监控短信的模块。你收发的每一条短信，只要包含敏感内容，都会被发现。打个比方，2011年发起“茉莉花集会”的时候，如果你在【群发】的短信中包含茉莉花3个字，估计当天就会有六扇门的走狗去敲你家门。

◇IM（即时通讯）

一提到聊天软件，天朝的网友自然会想到 QQ。而 QQ 恰恰是【最危险】的 IM 工具。因为 QQ 的用户群实在太大了（好几个亿），党国早就在疼逊的服务器上部署了监控和过滤的软件（其原理与 SMS 类似）。只要你在 IM 聊天中涉及了太多的敏感内容，就会被盯上。

其它几款【国产】的 IM（比如：阿里旺旺、网易泡泡……），也都有这类风险，切【不可】使用。

那么，该选择哪些【国外的】IM 捏？考虑到 IM 工具的更新换代很快，俺提几个原则，供大伙儿参考：

原则1：【运营方】的选择

一般来说，非营利机构（含“开源社区”）比“商业公司”更靠谱（具体的原因及分析，参见《如何保护隐私》系列的第一篇）。

如果你因为种种原因，不得不使用商业公司的 IM，选一个【口碑足够好】的商业公司的 IM。

原则2：是否【开源】?

一般来说，“开源的 IM”比“闭源的 IM”更靠谱。其中一个原因是：“源代码公开”有利于【代码审计】，防止 IM 软件暗藏后门。

原则3：是否依赖【手机号】？

这里所说的“依赖”至少包括两种：其一，IM 绑定手机号；其二，注册 IM 时需要用手机短信进行验证。只要有这两者之一，都算是“依赖”。

尽量使用那种【无需】依赖手机号的 IM 工具。道理很简单——

一旦 IM 工具依赖手机号，也就意味着——IM 的运营方可以拿到你的手机号。对大部分网民而言，通过“手机号”很容易就可以定位到【自然人身份】。

如果你因为种种原因，不得不用某款依赖手机号的 IM 工具，你要【确保】——该手机号与你的自然人身份【完全无关】（比如说，你可以购买那种【不记名】的手机卡，并且在购买时用【现金】支付）。

原则4：是否支持【桌面】操作系统?

尽量使用那种可以运行在桌面操作系统的 IM 工具。

为啥捏？（相比手机/平板而言）桌面操作系统可以进行更好的安全加固（比如说：基于【虚拟机】进行隔离）

原则5：是否提供【端到端加密】？

“端到端加密”，洋文称之为“End-to-End Encryption”，简称 E2EE。通俗地说就是：确保聊天双方的数据传输是【全程加密】。

显然，有这个功能的 IM 更靠谱——由于聊天内容的传输是【全程加密】，因此 IM 服务器在中转聊天信息的时候，看到的全是【密文】。这就避免了“IM 服务器偷窥聊天内容”。

原则6：是否【免费】？

很多人总是觉得：“付费”的东西更好。

但对于【隐匿性】这个领域而言，“付费”是一个很大的风险点。因为“付费环节”（不论是“线上 or 线下”）都很容易暴露身份信息。

原则7：【架构】的选择

IM 的运作包括不同的架构，常见的有：中心式、联邦式、点对点式（P2P）。

尽量选择后两种——有助于对抗政府的审查与封锁。

◇Email（电子邮件）

Email 跟 IM 类似，【不】要用【国内】邮件服务商提供的邮箱（比如：网易、新浪、搜狐、腾讯、等等）。一旦你用了这些邮箱，今后你往来的每一封邮件，都有可能被党国监控和审查。

既然国内邮箱【不】靠谱，那么国外的邮箱是否就足够安全捏？也不一定哦！

2004年出过一个很轰动的“师涛案”，估计有些网友还记得吧？（不知道此事的网友，请看“这里”的介绍）师涛就是因为用了雅虎邮箱而倒霉的。当年国安局的人找雅虎交涉，让雅虎交出师涛邮箱的信息。据说国安局尚未施加压力，这个没骨气的雅虎就乖乖交出来了。师涛因此被判10年监禁。

所以，光是国外的邮箱还不够，还得找【有骨气】的机构（此处所说的“机构”包括“商业公司 or 非营利组织”）。

★常见的【安全隐患】

前面介绍的，都是如何选择靠谱的通讯工具。但是，光选对工具还不够，还得注意一些潜在的安全隐患。

◇自动登录的隐患

此次的朱虞夫案，党国爪牙之所以能拿到 Skype 的聊天记录，据说是因为朱虞夫的 Skype 设置为自动登录。由此可见，自动登录是一个潜在的隐患。如今，不光聊天工具可以自动登录，邮箱也可以自动登录。在这种情况下，万一你的电脑被党国缴获，你的邮件内容和聊天记录，就立刻就曝光了。

要避免此问题，有两种解决方法：其一，不使用自动登录；其二，使用加密存储。

不使用自动登录的话，每次都要输入口令（假如同时用几个邮箱或 IM，就得输入多次口令），估计大部分人会嫌麻烦。所以，“加密存储”属于即安全又方便的做法。加密存储当然也要输入口令，但是只需输入一次（具体如何搞，后面会说）。

◇本地存储的隐患

除了自动登录，另一个安全隐患是【本地存储】。很多 IM（比如：Skype、MSN）的聊天记录是存储在电脑本机的。像 MSN 的聊天记录甚至是不加密的。一旦朝廷拿到你的电脑，你全部的聊天历史都会曝光。

另外，有不少人喜欢用 Outlook 并把邮件同步到本地。如果这么干，也会面临跟“聊天历史”一样的安全风险。

那该咋办捏？还得靠【加密存储】来搞定！这也是下一个章节的主题。

★如何【加密存储】？

既然上述两大隐患都可以用加密存储来解决，俺自然要说说具体的加密方法和操作步骤。

◇加密工具的选择

说到“加密本地数据”，TrueCrypt（简称 TC）实在是不二之选。关于它的优点及功能介绍，请看俺专门写的扫盲帖（在“这里”）。简而言之，TrueCrypt 是目前最靠谱的文件加密工具。据说老美的 FBI，如果碰上嫌犯使用 TrueCrypt 加密数据，也是束手无策。

补充说明：

本文写于2010年；到了2014年的时候，TrueCrypt 突然停止开发，于是俺又介绍了它的两款替代品，分别是 VeraCrypt 和 dm-crypt（LUKS）。这两款工具都可以完美兼容“TrueCrypt 的加密盘【格式】”。

换句话说：虽然 TC 这个【软件】已经死了，但是 TC 使用的【加密盘格式】，将长期存在。

◇哪些文件需要加密?

对于 Email 而言

前面俺已经建议过，【不要】把邮件同步到本地。如此一来，你只需解决邮箱“自动登录”的问题。

几乎所有的 Web 邮箱（网页邮箱），都是依靠“cookie”来实现自动登录滴。主流浏览器（包括：IE、Firefox、Chrome、Edge）的 cookie 文件存放于当前用户的【`%APPDATA%` 目录】之下（在资源管理器的地址栏输入 `%APPDATA%`，再敲回车，便可进入到该目录）。

上述所说，针对的是 Windows 系统。如果你用的是 POSIX 系统（Linux & MacOS），“cookie 文件”默认都位于当前用户的【HOME 目录】下。

对于 IM 而言

很多聊天工具保存“自动登录口令”以及“聊天历史”的那些文件，也是在【`%APPDATA%` 目录】之下。

上述所说，针对的是 Windows 系统。如果你用的是 POSIX 系统（Linux & MacOS），聊天工具保存“自动登录口令”以及“聊天历史”的那些文件，默认都位于当前用户的【HOME 目录】下。

◇如何实现加密?

对于 Windows 用户

从上一个小节的介绍可以看出，那些敏感的文件都位于【`%APPDATA%` 目录】之下。一般来说，`%APPDATA%` 目录所在的盘就是系统盘（也叫“系统分区”）。因此，你用磁盘加密软件（TrueCrypt 或 VeraCrypt）【加密系统盘】，即可确保安全。

加密完系统盘之后，你每次开机，都需要先输入“系统盘解密口令”。口令正确，操作系统才能启动起来。如果没有口令，别人即使拿到你的硬盘，也甭想看到系统盘中的任何文件（自然也就看不到“`%APPDATA%` 目录”）。

对安全性要求更高的同学，可以考虑加密【整个硬盘】（俗称“全盘加密”）。当你做到了“全盘加密”，“系统分区”当然也被加密了。

对于 Linux 用户

从上一个小节的介绍可以看出，那些敏感的文件都位于【HOME 目录】之下。俺建议把 `/home` 单独分一个区，然后用 dm-crypt（LUKS）加密该分区。相关配置参见“这篇教程”。一旦加密了整个 `/home` 分区，每个用户的“HOME 目录”自然也被加密了。

对安全性要求更高的同学，可以考虑加密【整个硬盘】（俗称“全盘加密”）。当你做到了“全盘加密”，`/home` 目录当然也被加密了。

◇注意事项

\1. 再啰嗦一次，“TC 加密格式”很牛逼——口令丢了，神仙也救不了你。
\2. 如果你是【头一次】用 TrueCrypt 或 VeraCrypt 或 dm-crypt/LUKS，在加密前，务必先做好备份工作，以防配置错误，把数据搞丢了。
\3. “加密系统盘 or 全盘加密”，对性能会有一定影响。影响程度多大，取决于你选择哪种加密算法。
\4. 已经使用磁盘加密的电脑，当你不用电脑的时候，要【关机】，而不要使用“待机”，也不要使用“休眠”（具体原因参见如下“引申阅读”）

引申阅读：

本文发布于2012年；之后的几年，俺又陆续写了一些与“磁盘加密”相关的教程（如下）：
《TrueCrypt 使用经验》（系列）
《扫盲 VeraCrypt——跨平台的 TrueCrypt 替代品》
《扫盲 dm-crypt——多功能 Linux 磁盘加密工具（兼容 TrueCrypt 和 VeraCrypt）》
《如何用“磁盘加密”对抗警方的【取证软件】和【刑讯逼供】，兼谈数据删除技巧》

★虚拟机的【局限性】

看到有不止一位网友在本帖留言，建议用虚拟机解决安全隐患。所以，俺额外补充这一章节。

◇无法防范“服务端的历史记录”

即便用了虚拟机，如果你依旧使用国内的 IM 工具或电子邮箱，那你的邮件历史和聊天历史还是会被保存在服务器端。如此一来，党国的爪牙就能拿到这些历史信息，从而有助于追踪你的身份。

◇无法彻底解决“加密问题”

有些同学可能会考虑用 EFS（加密文件系统）来解决加密问题。但是其安全性【未必】能达到要求。请看俺的分析——

打个比方，假如你在虚拟机中安装了 Windows，然后在 Windows 中装了 Skype。为了保密，你把 Skype 的相关目录用 EFS 加密。你以为这样就安全了？非也！

一旦你的电脑被党国缴获，专业的安全取证人员不难发现你在用虚拟机。然后把你的虚拟机文件找出来，对其中的系统用户进行分析。在刚才的例子中，如果你仅仅用 EFS 加密 Skype 相关目录，那安全取证人员只需攻破你的 Windows 用户口令，即可拿到你加密目录中的所有文件。要知道，攻破 Windows 用户口令，可比攻破 TrueCrypt 容易多了。

◇难以隐藏“虚拟硬盘文件”

对于大部分虚拟化软件，虚拟系统（Guest OS）中的虚拟硬盘通常对应于真实系统（Host OS）中的某个单独文件（该文件称之为“虚拟硬盘文件”）。比方说，你虚拟出来的系统有2个 4GB 的虚拟硬盘，那你的电脑中，通常也会有2个接近 4GB 的大文件。另外，真实系统中必然装有相应的“虚拟化软件”（比如：VMware 或 VirtualBox）。

有经验的警方取证人员，一看到你电脑中装有 VMware 或 VirtualBox 这类软件，就会在你的硬盘上搜寻虚拟机的“虚拟硬盘文件”。这类文件通常都有几个 GB，很好找。

不要跟俺说，你用了隐藏目录之类的把戏——那种把戏只能骗骗技术菜鸟。

◇小结：“虚拟机”应该结合“磁盘加密”

综上所述，为了安全起见，【不能】仅仅依靠虚拟机来作为防范措施。真要用虚拟机，还得配合【磁盘加密软件】一起用（把虚拟机放到【加密盘】中），才够安全。

[如何隐藏你的踪迹，避免跨省追捕5]：用多重代理隐匿公网 IP

文章目录

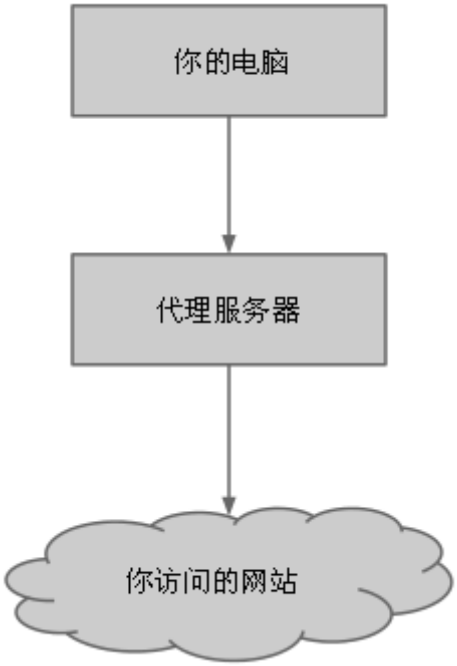
最近，天朝的两会即将通过新版本的刑事诉讼法（在新条款中，党国的爪牙可以【**合法地**】进行秘密拘捕——太阴险啦，良心大大滴坏！）所以，俺要继续完善“如何隐藏踪迹”系列，帮助大伙儿躲避朝廷的网络追捕。

关于多重代理，在前几个月的翻墙贴《扫盲 VPN 翻墙——以 Hotspot Shield 为例》中，稍微提到过。今天来完整地介绍一下。

★啥是“多重代理”?

先声明一下：本文所说的“代理”是广义滴——既包括“传统意义上的代理”，也包括“VPN”。

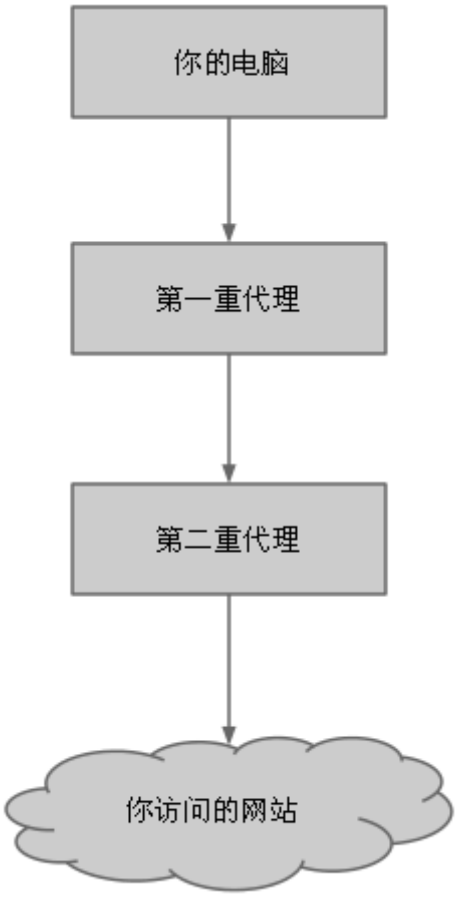
平时咱们用代理来翻墙，大部分属于【单重】代理（如下图）。也就是说，不论你用的是 VPN 还是 HTTP Proxy 还是 SOCKS Proxy，当中都只有【一个】服务器进行中转。



单重代理可以（在一定程度上）保护你的隐私，防范跨省追捕。关于这点，俺在本系列的第1篇《网络方面的防范》已经介绍过了。但如果你对隐私的防范，要求比较高（比如说，你是六扇门的重点关注对象），那单重代理的安全性就【不够】啦！为了进一步增加安全性（隐匿性），你需要使用【多重】代理。

那“多重代理”是啥样的捏?

严格来讲，中转次数【超过1次】的，都可以算“多重代理”。为了简单起见，俺画了下面这幅“双重代理”的示意图。



★需要哪些软件?

◇Tor + 其它翻墙工具

理论上，你可以随便挑选两款翻墙工具，然后搞出一个二级代理。但是这样的效果未必理想。

根据俺的经验，最佳组合是：用 Tor（俗称"套"）搭配其它的翻墙工具（比如：赛风、无界、自由门、VPN...），组合出多重代理。

这种玩法称之为【戴套的多重代理】，与 Tor 配合的另一个翻墙工具，称作"Tor 的【前置代理】"。

俺要强调一点：作为 Tor 的"前置代理"，本身必须是【加密】代理（有助于提高安全性）。

◇为啥要"戴套"?

长期翻墙的网友，应该都听说过 Tor 这个老牌的翻墙工具（俺曾经扫盲过"戴套翻墙"的技术）。那些从来没听说过 Tor 的网友，可以看维基百科的"这个页面"。

其实拿 Tor 来翻墙，颇有杀鸡用牛刀的嫌疑。Tor 的主要强项在于：**提供匿名的网络访问，保护你的隐私。**比如名气很大的维基解密（WikiLeaks），还有名气很大且很牛逼的匿名黑客组织（洋文叫"Anonymous"，最近连续黑掉多个大公司及美国政府部门），他们的成员都是用 Tor 来确保自己的匿名。

为啥 Tor 能确保匿名捏?

其一，

Tor 在全球有很多节点，当你利用 Tor 上网的时候，从你的电脑到某个网站，需要经过若干个 Tor 节

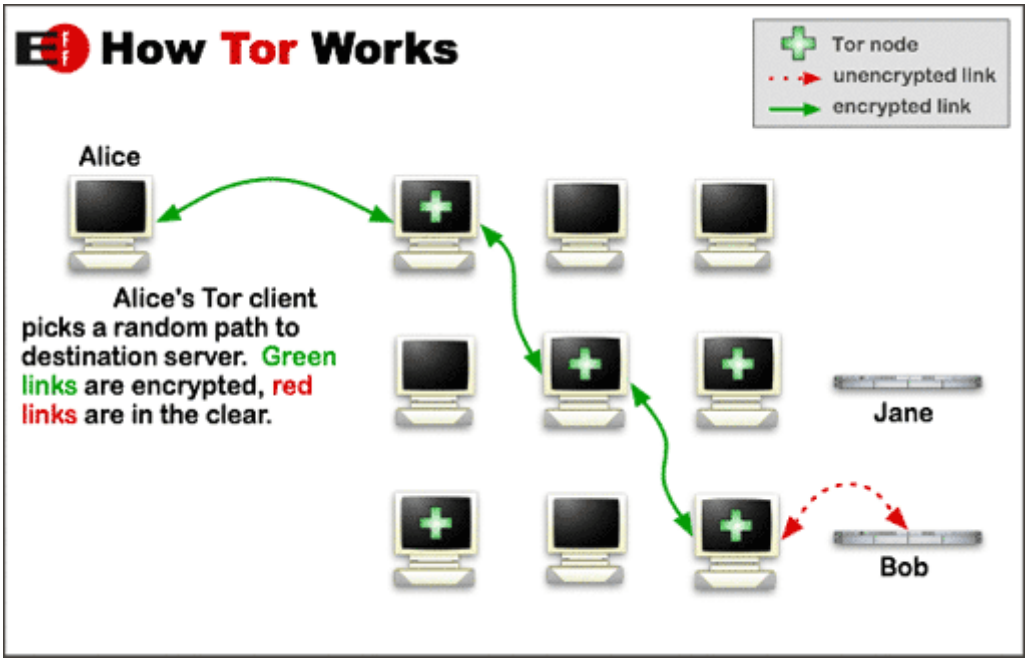
点（通常是3个节点，俗称“三级跳”）。

其二，

第1个节点虽然知道你的公网 IP，但【不】知道你访问了啥网站；第3个节点虽然知道你访问了啥网站，但【不】知道你的公网 IP；至于第2个节点，既【不】知道你的公网 IP，也【不】知道你访问了啥网站。

其三，

上述的“三级跳”线路是【动态变化】滴（通常每隔10分钟变化一次），这就使得警方难以逆向追溯。



（Tor 网络的示意图）

◇为啥要给 Tor 加【前置代理】

假如你有些悟性，看完上述示意图之后就会发现：Tor 本身就是一个多重代理！既然这样，为啥还要拿 Tor 跟其它翻墙工具搭配捏？如下有如下几个原因——

其一，

因为万恶的 GFW 把全球大部分的 Tor 节点都列入“IP 黑名单”。因此，如果你不幸身处天朝，是很难直接访问到 Tor 节点滴！【前置代理】可以帮助你本机的 Tor 客户端接入全球的 Tor 网络。

其二，

即使你身处【墙外】，俺依然建议：使用【Tor + 前置代理】。因为你还要考虑到——万一 Tor 软件出现某个致命的安全漏洞，可能会导致 Tor 的“匿名化措施”失效。而使用“Tor + 前置代理”，就类似于【双保险】。

其三，

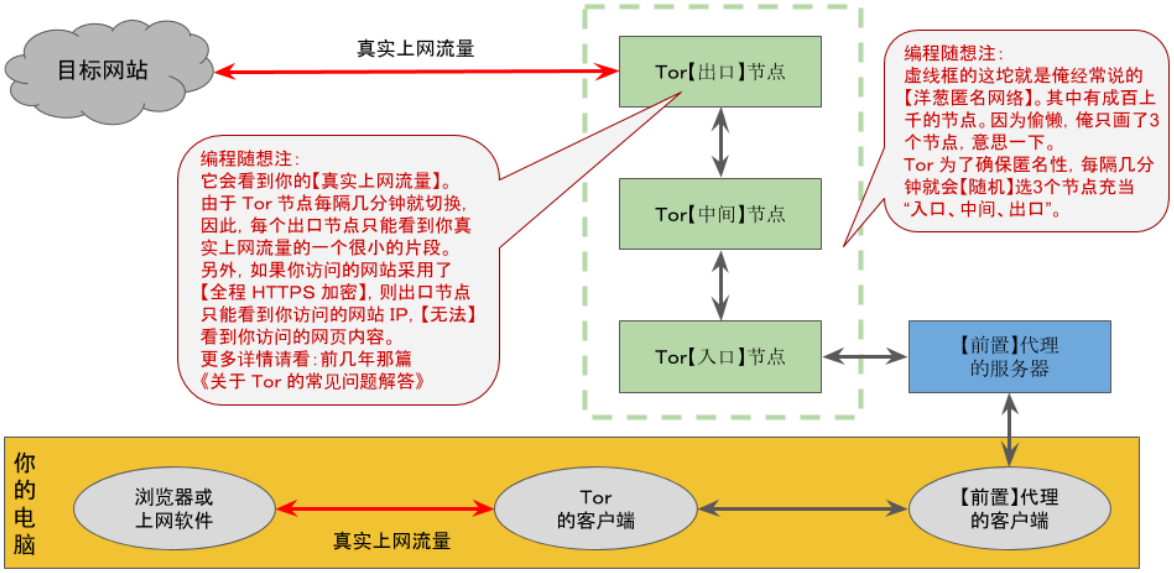
对于安全性要求很高的用户，你还要做到——既使用 Tor，又不要让 ISP 及警方发现你在使用 Tor。因为 Tor 用户毕竟还很【小众】，更容易引起怀疑。当你采用【Tor + 前置代理】的方式，即使 ISP 或警方监视了你的流量，看到的是【前置代理】的流量，看【不】到 Tor 的流量。

（注：前面说过——“前置代理”必须是“加密代理”。因此，“Tor 流量”被包裹在“前置代理”的【加密】流量内部）

◇示意图

为了让大伙儿有个直观的印象，放一张示意图（如下）。这张图来自俺的另一篇博文《如何用 Tor 访问对 Tor 不友好的网站——扫盲“三重代理”及其它招数》。

【双重代理】数据流示意图



（【双重代理】数据流示意图）

★如何配置?

其实配置并不难，只需如下几步：

◇第1步：运行翻墙工具（前置代理）

首先，你需要准备好一个【能用的】翻墙工具——可以是 HTTP 代理（比如：无界、自由门、赛风），也可以是 SOCKS 代理，还可以是 VPN（比如 VPN Gate）。先把这个翻墙工具运行起来，它用作 Tor 的【前置代理】。

◇第2步：安装 Tor

如果你使用 Linux 并且发行版的官方仓库已经包含了 Tor，那么就从官方仓库直接安装。
否则的话，请【翻墙】到 Tor 的官方网站，下载一个 Tor 的软件包（下载页面在“这里”）。
目前官网上提供两种软件包，分别是：
“Tor Browser Bundle”（面向傻瓜用户）
“Expert Bundle”（俗称“裸 Tor”，面向高级用户）

◇第3步：配置 Tor

如果你下载了“Tor Browser”，参见如下这篇教程：
《“如何翻墙”系列：扫盲 Tor Browser 7.5——关于 meek 插件的配置、优化、原理》
如果你下载了“Expert Bundle”，并且你的操作系统是 Linux/UNIX，参见如下这篇教程：
《扫盲 Arm——Tor 的界面前端（替代已死亡的 Vidalia）》

◇第4步：设置浏览器

（刚才说了）Tor 有两种软件包（Bundle）。这两者的【监听端口号】略有差异。因此在配置浏览器时，也略有差异。分别说明如下：

Tor Browser Bundle
Tor 的监听端口为 `9150`
由于这种软件包已经内置了 Firefox，且内置的 Firefox 已经绑定到 Tor。因此，你【无需任何设置】。

Expert Bundle（裸 Tor）
Tor 的监听端口为 `9050`
对于这种软件包，你要把浏览器的【SOCKS 代理】指向 Tor（地址 `127.0.0.1` 端口 `9050`），就可以通过 Tor 匿名上网了。

◇第5步：设置【其它】上网软件

如果你想让其它的第三方网络软件（比如：聊天软件、下载工具）通过 Tor 来隐藏你的公网 IP，也很容易。参见刚才浏览器的代理设置——SOCKS 代理，地址 `127.0.0.1`，端口号使用 `9050`（裸 Tor）或 `9150`（Tor Browser）。以 MSN 为例，你只需在 MSN 的网络设置界面填写 Tor 的 SOCKS 代理即可。

◇第6步：测试 & 验证

为了保险起见，你还需要做一些测试，验证一下你是否真的通过 Tor 上网。

对于浏览器
用浏览器访问 Tor 官网的“检查页面”（在“这里”）；这个页面会告诉你，当前是否已经通过 Tor 访问。

对于【其它】上网软件
需要到“Tor 的管理界面”观察一下 Tor 的网络流量【变化】。当你使用这些第三方软件的时候，Tor 的管理界面如果显示有流量，就说明这些网络软件已经通过 Tor 联网了；反之（如果使用第三方软件时，Tor 的管理界面没有显示流量），则说明你配置有误——第三方软件【没】通过 Tor 联网。

★多重代理的【好处】

◇防范追踪

举个例子：

假设你用【单重】的 VPN 翻墙并发表一些抨击党国的言论。万一 VPN 提供商在 VPN 服务器上保存了你的上网日志，而党国又逼迫该 VPN 供应商交出这些日志。那么，党国的爪牙就【有可能】分析出你的上网行为。

用了多重代理之后，任何一个代理服务器记录你的网络流量，都无法对你的流量进行分析。

举例如下：

假设你用的是“Tor + 赛风”。虽然“赛风服务器”知道你的真实公网 IP，但是无法知道你访问哪个网站，也不知道你访问的的内容（因为 Tor 的流量是加密滴）；而 Tor 的“最后一个节点”（出口节点）虽然知道你访问了哪个网站以及访问的内容，但是它不知道你来自哪里（不清楚你的真实公网 IP）。

◇伪装国籍

除了上述好处，使用 Tor 还有另一个好处——伪装国籍。比方说，你想让自己看起来像是美国的用户，只需要修改 Tor 的配置文件，使用美国境内的【出口节点】。这种情况下，你访问的网站就会以为你来自美国。

★多重代理的【坏处】

当然啦，有利就有弊。以下是多重代理的一些缺点：

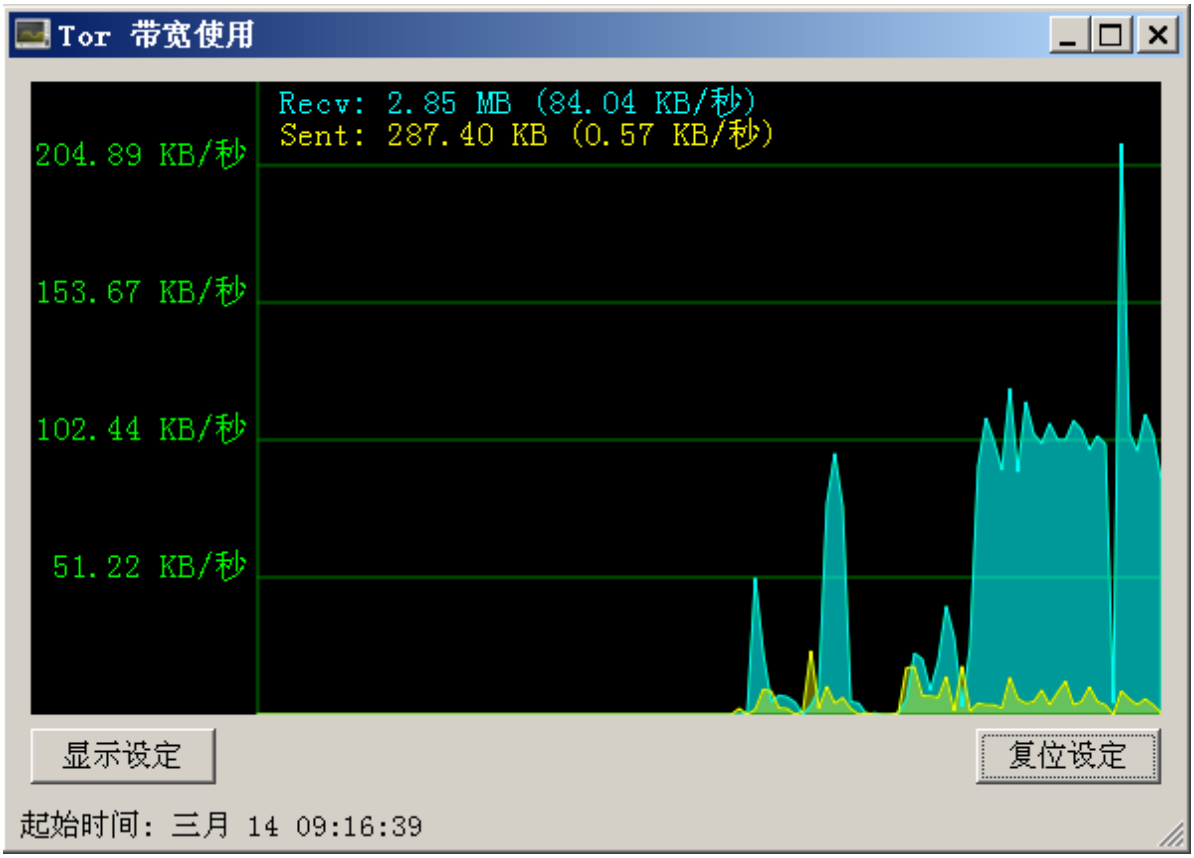
◇配置复杂

跟“单重代理”比起来，“多重代理”显然需要更多的设置。很多网友属于技术门外汉，多半对它望而却步。所以，俺才会专门写这么一篇博文，扫盲多重代理。

◇性能下降

通常来说，多重代理的性能会比单重代理要差一些。

根据俺近几年的经验——只要【前置代理】的速度足够快，“前置代理 + Tor”的速度也慢不到哪里去。以下是俺电脑上的 Tor 流量截图（基于“赛风+Tor”）。



[如何隐藏你的踪迹，避免跨省追捕6]：用虚拟机隐匿公网 IP（原理介绍）

文章目录

在多位读者的强烈要求下，本文终于隆重推出了 :) 今天俺来介绍一下：如何利用操作系统虚拟机（以下简称“虚拟机”）来隐匿自己的公网IP。

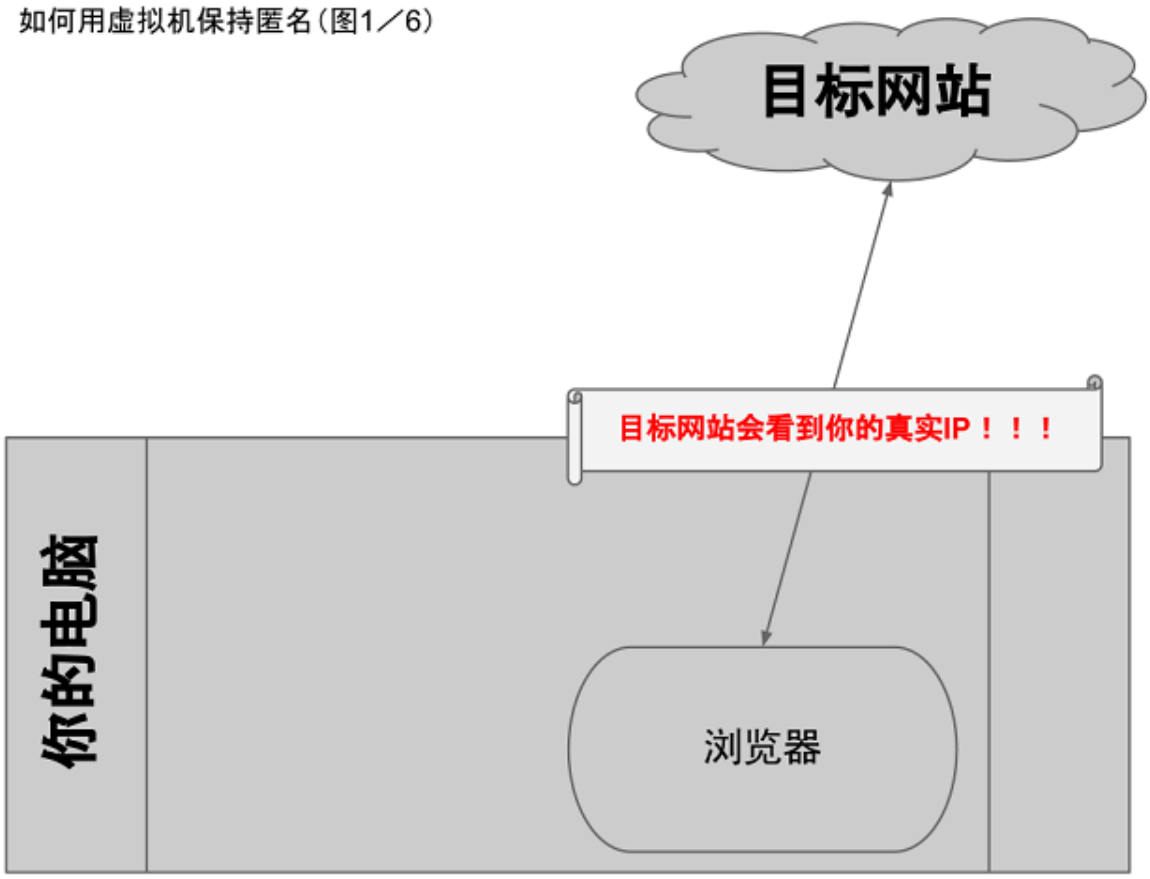
★准备工作

为了给本文打基础，俺在最近两个月特地写了一个《扫盲操作系统虚拟机》的系列博文（链接在“这里”）。如果你不是 IT 技术人员，或者你对虚拟机了解不多，**强烈建议**你先把这个虚拟机扫盲系列看完，然后再来看本文。

在本系列第一篇《网络方面的防范》，俺专门解释了：“什么是公网 IP”，也特别强调了“公网 IP 暴露”的危险性。如果你比较健忘的话，先去把那篇再复习一下。

★无代理的情形

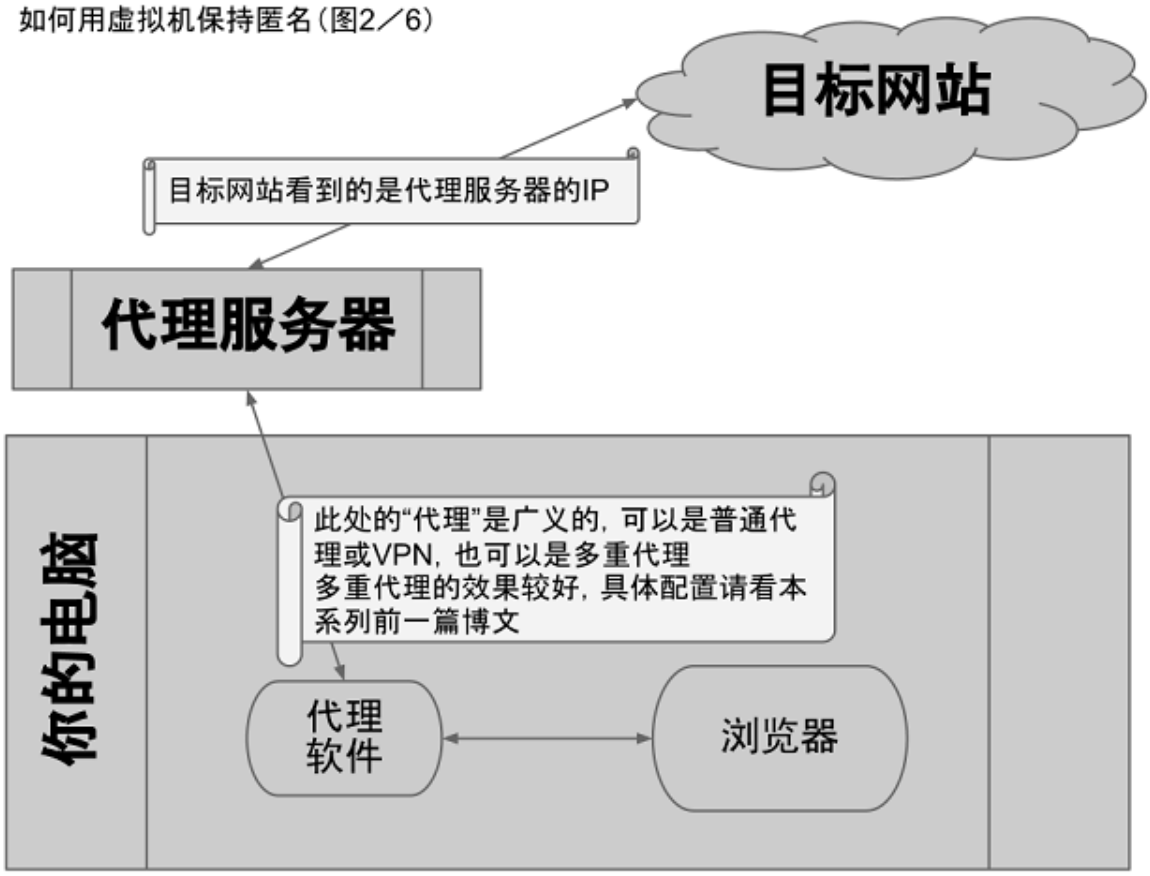
如果你上网不使用代理，那么你访问的网站就会看到你的公网 IP。示意图如下：



★普通代理的情形

此处说的代理是广义的，包括普通代理、VPN、多重代理、等。如果用了代理，你访问的网站看到的公网 IP 实际上是代理服务器的 IP。

示意图如下



但是不要高兴得太早，仅仅依靠普通的代理，是远远不够滴！请看下面的例子。

★Flash 如何暴露你的【公网 IP】？

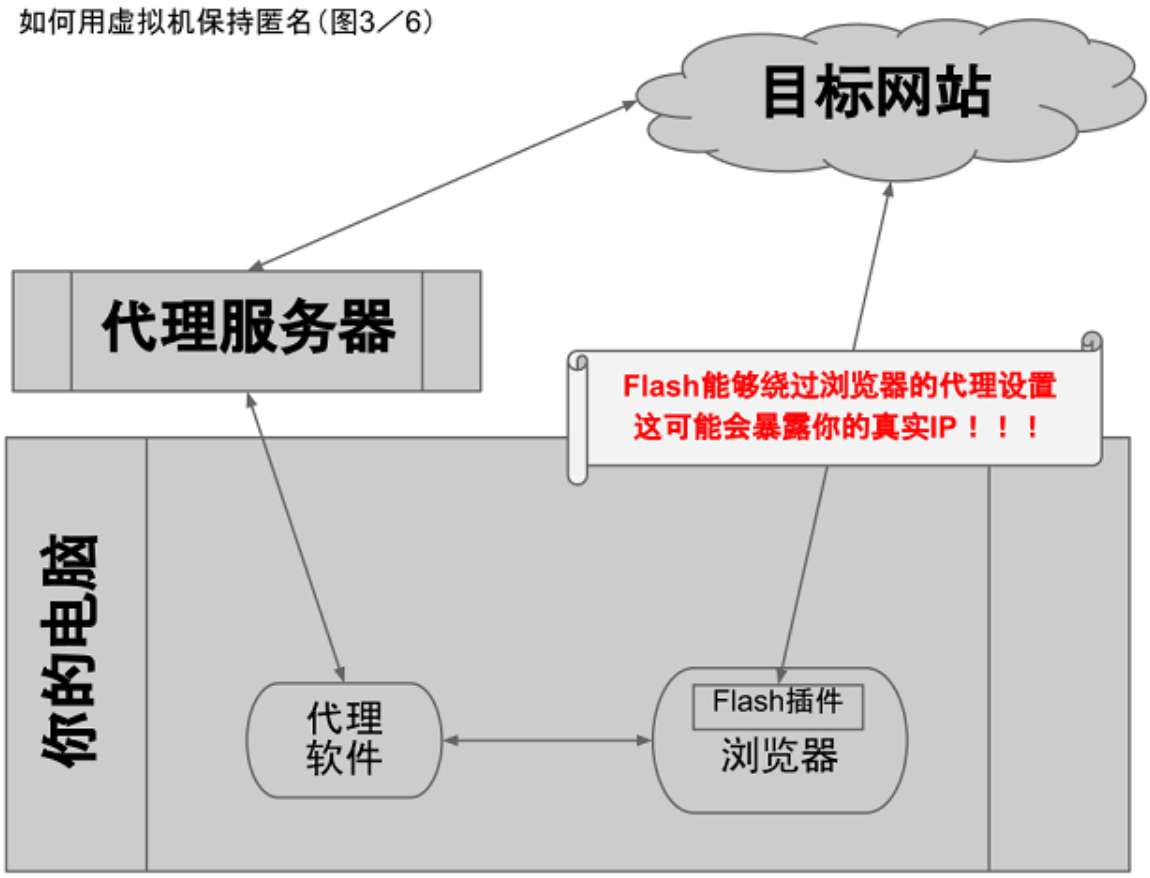
比如你在浏览器中设置了代理，那么浏览器确实会通过代理来加载网站的页面。**但是**，浏览器如果安装了某些插件（最典型的是 Flash），这些插件是可以做到绕过代理，直接去访问目标网站的。

举个例子：

比如你想在某个国内的网盘上传政治敏感文件。因为这个网盘是国产的，你担心网盘服务器会记录你的公网 IP。所以捏，你设置了浏览器的 HTTP 代理，通过翻墙代理来访问这个网盘的页面。

如今的很多网盘是利用 Flash 插件进行文件上传的。而且 Flash 插件在上传文件的时候，往往是不经过浏览器的 HTTP 代理，**直接连接**网盘服务器。如此一来，你的公网 IP 还是暴露了 :(

示意图如下

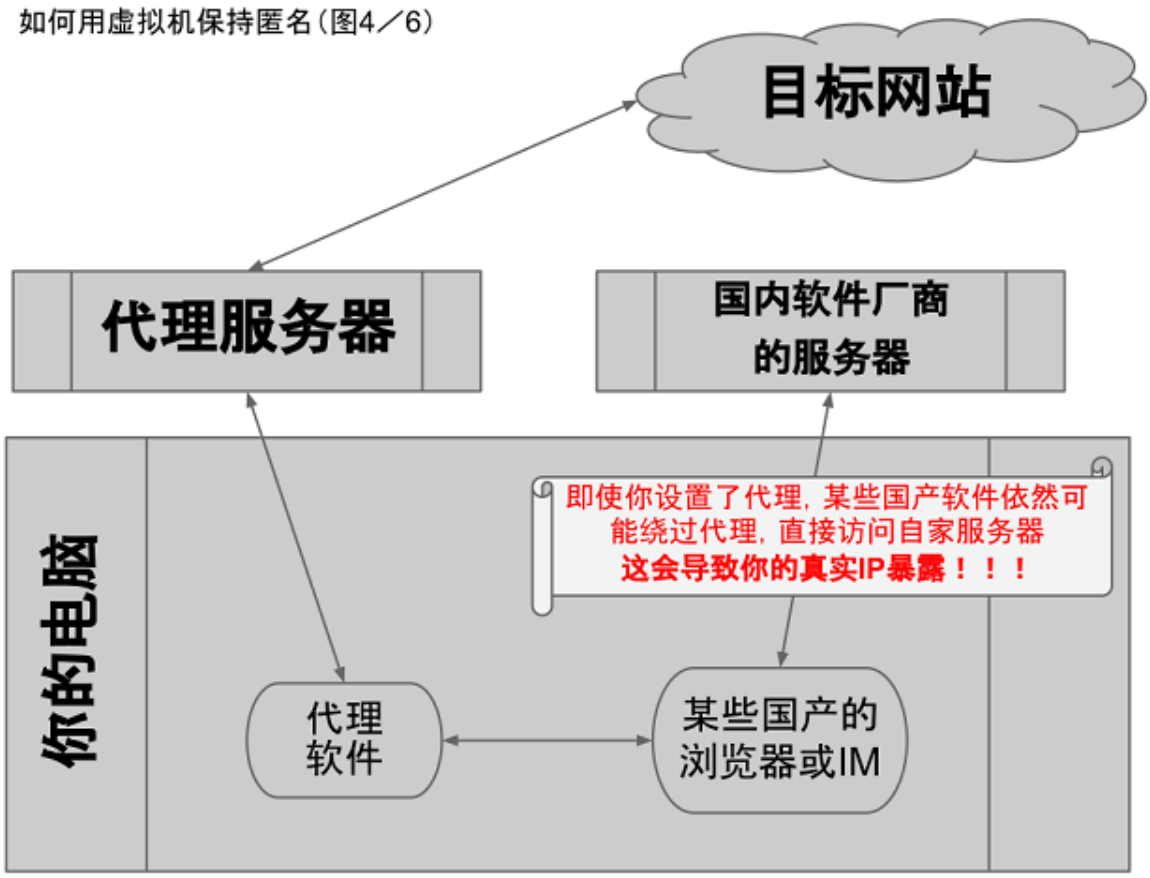


★某些国产软件如何暴露你的【公网 IP】？

除了浏览器插件（以 Flash 为主）会暴露你的公网 IP，还有其它很多国产的网络软件也会暴露你的公网 IP。比较典型的有 QQ、腾讯浏览器、360浏览器、迅雷、等等。关于危险的国产软件，本系列第2篇《个人软件的防范》已经聊过，这里就不再浪费口水了。

上述这些网络软件虽然都提供了代理的功能。**但是**，即使你设置了代理，这些软件依然有可能在后台，悄悄地访问自己公司的服务器，传输某些不可告人的信息。而且这些软件访问自家的服务器，往往是绕过代理设置，**直接连接**——这就会导致你的公网IP暴露。

示意图如下



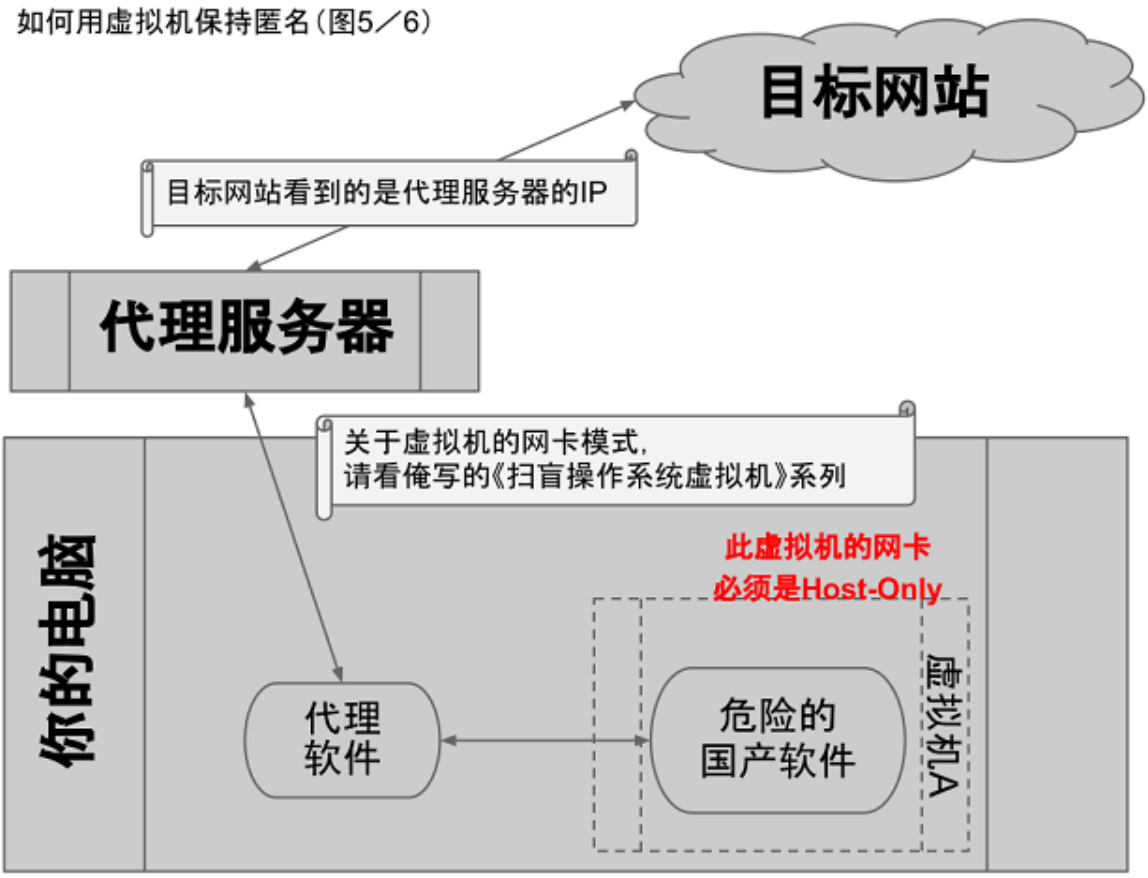
★【单】虚拟机的方案

那么，如何避免上述这几种危险的情况捏？首先来看一下“单虚拟机”的方案。

咱们可以把那些有危险的软件安装在虚拟系统（虚拟机A）内，然后把该虚拟唯一的虚拟网卡设置为【**Host-Only**】模式（关于网卡模式的设置，请看“这里”）。由于唯一的虚拟网卡是【**Host-Only**】模式，所以 虚拟机（Guest OS）内的任何软件都**不可能【直接访问】外部网络。**

那么，如何让这些危险的软件联网呢？你可以在 Host OS 里面安装一个代理软件。虽然这个 Guest OS 无法访问外网，但还是可以访问 Host OS 的，所以也就可以连接到 Host OS 里面的代理软件。然后你设置这些危险软件的 HTTP 代理或 SOCKS 代理，让它们通过代理连接到互联网。

示意图如下



★【双】虚拟机的方案

说完“单虚拟机方案”，再来说说“双虚拟机方案”。

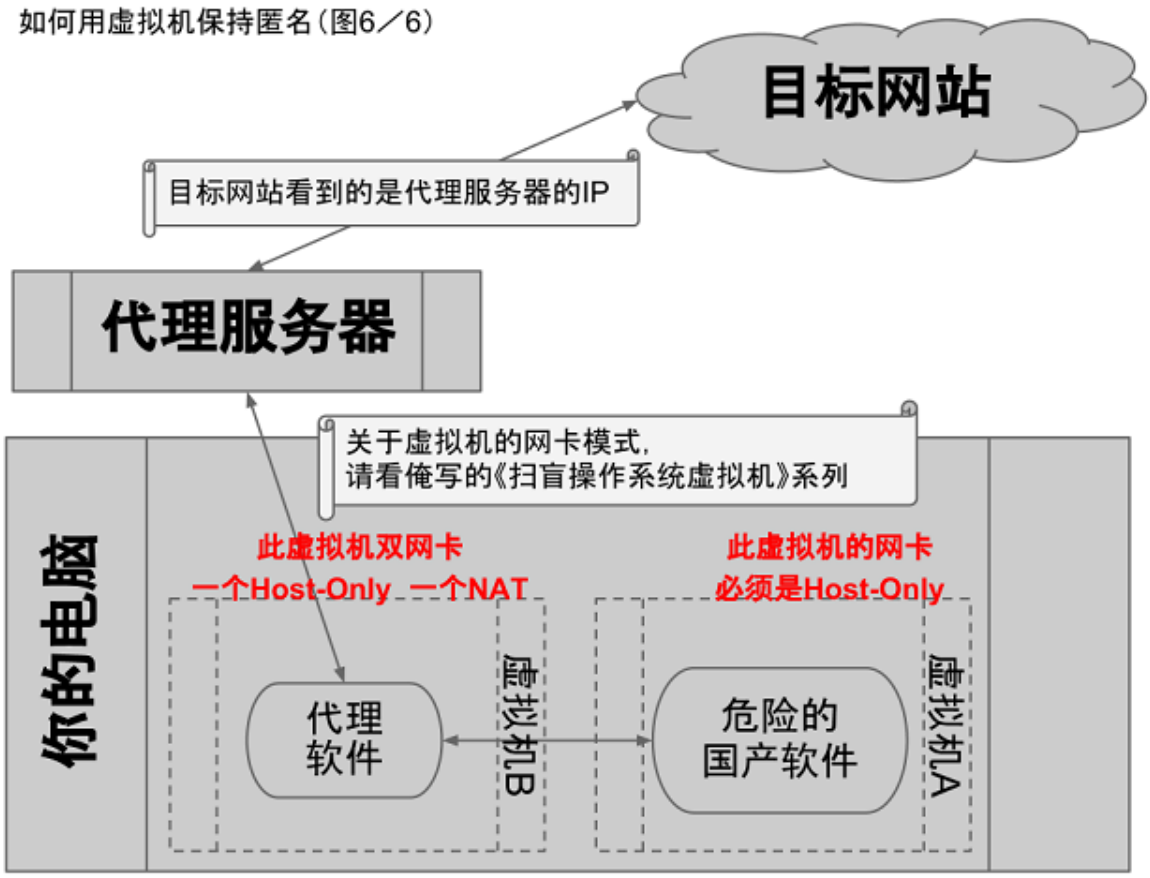
对很多网友而言，单虚拟机已经足够了。那么为啥俺还要介绍双虚拟机捏？主要有如下几方面考虑

1、某些翻墙代理软件不是开源的，有些网友对这类软件不放心

2、某些网友的 Host OS 不是 Windows，但是很多翻墙代理是 Windows 软件

所以，咱们可以在“单虚拟机”的基础上再扩展一下，把代理软件也放到虚拟机中（下图的虚拟机B）。虚拟机B 的网卡很有讲究滴——**必须是双网卡**。一个网卡设置为 Host-Only 模式，以便跟虚拟机 A对接；另一个网卡设置为 NAT 模式，以便访问外网。

示意图如下



★结合【多重代理】

在本系列的前一篇，俺介绍了“用多重代理保持匿名”。今天介绍的这2个招数都可以跟多重代理搭配

打个比方，假设你用“Tor + 自由门”来构造多重代理。对于单虚拟机的方案，你把“Tor、自由门”都安装到 Host OS 即可；对于双虚拟机的方案，你把“Tor、自由门”都安装到 虚拟机B 即可。

★结尾

关于“虚拟机结合代理来隐匿公网IP”，基本原理就介绍到这里。本系列的下一篇，俺写一个傻瓜化的配置教程，附上详细的截图，告诉你如何操作。

[如何隐藏你的踪迹，避免跨省追捕7]：用虚拟机隐匿公网 IP（配置图解）

文章目录

4天前发布了《用虚拟机隐匿公网 IP》，很多读者到博客留言，抱怨说讲得太简单。缺少傻瓜化的、带截图的配置教程。俺恭敬不如从命，今天再发一篇《用虚拟机隐匿公网 IP（配置图解）》，竭尽所能，写得尽量傻瓜化。

为了以示区分，前一篇的标题修改为《用虚拟机隐匿公网 IP（原理介绍）》。

顺便说一下：

前面那篇发到 G+ 上，貌似被顶到 G+ 热门榜，转发数很多。看来有不少网友关注"隐匿身份"这个话题 :) 所以俺今后会多花点时间，普及这方面的相关常识。

★准备工作

◇思想上的准备

首先，如果你对虚拟机软件不太熟悉，**【一定要先看完】**俺之前写的《扫盲操作系统虚拟机》。

其次，再把本系列的前一篇博文（链接在"这里"）**认真看完。**那篇博文是介绍"虚拟机隐匿公网 IP"的原理，配有精美示意图 :)

◇虚拟机软件的准备

既然要用虚拟机隐匿公网 IP，当然要先把虚拟机软件准备好。假如你不晓得该选哪种虚拟机软件，请看俺之前的一篇博文（请翻墙看"这里"），专门介绍虚拟机软件的选择。

本文主要拿 VMware Workstation（以下简称 VMware）和 VirtualBox 来介绍。

◇代理软件的准备

先声明一下，本文提到的"代理"一词是广义的，包括：普通代理、VPN、多重代理。

由于本教程是拿虚拟机跟代理软件进行组合搭配，所以你还得懂得使用代理软件。这点应该不难——只要你有翻墙的经历，你就已经在同"代理软件"打交道了。如果你从来没玩过翻墙，请先学习俺博客上的诸多翻墙教程（包括：TOR、I2P、赛风、世界通、自由门、无界 .....）。

关于代理的类型，俺强烈建议用**多重代理**（教程在"这里"，需翻墙）。为啥捏？如果你对隐匿性的要求不高，根本都不需要看本教程。你来看本教程，就说明你对隐匿性有较高的要求。既然如此，当然要用多重代理啦——这可以大大增加逆向追踪的难度。

注意事项：

要确保你用的代理软件，是监听在 `0.0.0.0` 地址，而不是监听在 `127.0.0.1` 地址。如果代理软件只监听在 `127.0.0.1` 地址，那么其它虚拟机的网络软件是【无法】连接到这个监听端口滴！

如何查看代理软件在哪个地址上进行监听捏？

用如下命令，就可以看到当前系统中开启的【所有】监听端口以及该监听绑定的地址。

注：前一个命令用于 Windows；后一个命令用于 POSIX 系统（Linux & UNIX）

```
netstat -an | find "LISTEN"
netstat -an | grep "LISTEN"
```

那么，万一你的翻墙工具的监听端口【没】绑定到 `0.0.0.0` 该咋办捏？别担心，俺后来又专门写了一篇教程《多台电脑如何共享翻墙通道》，教你如何解决监听端口绑定地址的问题。

如果你光使用 VPN 作为“代理”。由于 VPN 本身是不开启监听端口的。那么你就必须想办法共享 VPN 的翻墙通道。至于如何共享，请看《多台电脑如何共享翻墙通道》。

◇关于操作系统的说明

本教程适用于目前的各种主流操作系统，包括但不限于 Windows、Linux、Mac OS X .....

考虑到目前 Windows 系统的用户占绝大多数，本教程拿 Windows 系统来说事儿。希望 Linux 系统和 Mac OS X 系统的用户别怨俺偏心。

注意事项：

要特别小心真实系统和虚拟系统的防火墙设置。很多人就是因为防火墙没配好，导致代理无法连通。

如果你碰到上述困难，可以参考如下博文——用 netcat 辅助你进行诊断。

《扫盲 netcat（网猫）的 N 种用法——从“网络诊断”到“系统入侵”》

★真实系统（Host OS）的物理网卡

先跟大伙儿说一下：今天这个教程跟物理网卡**【没有】**半毛钱关系。跟你电脑上装了多少块物理网卡，也**【没有】**半毛钱关系。在本文后续的介绍中，**【不会】**再涉及到物理网卡。待会儿在配置代理的时候，也**【不会】**再涉及物理网卡上的 IP 地址。切记！

另外，今天这个教程，跟你用的上网方式也**【没有】**关系。不论你是在公司上网还是在家用宽带，本教程都适用。

★真实系统（Host OS）的虚拟网卡

【这部分是重点，注意看喽！】

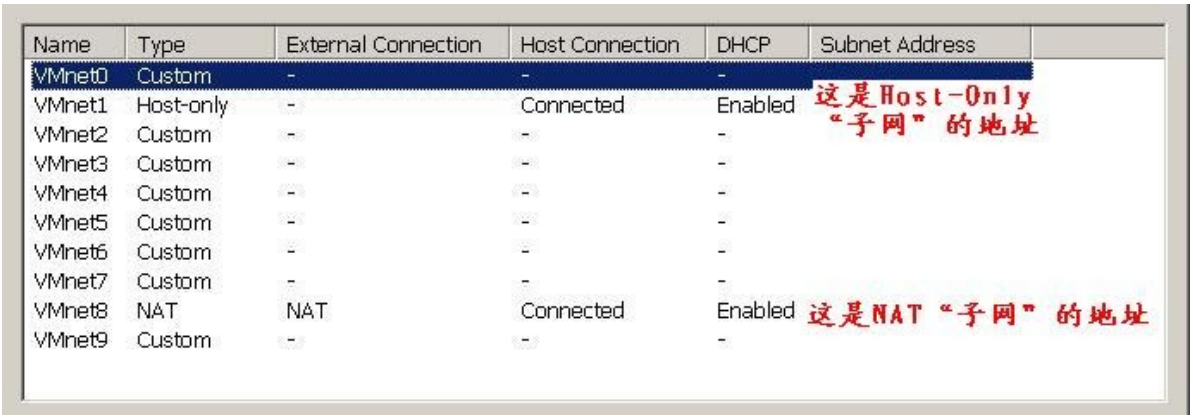
一旦你安装完虚拟机软件，那么你的操作系统中就会多出新的虚拟网卡和虚拟子网。下面俺根据 VMware 和 VirtualBox 分别说明。

◇VMware

每次安装 VMware，新增加的虚拟子网，网络地址都可能会不同，所以俺多费点口水。

首先到 Windows 控制面板的网络连接看一下，如果看到下图，就这说明 VMware 已经帮你加入了2个虚拟网卡，这俩网卡分别位于 NAT 虚拟子网和 Host-Only 虚拟子网。

然后，你到 VMware 主菜单上点“Edit“菜单，然后再点”Virtual Network Editor“菜单，会出现虚拟子网的对话框，通过该对话框可以看到 VMware 创建的所有虚拟子网。你会看到好多个虚拟子网，咱只要关心其中两个——分别是 Type 为【Host-Only】和 Type 为【NAT】的。然后，把这两个子网的”Subnet Address“分别记下来（**千万别把这俩记混了**），待会儿要用到。截图如下



（请注意，上述截图中列出的都是虚拟子网的**【网络地址】**，表示的是整个子网，所以最后一位是 `0` ）

◇VirtualBox

每次安装 VirtualBox 的时候，它创建的虚拟子网，网络地址总是一样的，所以 VirtualBox 的操作比较简单。

对于 NAT 模式，默认的虚拟子网总是 `10.0.2.0`；对于 Host-Only 模式，默认的虚拟子网总是 `192.168.56.0`（请注意，这两个是虚拟子网的**【网络地址】**，表示的是整个子网，所以最后一位是 `0` ）

牢记这俩子网的网络地址（千万别把这俩记混了），待会儿要用到。

★【单】虚拟机方案

◇安装虚拟系统（Guest OS）

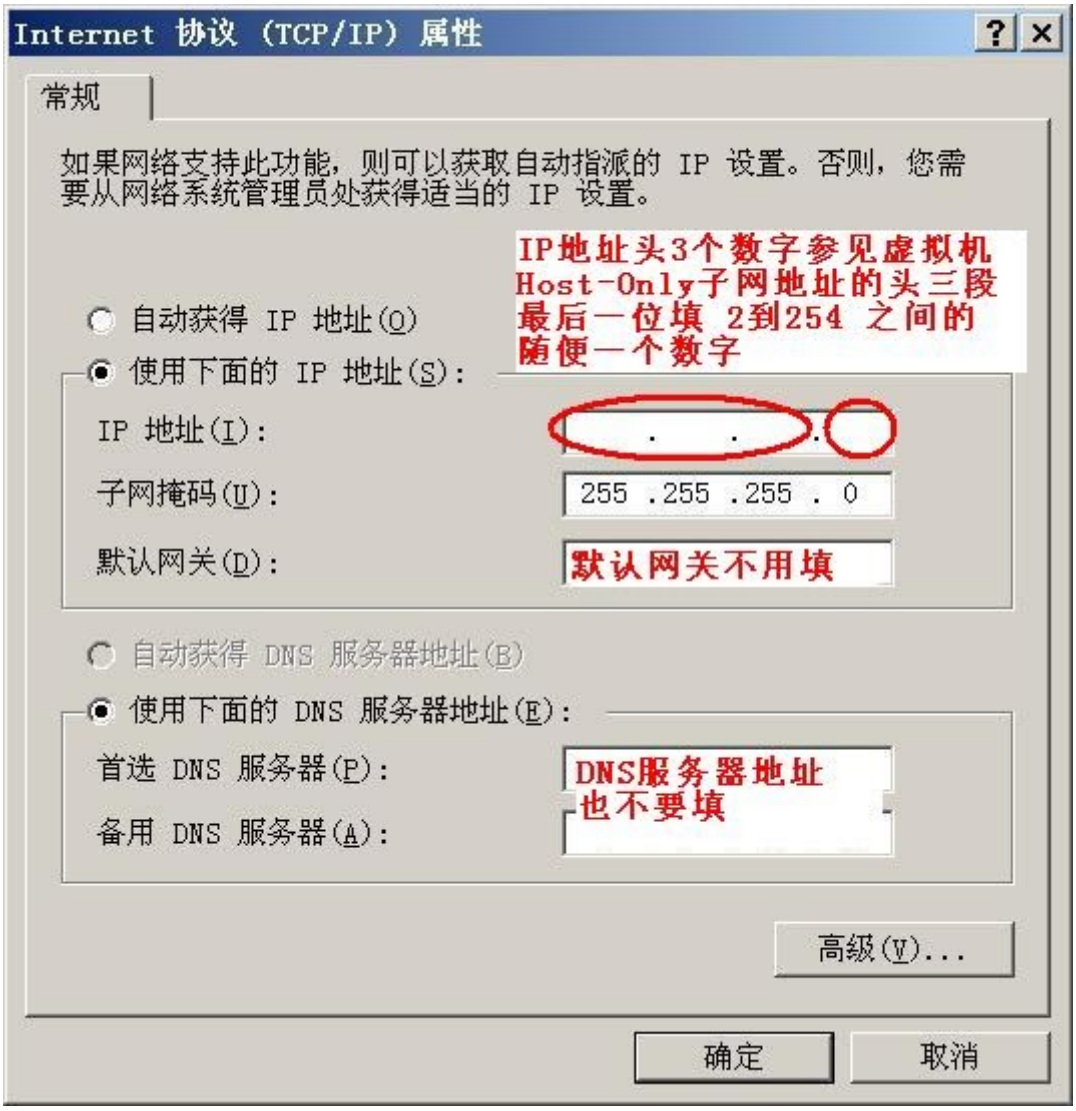
如何在虚拟机软件中安装 Guest OS，《扫盲操作系统虚拟机》系列教程已经有详细的操作图解，此处不再啰嗦。

◇虚拟系统的网卡配置

（这部分是重点，看仔细喽！）

首先，你要在虚拟机软件中设置该“虚拟系统A”的网卡模式，要设置为**【Host-Only】**。

其次，要进入“虚拟系统A”，到“控制面板”的“网络连接”里，找到那块网卡，右键菜单点“属性”，再点“TCP/IP”的属性，会出现如下截图



IP 地址

（**这步一定要小心，别填错了**）

IP 地址一共四段：

头三段，分别填写【**Host-Only**】子网的网络地址的头三个数字。请回顾刚才的章节——★真实系统（Host OS）的虚拟网卡。

第四段，你可以填 `2` 到 `254` 之间的任何一个数。

子网掩码

填写 `255.255.255.0`

默认网关

不用填

DNS

不用填

这块【Host-Only】网卡配好之后，为了验证你是否配置成功，可以执行如下步骤验证：

进入"虚拟系统"，用 `ping` 命令测试一下真实系统的那块【Host-Only】网卡的 IP 地址。如果能 ping 得到就说明你配对了。

◇代理软件的安装

在单虚拟机方案中，代理直接安装在 Host OS 里面。关于代理软件的安装，就不用再啰嗦了吧?

◇上网软件的配置

（**这部分也是重点，看仔细喽！**）

为网络软件配置代理的时候，通常要填写代理的 IP 地址和端口号。端口号通常不会搞错。因为每一款代理软件开启的端口号是固定的。但是 IP 地址常常会填错。很多人就是栽在这一步。

填写代理的 IP 地址，千万【**不能**】填 `127.0.0.1`，因为这个地址表示【本机】，也就是 Guest OS 自己。

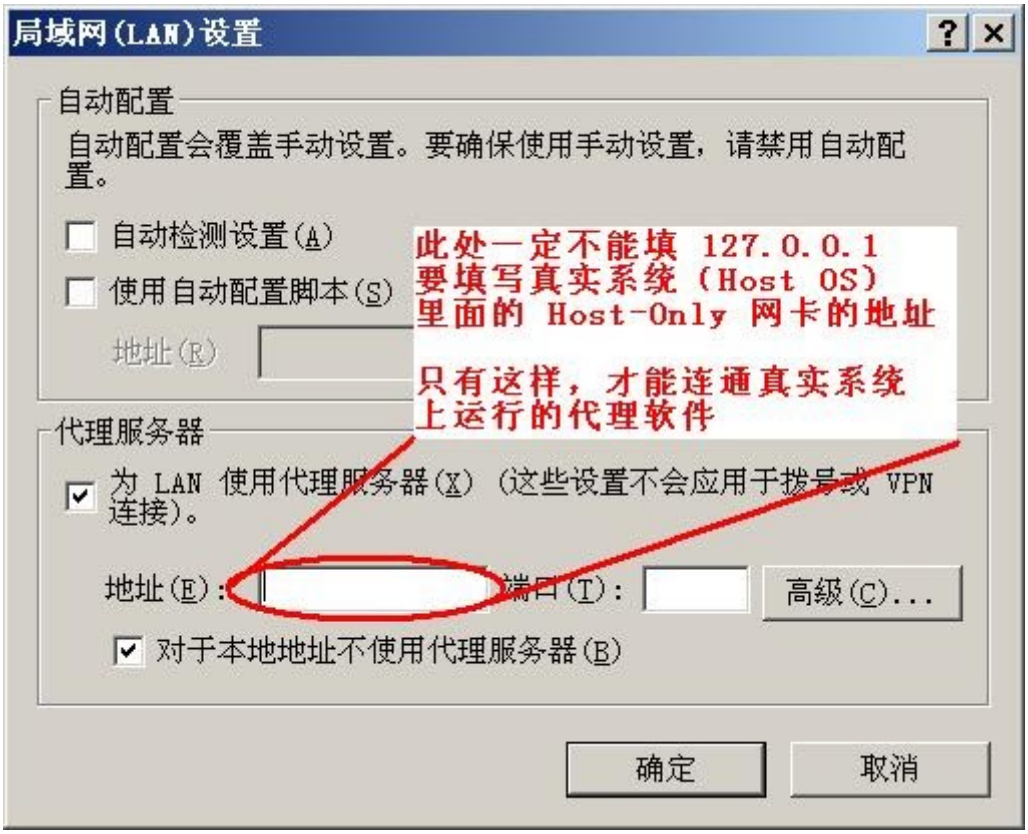
正确的写法是：填写真实系统的那个【**Host-Only**】网卡的 IP。

这个 IP 地址一共四段：

头三段，分别填写【**Host-Only**】子网的网络地址的头三个数字。请回顾刚才的章节——★真实系统（Host OS）的虚拟网卡；

第四段，填 `1`

俺以 IE 浏览器为例，截图如下



★【双】虚拟机方案（NAT + HostOnly）

◇安装两个虚拟系统（Guest OS）

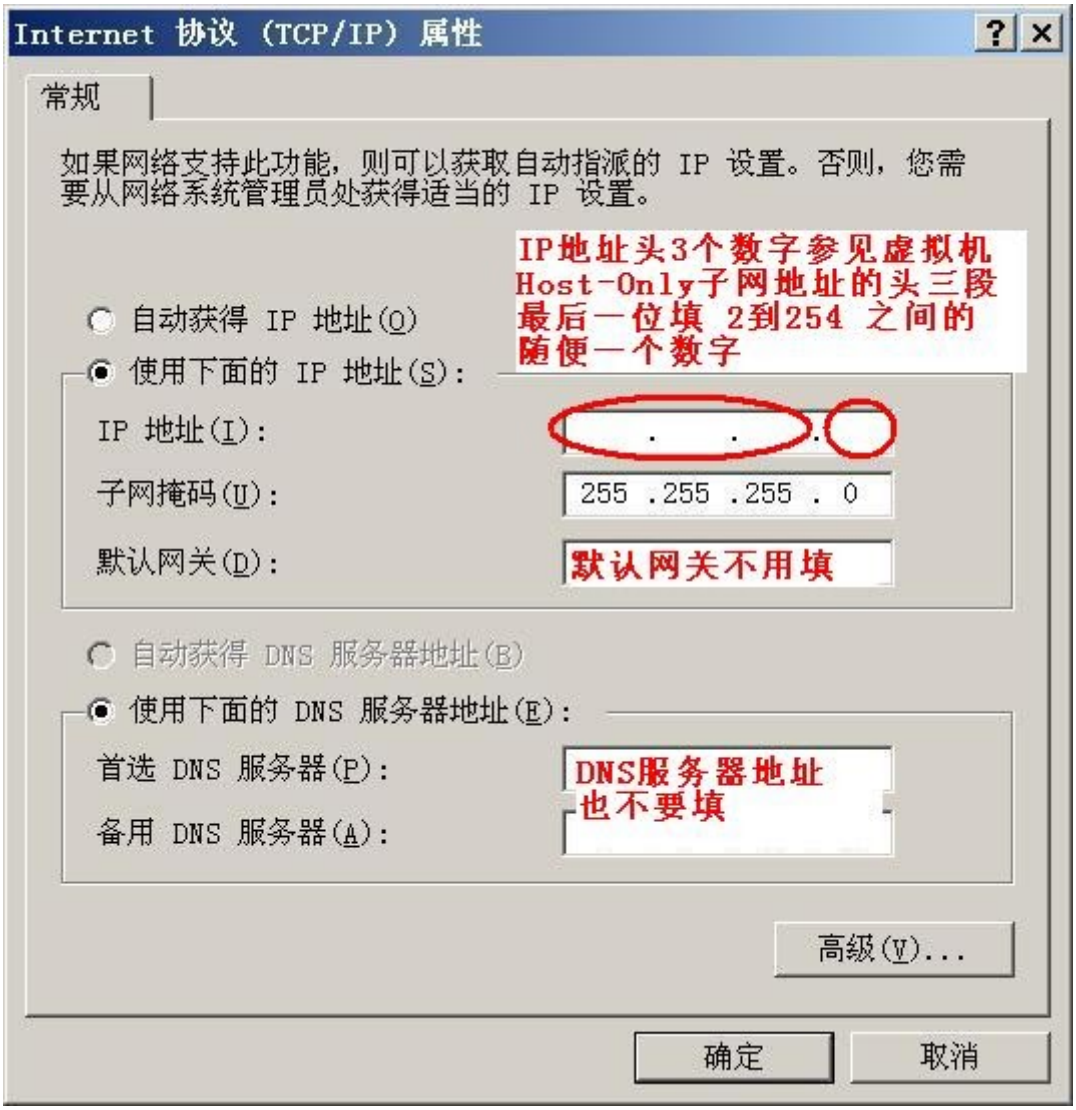
如何在虚拟机软件中安装 Guest OS，《扫盲操作系统虚拟机》系列教程已经有详细的操作图解，此处不再啰嗦。

要使用双虚拟机方案，你需要准备【**两套**】虚拟系统（Guest OS）。

◇虚拟系统A 的网卡配置

首先，你要在虚拟机软件中设置该“虚拟系统A”的网卡模式，要设置为【**Host-Only**】。

其次，要进入“虚拟系统A”，到“控制面板”的“网络连接”里，找到那块网卡，点属性，会出现如下截图



IP 地址

（**这步一定要小心，别填错了**）

IP 地址一共四段：

头三段，分别填写【**Host-Only**】子网的网络地址的头三个数字。请回顾刚才的章节——★真实系统（Host OS）的虚拟网卡。

第四段，你可以填 `2` 到 `254` 之间的任何一个数。

子网掩码

填写 `255.255.255.0`

默认网关

不用填

DNS

不用填

这块【Host-Only】网卡配好之后，为了验证你是否配置成功，可以执行如下步骤验证：

进入“虚拟系统A”，用 `ping` 命令测试一下真实系统的那块【Host-Only】网卡的 IP 地址。如果能 ping 得到就说明你配对了。

◇虚拟系统B（网关）的网卡配置

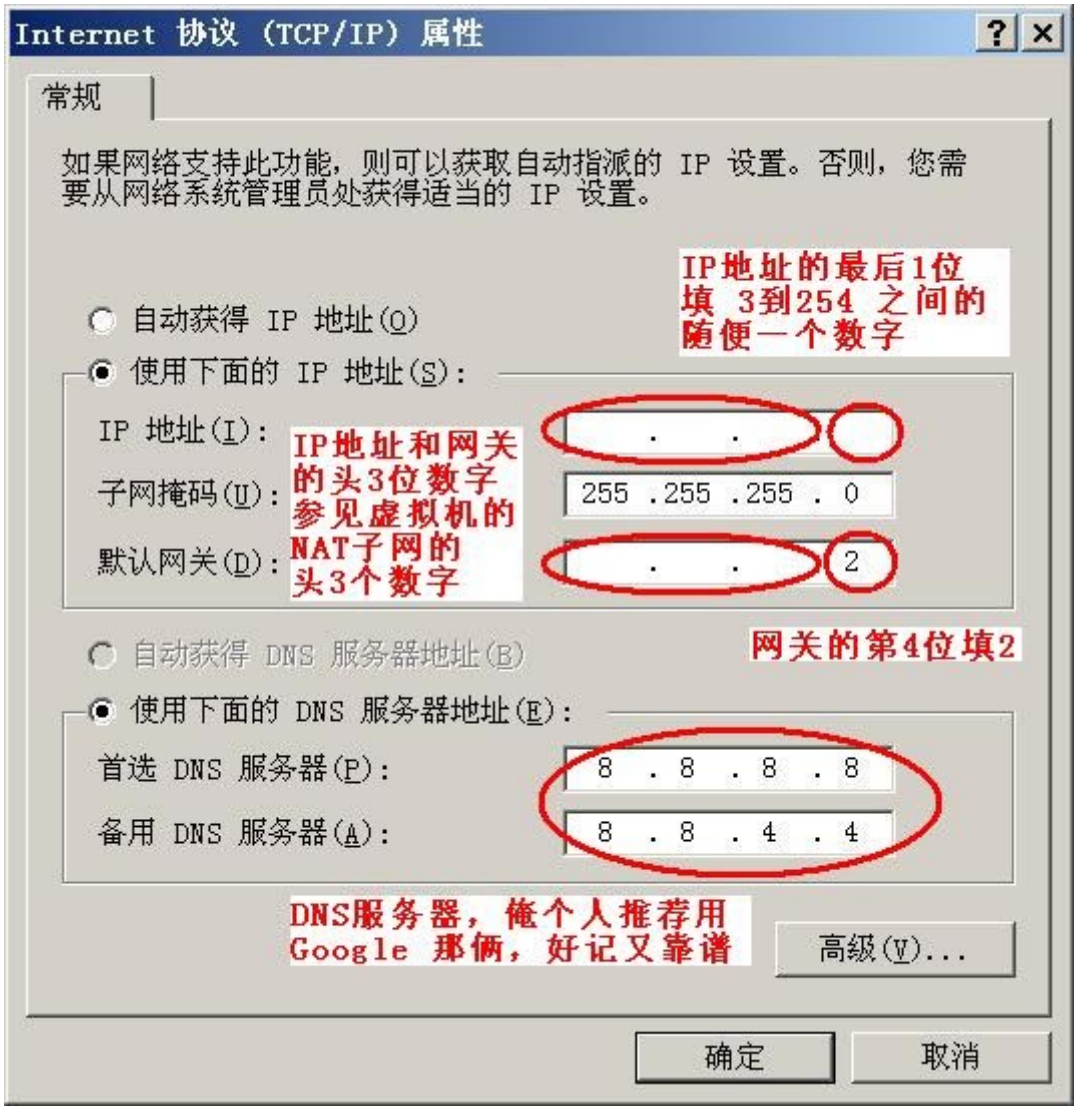
“虚拟系统B”要配置两块网卡，**【这部分非常非常容易搞错，一定要看仔细喽！】**

第1步

刚装好的“虚拟系统B”，默认已经有一块网卡了。你先把这块网卡的网卡模式，设置为【**NAT**】。

第2步

进入“虚拟系统B”，到“控制面板”的“网络连接”里，找到那块网卡，右键菜单点“属性”，再点“TCP/IP”的属性，会出现如下截图



IP 地址

【这步一定要小心，别填错了】

IP 地址一共四段：

头三段，分别填写【**NAT**】子网的网络地址的头三个数字。请回顾刚才的章节——★真实系统（Host OS）的虚拟网卡。

第四段，你可以填 `3` 到 `254` 之间的任何一个数。

默认网关

（**这步也要小心，别填错了**）

默认网关的地址也是4段，头三段就照抄刚才 IP 地址的头3个数字。第4个数字填写 `2`（不论是 VMware 还是 VirtualBox 都填 `2`）

DNS

此处填写你常用的 DNS 服务器，俺个人建议填 Google 的那俩（`8.8.8.8` 和 `8.8.4.4`）

子网掩码

填写 `255.255.255.0`

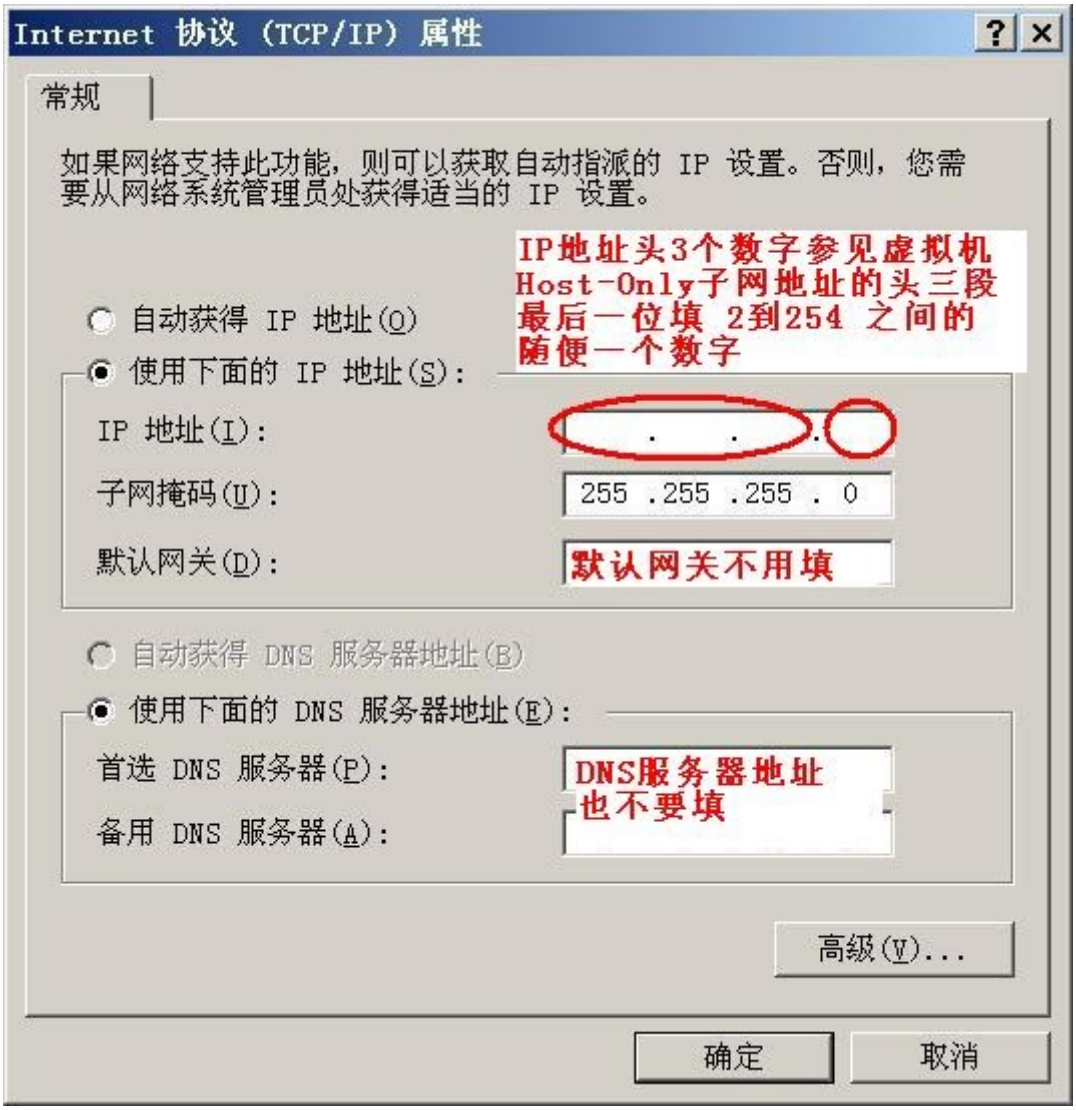
这块 NAT 网卡配好之后，为了验证你是否配置成功。可以在“虚拟系统B”里面开一个浏览器（不设代理），访问一下互联网。如果能访问，说明这块 NAT 网卡 OK 了。

第3步

打开虚拟系统的设置对话框，再添加一块网卡了。（如何加第二块网卡，请看《扫盲操作系统虚拟机》教程）然后把这块网卡的网卡模式，设置为【**Host-Only**】。

第4步

进入“虚拟系统B”，到“控制面板”的“网络连接”，找到新添加的【**第二块**】网卡，右键菜单点“属性”，再点“TCP/IP”的属性，会出现如下截图



IP 地址

【这步一定要小心，别填错了】

IP 地址一共四段：

头三段，分别填写【**Host-Only**】子网的网络地址的头三个数字。请回顾刚才的章节——★真实系统（Host OS）的虚拟网卡。

第四段，你可以填 `2` 到 `254` 之间的任何一个数。

再提醒一下：虚拟系统A和虚拟系统B各自有一块【Host-Only】网卡，这两块网卡的 IP 地址头三位是一样的，**第四位不能相同——否则 IP 地址会冲突。**

子网掩码

填写 `255.255.255.0`

默认网关

不用填

DNS

不用填

◇代理软件的安装

在双虚拟机方案中，**代理是安装在“虚拟系统B”（网关）里面的**，别搞混喽！关于代理软件的安装，就不用再啰嗦了吧？

◇上网软件的配置（以浏览器为例）

在双虚拟机方案中，那些有危险的**上网软件是安装在“虚拟系统A”里面的**，别搞错喽！

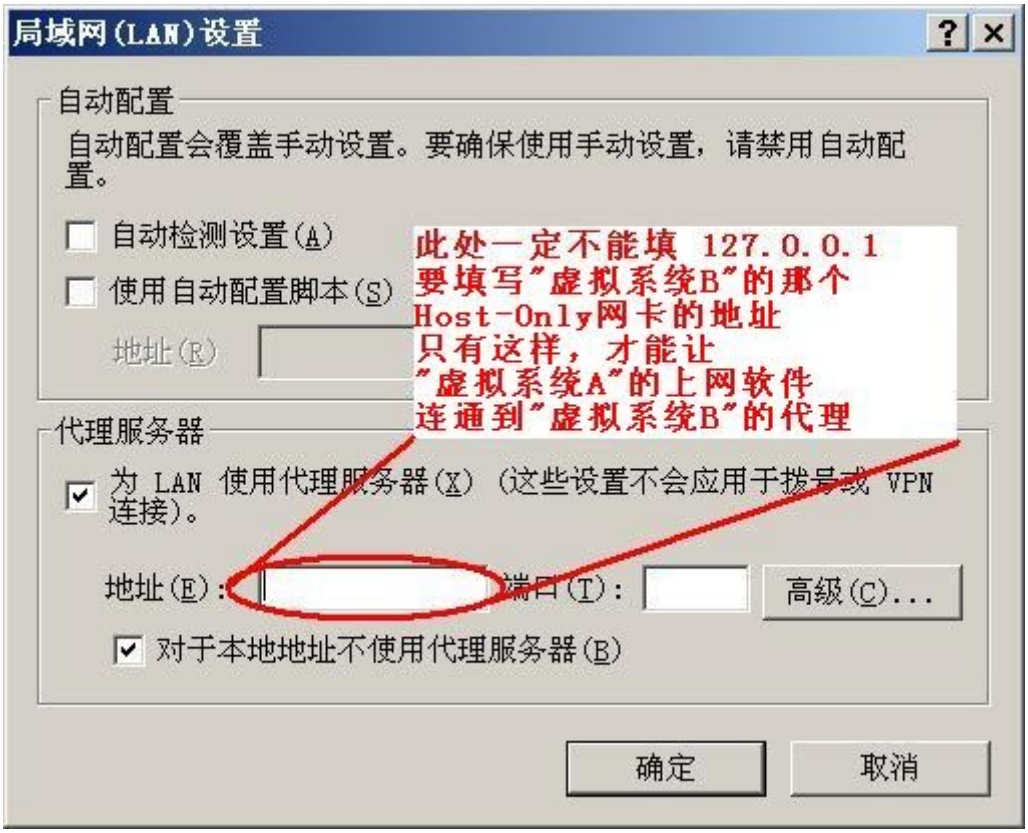
下面详细说说上网软件的代理配置。

为网络软件配置代理的时候，通常要填写代理的 IP 地址和端口号。端口号通常不会搞错。因为每一款代理软件开启的端口号是固定的。但是 IP 地址常常会填错。所以**请大伙儿把下面看仔细喽！**

填写代理的 IP 地址，**千万【不能】**填 `127.0.0.1`，因为这个地址表示本机，也就是“虚拟系统A”自己。

正确的写法是：填写另一个虚拟系统（虚拟系统B）的那个【**Host-Only**】网卡的 IP。

因为“虚拟系统B”（网关）有两块网卡，很多人填错了，栽倒在这一步。俺以 IE 浏览器为例，截图如下



◇注意事项

在双虚拟机方案中，虚拟机在启动时有可能会出现错误提示：“网络上有重名”。如果你碰到这个错误，只需把虚拟机网卡的 NetBIOS 功能禁用，即可解决。如何禁用 NetBIOS，请看“微软官网这里”。

★【双】虚拟机方案（NAT + Internal）

（经热心读者 meek 提醒，俺补充了这一章节）

在《扫盲操作系统虚拟机》系列的第5篇，俺介绍了 Guest OS 的各种网卡模式，其中也包括“Internal 模式”。这种模式类似于“Host Only 模式”，差别在于——Host OS 上的进程无法看到“Internal 模式”的虚拟网卡。

因此，“双虚拟机方案”也可以采用“NAT + Internal”的玩法。

好处是——隔离性更好；

缺点是——Internal 这种网卡模式 Virtual Box 支持而 VMware 【不】支持 （对 VMware 用户，可以通过添加“自定义网卡”来达到类似效果）。

如果你想采用“NAT + Internal”的方式配置虚拟网卡，那么整个配置过程跟前一章节介绍的“NAT + HostOnly”很相似，差别仅仅在于把两个虚拟网卡的“HostOnly 模式”替换为“Internal 模式”。所以俺就不重复罗嗦了。

★验证虚拟机的隔离性

无论是“双虚拟机方案”还是“单虚拟机方案”，配置完毕，联通代理之后，为了保险起见，你需要再验证一下虚拟机的隔离性。

\1. 先把代理关闭，到虚拟机A 中运行危险软件，看它是否能联网成功。（如果联网成功，说明你的配置有误）

\2. 进入虚拟机A，开启一个命令行窗口，用 `ping` 命令测试一下你常用的 DNS 服务器的 IP（如果能 ping 到，说明你的配置【有误】）

★结尾

由于涉及两套方案，而且涉及两种虚拟机软件，本文有点长，步骤有点多。如果你根据本教程配置完毕，还是不行，请到本文留言，俺会尽量解答。

[如何隐藏你的踪迹，避免跨省追捕8]：如何搭配“多重代理”和“多虚拟机”

文章目录

话说这个系列已经2年没更新了（之前的“第7篇”发布于2013年1月）。前几天看到有读者在博客评论区讨论“双重代理”如何部署在“双虚拟机环境”。于是俺借着这个话题，再发一篇。

今天这篇，主要谈谈——如何把“多重代理”和“多虚拟机”组合在一起。可能很多同学把这个问题想得过于简化。其实捏，不同的 Tor 前置代理以及不同的上网软件，有可能需要不同的部署方式。具体取决于——相关软件的靠谱程度，以及你对安全性的要求有多高。

事先声明：

如果你不了解啥是“多重代理”，或者你不了解啥是“多虚拟机方案”，建议先看完本系列的前面3篇：

《[如何隐藏你的踪迹，避免跨省追捕5]：用多重代理隐匿公网IP》

《[如何隐藏你的踪迹，避免跨省追捕6]：用虚拟机隐匿公网IP（原理介绍）》

《[如何隐藏你的踪迹，避免跨省追捕7]：用虚拟机隐匿公网IP（配置图解）》

★名词解释

为了避免产生歧义，先界定一下相关的术语。

◇多重代理

本文提及的“多重代理”，指的是——基于 Tor 的多重代理。在各种“多重代理”的方案中，这种形式是最实用的（兼顾了安全性和速度）。

顺便说一下：

去年（或者是前年）曾经有读者在博客评论中咨询“多重 VPN”。俺个人认为：多重 VPN 的方案，隐匿性【远远不如】基于 Tor 的多重代理。因为 VPN 服务器的的 IP 是相对固定的，即使你用了多重 VPN，被人反查（逆向追踪）的可能性还是比较大。在这方面，Tor 就远远好于 VPN。因为 Tor 的线路是【动态变化】滴（大约10分钟变化一次），导致逆向追踪非常困难。

◇多虚拟机（Guest OS）

在本系列前面的博文《用虚拟机隐匿公网IP（原理介绍）》，俺介绍了两种虚拟机方案——“单虚拟机”和“双虚拟机”。

今天这篇博文，面向的是安全性要求比较高的读者；因此，【不】讨论“单虚拟机”方案。

在双虚拟机方案中，俺把充当网关的虚拟机称为“网关虚拟机”，把那个网卡设为“Host Only”模式的虚拟机称为“隔离虚拟机”。

◇宿主系统（Host OS）

既然说到 Guest OS，顺便再提一下 Host OS。

“Host OS 的安全性”本文所有方案的前提。换句话说：如果 Host OS 不安全，本文的任何方案都【没有】意义。

★为啥要考虑部署方式?

如果你对安全性的要求比较高，对如下几种情况，你需要额外考虑部署的问题。

◇多个隔离虚拟机

“多个隔离虚拟机”是啥意思捏?

以俺本人为例，俺所有的日常工作都是在虚拟机中进行。（这么干的好处很多，具体参见“这篇博文”）。所以，俺会有一个单独的虚拟机，用于“编程随想”这个身份的上网；并且会有另一个虚拟机，用于俺的真实身份的上网。如此一来，至少就有2个“隔离虚拟机”了。

那么，在这种情况下，Tor 是应该安装到“网关虚拟机”还是安装在“隔离虚拟机”捏？大伙儿先思考一下。

◇网关虚拟机的环境【不】可靠

所谓的“网关虚拟机的环境不可靠”，有如下几种情况：
\1. 网关虚拟机的操作系统不可靠
（比如说，你的网关虚拟机安装的是 Windows 系统。有些人可能会担心 Windows 存在 NSA/美国国安局 的后门）
\2. Tor 的前置代理不可靠
（比如说，你用了国内 VPN 提供商提供的 VPN 做 Tor 的前置代理。该 VPN 需要安装客户端，而你对这款客户端不放心）
（比如说，你用了 无界 和 自由门 作为 Tor 的前置代理。这两款工具【不是】开源的。所以你对这两款工具不放心）
\3. 网关虚拟机安装了某些不可靠的软件
（比如说，有些电信提供商/ISP 的宽带拨号，需要在系统中额外安装一个拨号软件。而你对 ISP 提供的拨号软件不放心）

那么，在这种情况下，Tor 是应该安装到“网关虚拟机”还是安装在“隔离虚拟机”捏？大伙儿先思考一下。

◇你用的上网软件【不】可靠

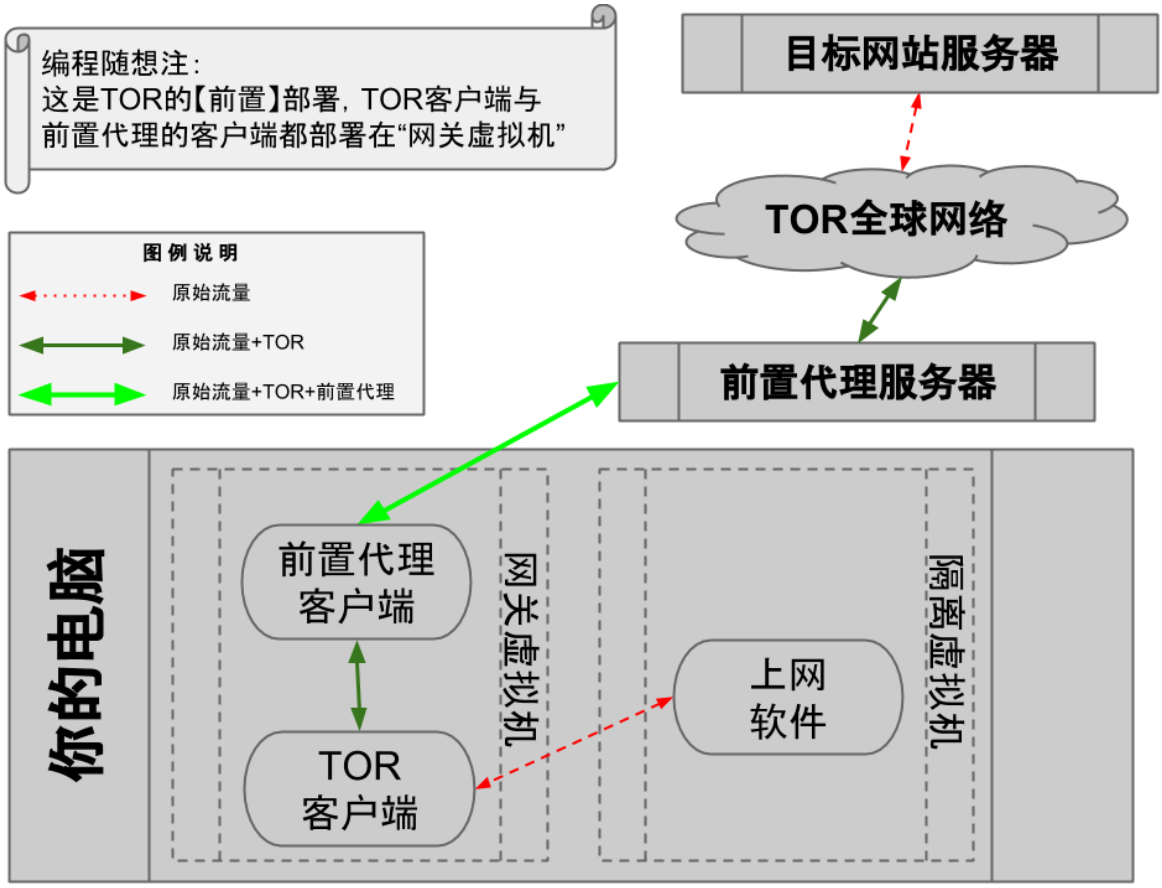
最后一种情况是：上网软件不可靠。
所谓的“上网软件”，顾名思义就是你用来上网的软件。可以是浏览器，也可以是聊天工具或者下载工具。
为啥会碰到“上网软件不可靠”的情况捏？
举例如下：
比如你想用 QQ 进行反党宣传。但是你又不想让腾讯知道你的公网 IP。于是你就需要组合“多重代理”和“多虚拟机”来确保你的公网 IP 不会暴露给疼逊的服务器。在这种情况下，QQ 客户端就属于“不可靠的上网软件”。

那么，在这种情况下，Tor 是应该安装到“网关虚拟机”还是安装在“隔离虚拟机”捏？大伙儿先思考一下。

★两种基本的部署方式

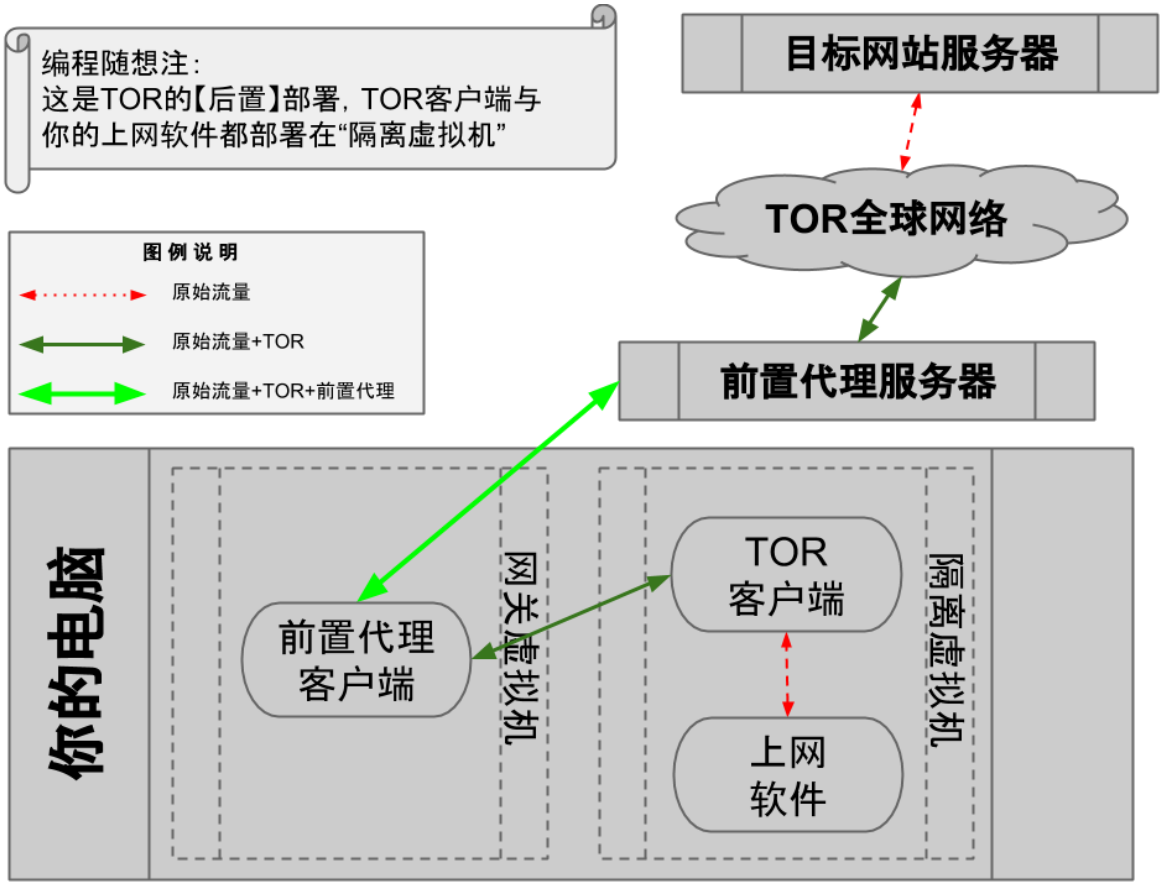
◇Tor 的【前置】部署——部署在“网关虚拟机”

为了形象起见，俺直接画一个示意图，如下：



◇Tor 的【后置】部署——部署在“隔离虚拟机”

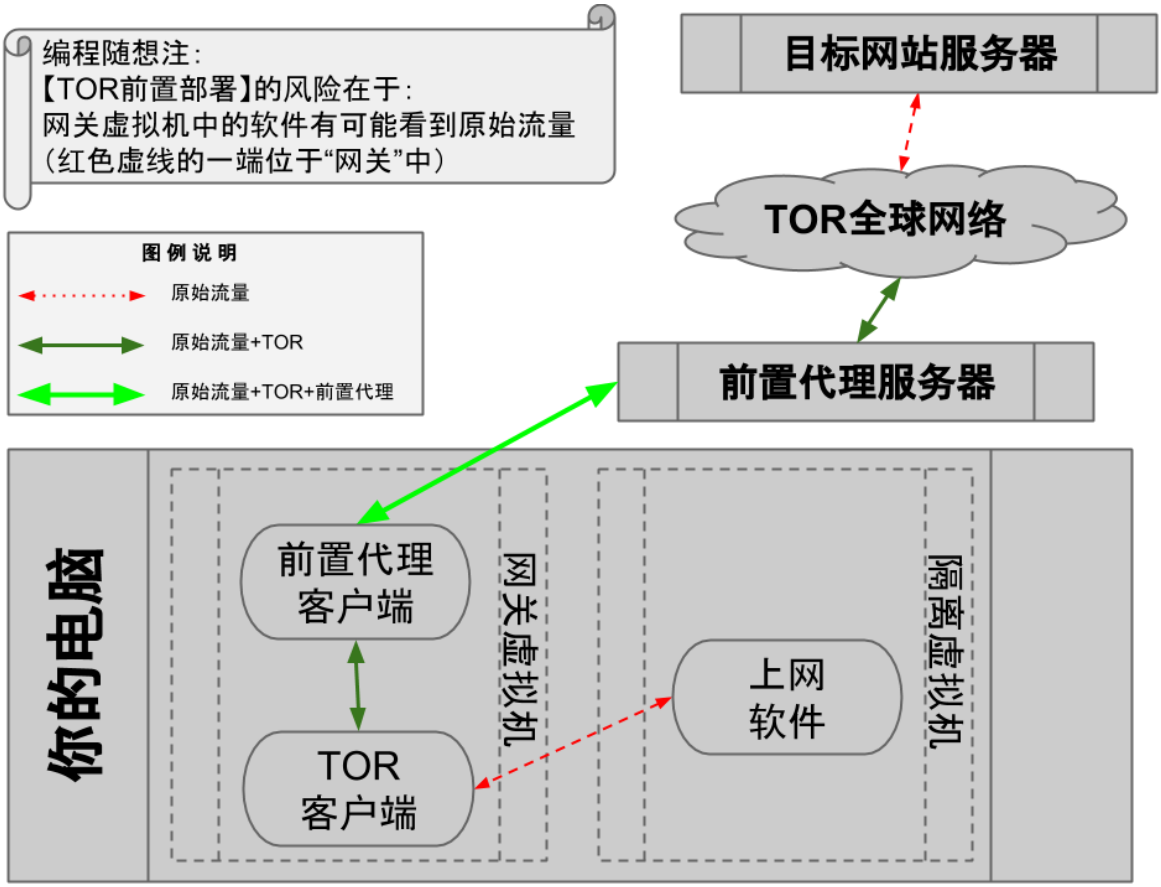
为了形象起见，俺直接画一个示意图，如下：



★这两种部署方式各自的风险

◇Tor 的【前置】部署——部署在“网关虚拟机”

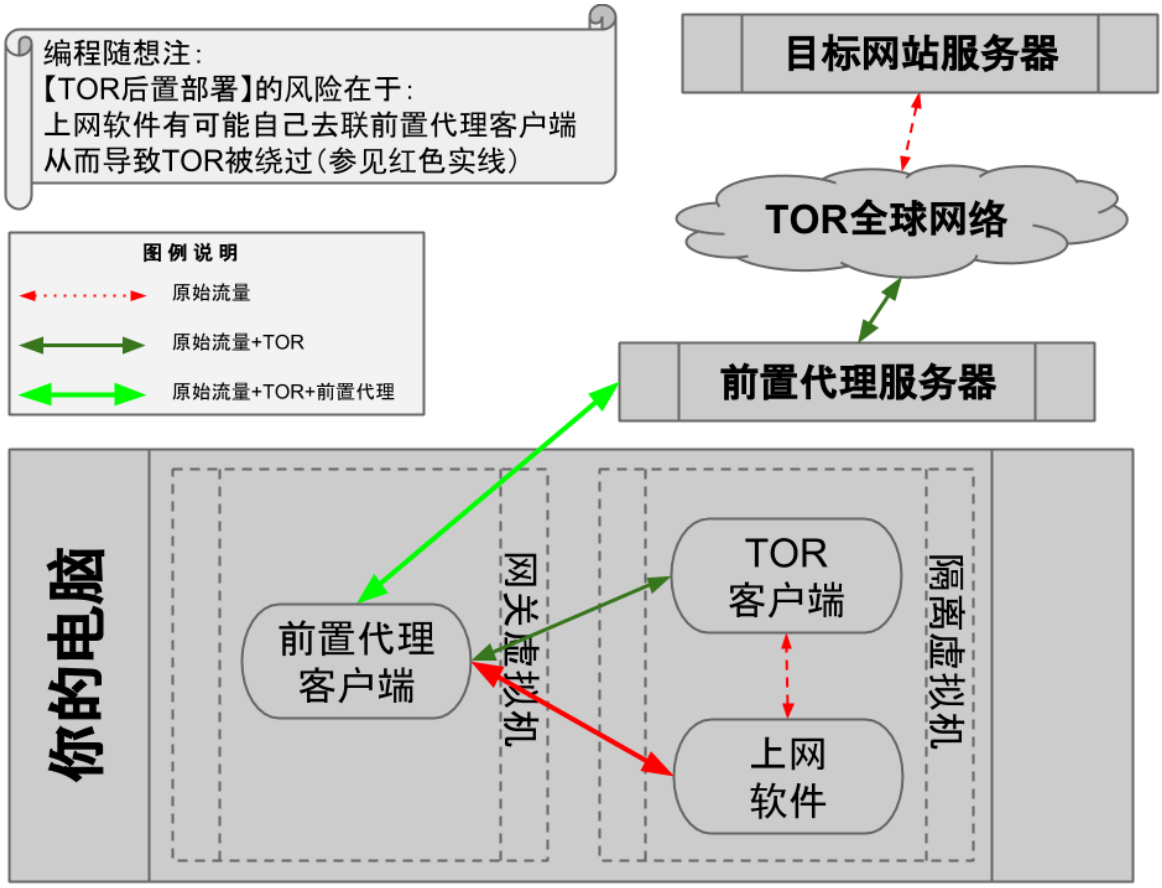
如下图所示，如果“网关虚拟机”的环境不可靠（比如说其中有恶意软件），那么该恶意软件有可能会监控“网关虚拟机”系统中的网络流量。如此一来，就会看到【原始流量】（图中的“红色虚线”）。



◇Tor 的【后置】部署——部署在“隔离虚拟机”

如下图所示，如果“上网软件”不可靠，那么该软件有可能直接尝试去连接“Tor 的前置理”（参见图中的“红色实线”），从而导致 Tor 被绕过。

虽然这种行为不会直接导致你的公网IP暴露，但是会大大降低你在网络层面的隐匿性。如果有人要对你进行逆向追溯，只要查出前置代理服务器上的访问日志，就【有可能】查出你的公网IP。

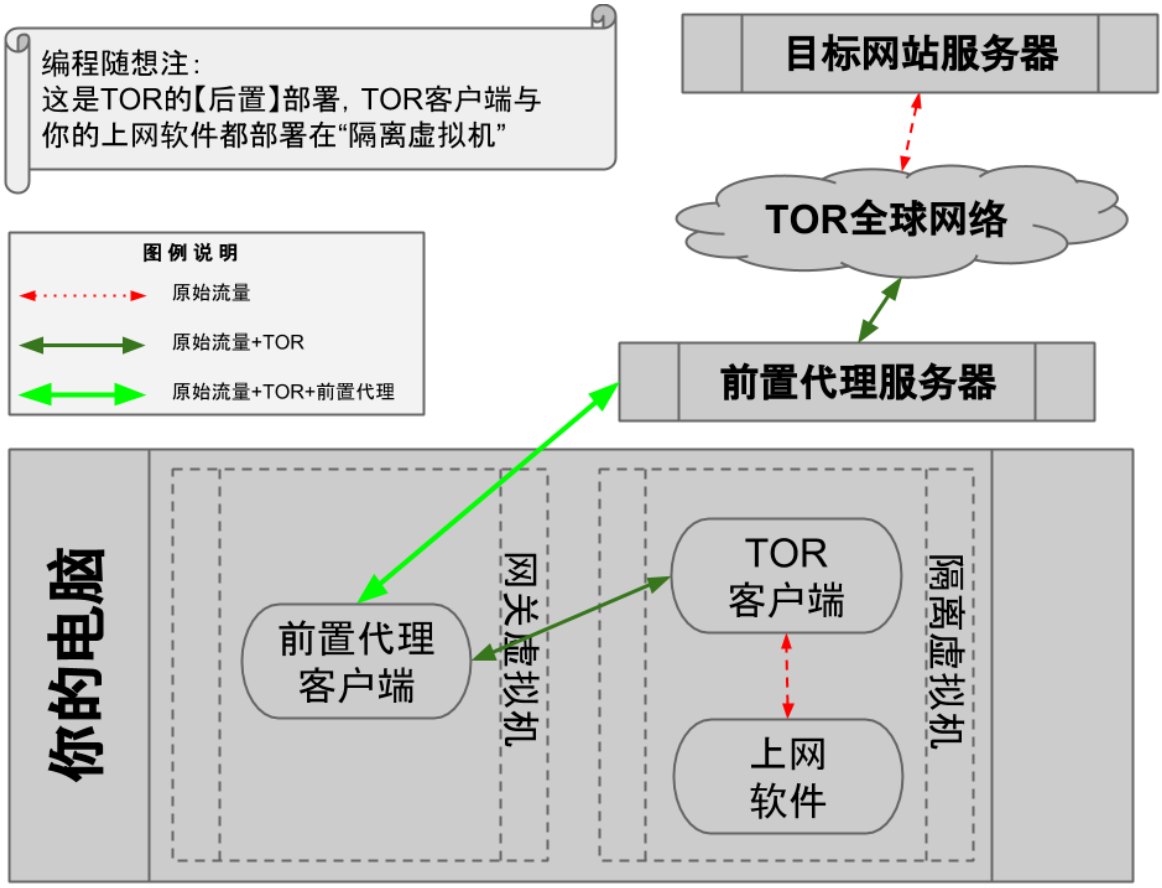


★不同情况下，该如何应对？

下面来说说应对的策略。

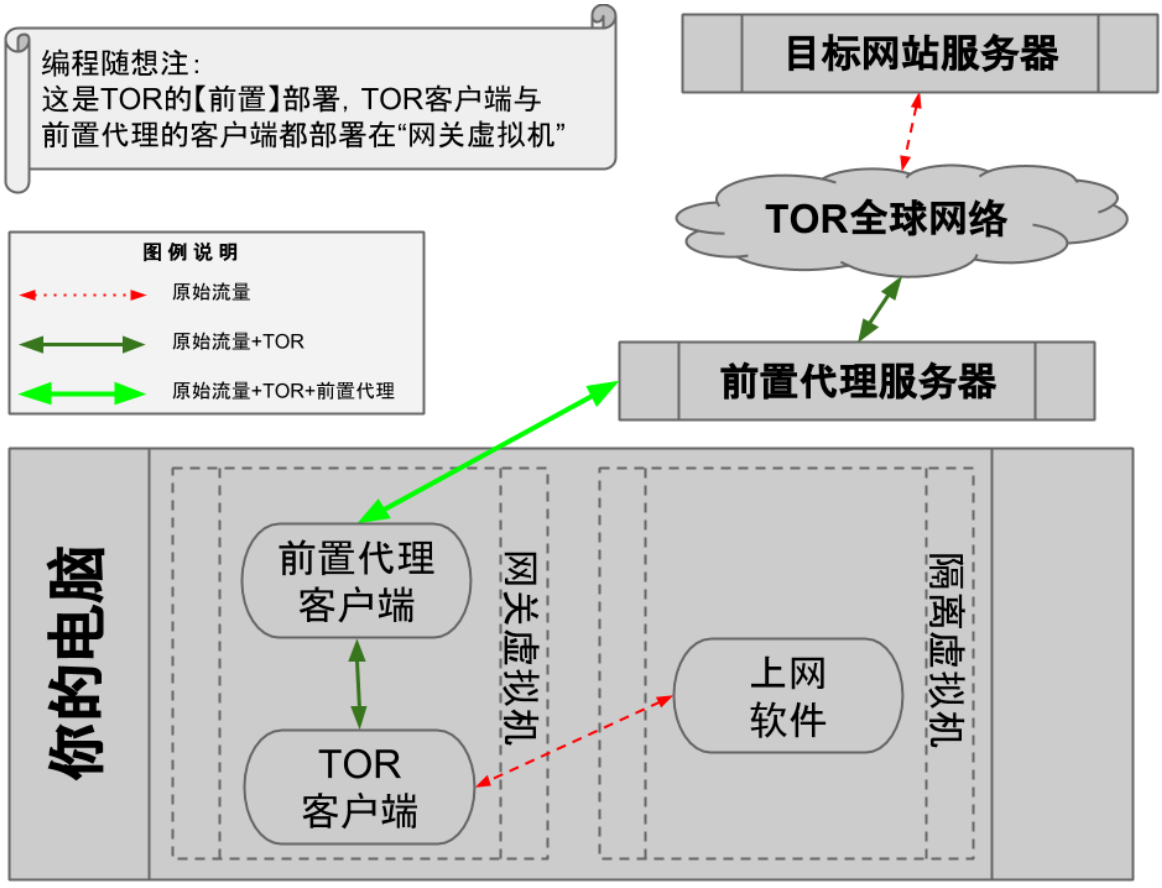
◇仅仅是“网关的环境”不靠谱

很显然，这种情况下，你必须采用“Tor 后置部署”的部署方式。示意图如下：



◇仅仅是“上网软件”不靠谱

很显然，这种情况下，你必须采用“Tor 前置部署”的部署方式。示意图如下：



◇都很靠谱，但是有多个“隔离虚拟机”共用同一个“网关虚拟机”

对这种情况，俺需要稍微费点口水。

首先，俺要唠叨一下：

之所以需要多个“隔离虚拟机”，就是想要达成某种【隔离性】。比如在本文开头提及俺本人的例子——俺把不同的上网身份隔绝在不同的虚拟机，就是为了尽可能避免两者的关联性。

如果采用“Tor 前置”部署

由于“网关虚拟机”只有一个。把 Tor 部署在网关导致你的多个“隔离虚拟机”共用【同一个 Tor 环境】。

“共用同一个 Tor 客户端”的风险在于：多个“隔离虚拟机”的上网流量，有比较大的可能性，这些流量会使用同一个（Tor）出口节点。由于的出口节点能够看到原始流量，万一原始流量是明文的HTTP，并且该节点是蜜罐，那么风险就会变大。

针对某热心读者的质疑，俺补充说明一下：

Tor 客户端会同时创建几个不同的虚拟链路，用于传输数据。每个链路包含三个 Tor 节点（分别是：入口节点、中间节点、出口节点）。只有“出口节点”能看到原始流量。

如果你正在使用的“隔离虚拟机”的个数超过 Tor 客户端创建的链路数量，那么就存在——“不同的隔离虚拟机共用同一个链路”——这种情况下，该链路的出口节点，有可能会发现这些不同的隔离虚拟机的原始流量之间的相关性。

反之，**如果采用“Tor 后置”部署**

由于每个“隔离虚拟机”都有自己的 Tor 客户端。所以这几个“隔离虚拟机”的上网流量，在同一个瞬间使用同一个（Tor）出口节点的概率小很多。就算碰巧都用了同一个“出口节点”，因为不同的 Tor 客户端建立的虚拟链路也是不同的，所以这个出口节点就会以为这些流量来自不同的上网用户（因此，你的多个网络身份【不会】被关联起来）。

◇网关环境不可靠，上网软件也不可靠

本来不想写这一节。但是俺估计：某些喜欢抬杠的同学会这么问。

老实说，这种情况是比较蛋疼的（很罕见）。但是同样有办法搞定。而且方法有两种，具体如下：

办法1

把“双虚拟机”改为“三虚拟机”。多出来的那个虚拟机用来单独安装 Tor（不妨称之为“Tor 虚拟机”）。

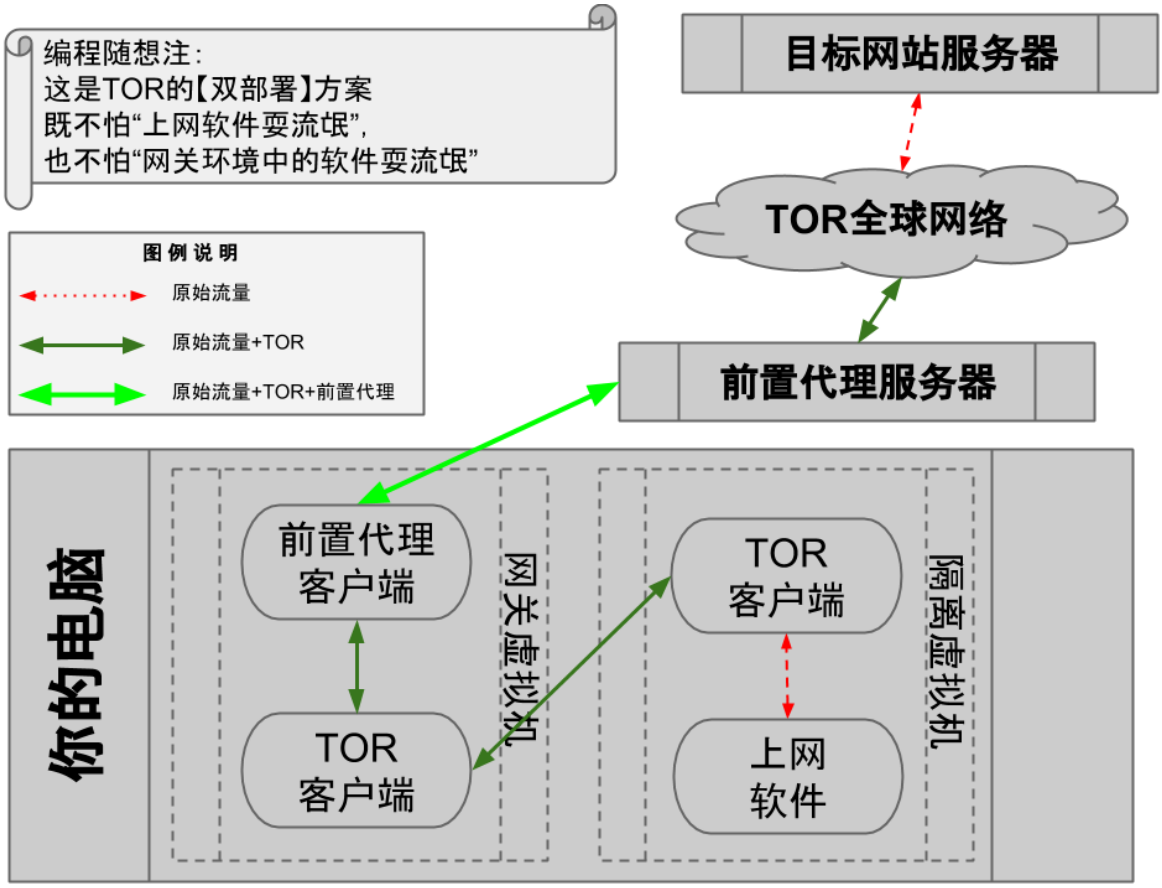
注意事项：

为了防止上网软件绕过 Tor，直接去联前置代理，需要在“网关虚拟机”上配置防火墙规则，使得“隔离虚拟机”无法直接访问“网关虚拟机”。

办法2

把“双重代理”改为“三重代理”。也就是说——Tor 既部署在“网关虚拟机”，也部署在“隔离虚拟机”。

为了让大伙儿直观些，再画一张示意图，如下：



小结

这两个方法，各有缺点：

方法1：因为“三虚拟机”会增加 Host OS 的性能开销，让人感觉有点“杀鸡用牛刀”。

方法2：虽然只用到两个虚拟机（比 方法1 的性能好），但是出现了“Tor over Tor”的组合。Tor 官网的文档【反对】这种组合方式（链接在“这里”）

★总结

描述了这么多种部署方式，不晓得大伙儿看了之后，会不会晕？

为了防止大伙儿晕菜，俺在结尾作一个总结发言：

不论是哪一种部署方案，出问题的根源都是因为【Tor 被绕过】。

对于“Tor 前置部署”，如果网关中的软件偷窥了进入 Tor 之前的流量，其效果就等同于——绕过 Tor 获取了原始流量。

对于“Tor 后置部署”，就更加明显了——上网软件如果耍流氓，有可能直接去联前置代理（同样是绕过 Tor）。

对于“只部署一个 Tor 客户端”的情况，你要确保 Tor 运行在可靠的环境中。如果“网关虚拟机”不可靠，就把 Tor 放在“隔离虚拟机”；反之，如果“隔离虚拟机”不可靠，就把 Tor 装在“网关虚拟机”。

如果两边都不可靠，要么加出一个单独的虚拟机装 Tor（刚才所说的“三虚拟机”），要么两边都装 Tor（刚才所说的“三重代理”）。

[如何隐藏你的踪迹，避免跨省追捕9]：从【时间角度】谈谈社会工程学的防范

文章目录

★引子

距离前一篇博文有14天了。距离上一次评论区留言（本月20日）也有9天了。这些天俺在干啥捏？主要是重构评论区的代码，顺便优化一下性能。仔细的读者应该能感觉到界面上的一些小变化。后面俺会单独写一篇博文，聊聊近期对评论区的改造。

几天不出现，又有人在评论区散布谣言（说俺被捕了）。估计造谣的家伙也就是之前搞大规模刷屏的职业五毛【团队】。

职业五毛的造谣正好给了俺灵感——今天来聊一下：俺为啥要时不时地静默一段时间？——这其实是防范社会工程学风险的重要措施。

★关于本系列的说明

话说本系列已经中断2年零2月——对俺的懒惰（一贯挖坑不填），再次表示严重抱歉 :(

除了俺的懒惰，还有一个原因是：（对于防跨省）技术方面已经聊得差不多了。所以要继续写的话，俺就要开始聊【非技术】的话题——所谓的【非技术】也就是指【社会工程学】层面。

关于“社会工程学的防范”，这是一个很大的话题，而且相关的注意事项非常零散，俺一次性想不了那么全。所以捏，就分成几篇来写，每篇聊其中一个角度。今天这篇是头一篇，以【时间角度】作为切入点来进行探讨。

对【社会工程学】相关的概念不太熟悉的同学，建议先看俺写的另一个系列《扫盲社会工程学》，然后再来看本文以及本系列后续博文。

★名词解释

首先来介绍几个相关术语。

◇在线时间

这是本文重点要聊的核心概念。

从字面意义而言就是：在线活动留下的时间印迹。

如果要更严密地说：“在线时间”是时间轴上若干个线段的并集。每个线段表示一次在线活动。

◇网络身份

在讨论“在线时间”这个概念的时候，一定要牢记：【“在线时间”是相对于“网络身份”而言的】。
啥意思捏？稍微解释一下：
每个人都可以有多个网络身份，不同的网络身份，其在线时间可以是不同滴。

◇【显式】在线时间 VS 【隐式】在线时间

顾名思义，【显式】就是指活动痕迹比较明显的；【隐式】则反之。
为了解释这两者的差别，以俺博客来举例。
当俺发出一篇博文，或者在评论区回复读者留言，这些活动都是很明显的，属于“【显式】在线时间”。
有时候，俺会去修改博客的界面代码（比如近期俺就在频繁修改“反刷屏代码”）。如果俺对代码的修改【没有】引起界面功能的变化，这种修改通常不会被读者感觉到。那么这种情况就算是“【隐式】在线时间”。
那么，【隐式】在线时间能否被刺探出来捏？接下来会聊到这个话题。

◇静默期

没有网络活动的那些时间段，称之为“静默期”。
“静默期”相当于时间轴上那些线段之间的间隙。

★“在线时间”会暴露哪些信息？

即使不参照其它因素，仅仅观察某个网络身份的“在线时间”，就已经可以收集到对应自然人的信息。如果这个网络身份的活动非常频繁并且维持足够长的时间，收集到的信息就会很丰富，足以刻画出其背后那个自然人的详细特征。
下面举几个例子：

举例1：时区
这个是最容易想到的。如果某人上线很频繁或者在线活动的时间很长，那么就可以观察出此人的作息规律（人总是要睡觉的）；通过作息规律就可以猜测出大致的时区范围（±2）。
比如说：早在前几年，某些有心的读者就通过俺回复评论的时间规律，判定俺在“东8区”。

举例2：省份
前几年，新疆经历了长时间的封网（物理断网），时间跨度是：2009年7月~2010年5月。假设某个网友在 Twitter 上频繁发推。然后在2009年7月突然停止发推并持续至少10个月，那么有【很大概率】——这个帐号背后的自然人位于新疆。
（请注意：俺说的是“很大概率”而不是“一定”，因为还存在其它小概率的情况）

这方面的例子还有很多，大伙儿自己琢磨。

★“网络身份”与“真实身份”的时间关联性/相关度

前一个章节提及的“暴露信息量”，都还【不算严重】——因为暴露出来的信息量比较小。
现在俺要来聊一下比较严重的情况。

◇一个反面案例

为了形象一些，拿俺本人来【虚拟】一个反面案例。

假设俺很勤快（实际上俺很懒），【每天】都在博客上回复读者留言，并坚持了很多年（实际上俺连续回复评论很少超过3天）。

由于博客的读者很多，俺身边的同事或朋友，也有人是读者。当然，一开始这些身边的同事或朋友，并不知道编程随想原来就是某某人。

现在，假设出现了突发的意外——俺因为出车祸住院一周。于是就【没法】再如往常一样【每天】回复读者评论。

由于这个规律已经持续了 N 年，一旦出现中断就会给读者留下很深刻的印象。

那么，俺身边的这几个读者就会发现一个很严重的巧合：编程随想以往很有规律的活动突然中断一周，而身边的某某人正好也在【同一时间段】住院。

这种严重的巧合，足以导致身份暴露。

（通过上述这个虚构的案例，你应该明白：两种身份的关联性意味着什么）

◇这个案例的本质

上述案例之所以导致身份暴露，本质上可以归纳为如下几点：

\1. 网络身份需要靠真实身份操作（这句是屁话）

\2. 网络活动的规律性太强（规律性越强，一旦“打破规律”给人的印象就越深刻）

\3. 意外事件导致网络身份和真实身份出现【同样的】“时间空档”（术语叫“高度相关性”）

★多个“网络身份”之间的时间相关性

如果同一个“自然人”操作多个“网络身份”，这些网络身份之间也会存在与上述案例类似的时间相关性。

（具体的原理及例子差不多，俺就省点力，不浪费口水了）

★如何降低上述这2种相关性？

像上述这样的反面案例，该如何防范捏？下面说说俺能想到的几点（俺想的不一定全，欢迎大伙儿补充）。

◇多人操作

这是最容易想到的一招。

如果有多个自然人操作同一个网络帐号，即使某个自然人出现意外，其他人依然可以继续操作该帐号。于是对应的网络身份就【不会】因为意外事故而出现静默期。

◇自动机制

“多人操作”的本质就是：几个人互相作为备胎。

另一个有点类似的招数是：用某种【自动机制】来作为备胎——在无法进行人工操作的时候，依靠自动机制进行操作。

比如某些博客平台或 SNS 平台提供“自动发文”的机制。另外还有第三方的一些服务（比如 IFTTT）可

以同时支持多种网络平台。

◇人为静默

最后，隆重推出俺本人使用的招数——人为制造【随机】“静默期”。

这个招数的好处在于：

\1. 【随机】出现的“静默期”会让网络身份的活动显得【没有规律】

\2. 如果之前经常出现静默期，一旦真实身份因为意外而无法上线，这时候导致的静默期就【不会太显眼】

◇小结：上述几招的缺陷及对比

“多人操作”的【缺点】

\1. 信任度的问题（像俺这种高危人士，没办法找到可信的人共同操作帐号）

\2. 风格的问题（比如几个人写同一个博客，风格多半不一致——读者会抗议）

“自动机制”的【缺点】

\1. 自动机制能做的事情很有限（事先写好博文自动发是可以滴，回复读者评论就没办法“事先写好”）

\2. 自动机制很容易被侦测/被识别（自动机制毕竟是“死”的，很难模拟活人）

“人为静默”的【缺点】

\1. 会降低网络身份的效率（大概就这一个缺点，这个缺点俺可以接受）

综上所述，在这3种防范措施中，俺选了最后一种。

★对“【隐式】在线时间”的探测

说完身份相关性的风险，再来说说“侦测”的话题。

前面提到说：“【隐式】在线时间”的活动痕迹不那么明显。那么，是否有办法刺探出来捏？下面就来介绍几种探测的手段（只是举例，未必全），然后再介绍防范的措施。

所有的刺探手段，都可以归纳为两大类，分别是：【主动型】和【被动型】。

◇被动探测

所谓的“被动探测”，就是【被动】地收集目标对象的网络活动痕迹（尤其是那些比较隐蔽的痕迹）。

举例1：

就拿前面的例子（俺修改博客代码）来说事儿。

如果某个有心人想要了解俺何时在修改博客代码，他/她完全可以写一个“页面抓取脚本”，循环抓取俺博客的页面；然后从抓取的页面中提取出其中的 JS 代码；最后再对比 JS 代码的【变化】（技术行话叫 diff）。一旦发现变化，就可以知道：俺在刚才的那个循环周期修改过博客代码。

举例2：

如果六扇门（公安、国安）想要收集某个目标对象的在线活动信息，并且目标对象使用的是【墙内的】网络服务。那么六扇门只需要找到对应的服务提供商，就可以从服务器上获得非常详细的帐号活动日志。

说到这里，补充一下：

通过【代理】的方式使用网络服务，可以隐藏自己真实的公网 IP，但是**【无法】隐藏自己的网络活动**

时间。一旦使用【墙内】的网络服务，朝廷方面可以很容易地拿到任意帐号在服务器上的活动日志，从而获得该帐号的“在线时间信息”。

◇主动探测

所谓的“主动探测”，就是【主动】发起一些行为，然后看目标对象是否会产生反应。

举例1：
如果目标对象有一个公开的邮箱，执行主动探测的家伙就可以往这个邮箱地址发试探性的 email，然后看看该邮件是否有【回信】，从而了解对象是否在线。
（注：这招是社会工程学的基本伎俩）

顺便说一个稍微高级点的邮件技巧——【不依赖回信】也能玩探测的把戏。
在发送的邮件中使用 HTML 格式；邮件正文中包含一个肉眼不易看见的小图片（大小为：1x1像素）；该图片位于邮件发送者自己控制的服务器上。
邮件接收者只要打开/查看这封信，就会导致图片被加载，那么在图片所在的服务器上就会留下一个访问记录。于是探测者就收集到了目标对象的在线时间了。
（注：这招是垃圾邮件发送者惯用的招数，通过这招，投放广告的商家就可以了解垃圾邮件有多大比例被受众打开/查看）

举例2：
再次拿俺博客来说事儿。
最近这几天，俺忙着改代码，没有去评论区回复读者留言，然后就看到好几条造谣的留言，用各种法子忽悠说俺被捕了。
职业五毛这么干，最容易想到的一个目的就是：制造恐慌气氛；但还有一个不太明显的目的，就是【激将法】——通过各种造谣，看看俺是否会跳出来反驳，从而了解俺是否在线。

★对上述侦测手段的防范

◇选择【靠谱的】服务提供商

俺要第 N 次唠叨：
高风险人士不要用【墙内的】服务。即使是【墙外的】，也要谨慎选择。不要用历史上有污点的公司提供的网络服务。

◇使用【非即时】的沟通方式

即时性的沟通方式（比如：文字聊天、语音聊天）暴露的在线时间实在太多了。改为【非即时】的沟通方式可以大大降低在线活动的时间跨度。
比如俺以“编程随想”这个身份活动的时候，从来不用 IM。俺更倾向于用博客评论区或者邮件进行沟通（尤其是前者）。

◇小心留意【私密的】沟通方式

前面俺提到了"基于 Email 的 主动探测"。除了 Email，还有 IM 以及 SNS 的私信，也都可以用来进行主动探测。

"Email、IM、私信"这几种沟通方式的共性就是【私密性】——其沟通内容是不公开的。

表面上看，私密性的沟通好像更安全；其实不然——这恰恰是个盲区。当网民在公开的场合（比如：BBS/论坛、评论区）发言反而会比较小心，而在私密的沟通时，更容易麻痹大意。

说到这个话题，顺便分享俺的经验：

在上个月（5月）的博文《庆贺本博荣获【更高级别朝廷认证】——谈谈近期的"帐号入侵、刷屏、钓鱼"》中提到说：

> 最近两三年，俺一直以【博客评论区】作为主要的沟通方式，邮件用得越来越少。今后还是会保持这个风格。

当时分析了【技术】层面的理由，今天来说说【非技术】方面的理由：

由于博客评论区是公开的，而且还提供 RSS 输出。所以俺只需要用 RSS 阅读器订阅相应的 feed，就可以看到博客评论区【所有的留言】。这整个过程都【不需要登录 Google 帐号】。

换句话说：俺可以在彻底静默的情况下，了解读者与俺的沟通情况。

◇保持平常心

前面俺提到了：基于"激将法"的主动探测。

要防范激将法，最彻底最根本的措施是：【调整好心态，保持平常心】。很多年轻人血气方刚，一激就跳出来了，说白了就是心态的问题。

既然说到心态，顺便聊聊性格的重要性。

在安全界（包括国内和国外）有很多技术高手，其中某些人的技术水平比俺要好得多，但是却阴沟翻船了，为啥捏？

俺观察下来，大部分情况都是因为性格和心理问题——要么太粗心、要么太冲动、要么太虚荣、要么太轻信……——这其中的任何一种心理缺陷，都足以抵消掉【所有的】技术防范措施。

[如何隐藏你的踪迹，避免跨省追捕10]：从【身份隔离】谈谈社会工程学的防范

文章目录

在今年最后一天，来发本年度最后一篇博文。

这次又隔了好多天才发文，距离上一篇博文已经20天。不过在22日，俺上博客回复了评论——所以俺静默的时间跨度【没有】超过2周。一切正常，大伙儿无须担心。

★引子

前几天，俺听说知名的翻墙软件 SSR 的作者（breakwa11）遭遇人肉。从这件事情可以看出——即便如 breakwa11 这样的技术高手，在防范人肉搜索方面也难免有疏忽之处。

所以，俺借着这个机会，继续聊“社会工程学的防范”。今天这篇以【身份的隔离性】作为切入点。

（在本系列中，关于“社会工程学的防范”，之前已经写过一篇，今天这篇是第2篇）

★“身份隔离”的重要性

在“隐匿性”这个领域，“身份隔离”是一个容易被忽视的盲点，甚至包括一些技术高手也忽略了这方面的防范。

忽视了身份隔离，可能会导致你的不同身份之间存在某种微妙的联系（关联性）。一旦有人想要追溯你的真实身份，这种（跨身份的）关联性会成为你的致命伤。

在本文中，俺首先对几个关键名词进行解释（以避免歧义）；然后，俺会介绍一些常见的“跨身份关联性”；最后，介绍一些防范的指导原则和技术手段。

★名词解释

◇身份

“身份”这个词儿很通俗，俺就不解释了。

在本文中会涉及两种“身份”，分别是：“自然人身份”和“网络身份”。

自然人身份（真实身份）

这个很好理解。所谓的“自然人身份”，也就是你的【身份证/护照】对应的身份。

对于大多数人而言，只会有【一个】“自然人身份”；少数特殊的人（比如间谍）可能会有不止一个“自然人身份”。

在本文中，为了简单起见，只讨论前一种情况。（对于大多数人而言，这已经够了）

网络身份（虚拟身份）

某些网民会给自己创造一个“虚拟身份”。这个虚拟身份在现实生活中【找不到】对应的自然人。

就以俺自己来举例——“编程随想”这个身份就是一个虚拟身份。

对任何一个网民而言，只要此人愿意，都能够创建出【多个】虚拟身份。

◇网络帐号

“网络帐号”是指那些需要登录（用户认证）的网络服务。

比如说你需要登录才能使用你的 Gmail 邮箱，那么这个邮箱帐号就是你的网络帐号。

◇“网络帐号”的【属性】

不管是哪一种网络服务，其“网络帐号”必定会包含很多属性。

俺以 Gmail 为例：

当你注册 Gmail 帐号时，会让你填写一个表格，里面有：用户名、手机号、地理位置……上述这些都是该帐号的“属性”。

◇“属性”的【信息量】？

关于“信息量”这个概念，当年在写《如何保护隐私》系列的时候，曾经聊过（参见“浏览器指纹”那篇）。

为了让大伙儿明白“信息量”这个概念，俺通过举例来说明：
假设你要定位某个人——
如果你只知道此人的“性别”，那么你只能把范围缩小到二分之一。
如果你只知道此人的“星座”，那么你只能把范围缩小到十二分之一。
如果你只知道此人的“身份证号”，那么你基本上可以唯一地锁定此人。
在这个例子中，“身份证号”的定位效果明显高于另外两个。在技术行话中，咱们称之为：“身份证号”的信息量高于另外两个属性。

下面俺列举一些常见的属性及其信息量的高低。

信息量很高
身份证号
手机号
手机 IMEI 串号
网卡 MAC 地址
邮箱地址
家庭住址

信息量中等
工作单位
毕业学校
生日（包含“年月日”）
籍贯（精确到区县）

信息量很低
性别
星座
年龄
籍贯（精确到省份）

★“网络帐号”的关联性

前面喷了一堆口水，解释了相关的概念。现在开始进入正题。
下面，俺归纳了几种常见的关联性。**这些关联性如果出现在【同一身份】的不同帐号，【没有危险】；但如果出现在【跨身份】的不同帐号，那就【非常危险】。**

◇相同的【高信息量】属性

举例：
你注册了两个邮箱，用于两个不同的身份。注册过程中需要进行短信验证，可惜你只有一部手机，于是你在这两个邮箱中设置了【相同的手机号】。如此一来，你就留下了一个潜在隐患。
万一“邮件服务器被入侵”或者“你的2个邮箱帐号被入侵”，入侵者发现这两个邮箱帐号绑定了同一个手机号，自然就明白：这两个帐号隶属同一人。

◇【操作系统】导致的关联性

当你操作网络帐号，总是需要某种客户端软件（比如：浏览器、手机 app 等）。这些客户端软件会在操作系统中留下帐号相关的某些信息。

比如说：浏览器会在 cookie 中记录帐号的信息，聊天工具也会在本地存储聊天帐号的用户名（user ID）。

如果你在同一个操作系统中使用了不同的网络帐号，一旦这个系统遭到恶意软件（木马、流氓软件）的攻击，攻击者就有可能发现这几个帐号信息，于是攻击者就意识到这几个帐号属于同一个自然人。

◇【公网地址】导致的关联性

如果几个帐号在【公网 IP 地址】上表现出某种相似性，并且次数足够多，那这些帐号就有可能存在关联性。

举例1：

假设你在某论坛上注册了3个马甲（不妨称之为 A B C）。为了避免暴露自己的公网 IP，你购买了一个 VPS 作为代理，来访问该论坛。（此时你采用的是【单重代理】）

在这个场景下，如果论坛的管理员对每个帐号的"访问者 IP"进行分析，就会发现：A B C 这三个论坛帐号【每次页面访问】都是来自同一个公网 IP。如果这个论坛管理员进一步追查，会发现这个 IP 来自某个 VPS 提供商。于是，管理员就开始怀疑——这三个论坛用户隶属同一个自然人。

举例2：

（不熟悉 TOR 原理的同学，请略过本例子）

看完上述例子，有些同学自然会想到【基于 TOR 的多重代理】。这也是本系列经常唠叨的一个招数。

现在，把上面那个例子稍作修改——你每次都使用 VPS+TOR 的方式操作这三个马甲。但这里有一个细节：你本地只安装了一个 TOR 客户端，也就是说：这三个马甲共用一个 TOR 通道。

这种情况下，如果你用三个马甲【同时发帖】，论坛的管理员还是会发现——A B C 三个帐号来自同一个自然人。为啥捏?

首先，TOR 的线路每隔10分钟变化一次，每次变化之后，出口节点就不同了。

其次，因为这三个马甲共用一个 TOR 通道。不管 TOR 的出口节点怎么变化，在同一个瞬间，这三个马甲使用的出口节点必定是相同的。如果三个马甲同时发帖，这三个帐号在论坛服务器上留下的"访问者 IP"必定是相同的。有经验的管理员就可以看出这三个帐号的关联性。

◇【时间】导致的关联性

如果两个帐号在【时间】上表现出某种规律，并且持续足够久，那这两个帐号就有可能存在关联性。

举例：

假设你在某个系统中安装了两个 IM 客户端（对应两个不同帐号）。并且你把这两个 IM 客户端都设置为：自动运行 且 自动登录。

于是，每次你开机之后，这两个帐号就会在间隔很短的一个时间段内【同时上线】；每次你关机之后，这两个帐号也会在很短的时间间隔内【同时下线】。

这种现象只要持续足够多天，基本上就可以断定这两个帐号在同一个操作系统，因此也就可以断定属于同一人。

★几个通用的指导原则

首先来聊几个【通用的】原则。

为啥俺要强调【通用】？因为这些原则与具体的技术手段无关。不管将来技术如何发展，这些原则依然能适用。

◇原则1：“虚拟身份”的划分越细越好

前面说了：每个人都可以有一个自然人身份和 N 个虚拟身份。

为啥俺认为：身份划分得越细越好？

其一，这个道理就如同船舶设计中的“水密隔仓”。一旦出了意外，只有少数隔仓进水，整艘船还不至于沉掉。

其二，身份划分得越细，每个身份包含的信息量也就越少。一旦有人想要对你的【某个身份】进行人肉，对方可分析的信息量也越少。

◇原则2：对于匿名性要求很高的身份，包含的帐号要尽量少

帐号越多，就越容易出纰漏。一旦出现纰漏，就会威胁到身份的匿名性。

◇原则3：一个“帐号”只能隶属【一个】身份

换种说法：禁止多个身份共用一个帐号。

【不同身份】使用同一个网络帐号是非常危险滴！这种做法一旦被有心人发现，别人就知道：这几个身份对应的是【同一个自然人】。

◇原则4：需要匿名的“虚拟身份”，其所属的“帐号”与其它身份的帐号必须【彻底隔离】

如何做到“彻底隔离”捏？在下面的章节，俺会介绍一些具体技术措施。

★一些具体的技术措施

◇对于匿名性要求很高的身份，其相关帐号的操作，要遵循本系列前面几篇的建议

在本系列的前面几篇，已经讲了很多隐匿性的措施。

如果你的某个身份有很高的匿名性要求，其所属帐号的操作，一定要遵循前面几篇教程的建议。比如说：软件的选择、系统的加固、存储的加密、多重代理、虚拟机隔离……

◇每个匿名身份要有一个【专属】的系统

此处所说的“系统”可以是虚拟的 Guest OS，也可以是一台物理机。（关于 Guest OS 的介绍，可以看另一个系列《扫盲操作系统虚拟机》）

你把该身份所属的帐号都放在这个系统中。这个系统中【不能】有其它身份的帐号。

◇每个匿名身份有一个专有的 TOR 通道

如果看过前面提到的“基于公网地址的关联性”，你就明白为啥要搞【专有的 TOR 通道】。

如果【多个身份】下的帐号共用一个 TOR 通道，【有可能】导致这些（不同身份的）帐号共用同一个【出口节点】。这会有潜在的隐患。

所以，为了保险起见，每一个身份至少要有一个专属的 TOR 通道。

◇对于匿名性要求很高的身份，其所属帐号【不要】绑定手机号

在天朝，手机是强制实名制的。如果你在某个网络帐号中绑定了手机号，一旦该帐号出现意外（比如帐号被盗），那么你在帐号中绑定的手机号就会暴露，进而导致你的身份暴露。

如果你因为某些特殊原因一定要绑定（比如为了短信验证），可以有两种办法：

1、找一个“虚拟手机号码”的服务，可以帮你搞定短信验证

2、去境外搞一个【不记名】的手机卡

◇对于匿名性要求很高的身份，【不要】在手机上操作相关帐号

“手机操作系统”在功能上远远不如“桌面操作系统”。很多安全防范的措施，桌面系统可以搞定，而手机系统做不到。典型的例子是“操作系统虚拟化”。

另一个让俺对手机很不爽的地方是——手机系统中有太多不开源的模块。苹果的 iOS 就不必说了。即使是 Google 的 Android 系统，虽然是基于 Linux 内核，但有大量的系统模块是 Google 自己维护的闭源软件。如果你买的是国产手机，手机制造商还会在系统中加入一大堆乱七八糟的软件。这样的系统环境，实在难以让人放心。

综上所述，高危身份所属的帐号【不要】在手机上操作。

如何用“磁盘加密”对抗警方的【取证软件】和【刑讯逼供】，兼谈数据删除技巧

文章目录

春节假期结束啦，继续来谈信息安全的话题。

先顺便说俩事儿：

\1. 虽然本文与上一篇博文的间隔超过两周，但俺上一次【网络活动】是2月7日（在回复读者评论）。所以，依然属于【正常静默】。不要担心俺 :)

\2. 上一篇博文发出后，经几位热心读者提醒，俺又对“社会工程学”那个章节补充了几点内容。

★引子

前不久正值【开博十周年】之际，俺写了一篇《为啥朝廷总抓不到俺——十年反党活动的安全经验汇总》。

今天这篇是对前一篇进行补充，主要谈——**如何对付警方**？看到俺写这样的东西，公安系统的“网监部门”和“技侦部门”，肯定又不爽了 :)

★本文讨论的范围

为了避免某些读者来抬杠，先声明一下本文讨论范围。

◇面向【高风险政治人士】

和前一篇博文类似，这篇也是写给“高风险政治人士”（类似俺这种反党人士）的教程。

如果你有可能成为【警方】的重点关注目标，大概就可以算是“高风险”啦。

（注：此处的“警方”包括天朝的“公安部、国安部”或类似机构；也包括其它国家的类似机构）

当然啦，所有的技术都是【双刃剑】——都可能被滥用。某些在网络上干坏事的家伙，也会从本文中受益。关于这点，俺也很无奈 :(

但是，俺不会因为技术存在被滥用的可能性，就停止对技术的传播和普及。

◇本文只讨论【个人电脑】，【不】讨论手机或平板

俺已经多次谈过【移动设备的危险性】。显然，“高危人士”就【不该】在手机上进行任何危险操作。

因此，本文不打算谈“手机的话题”，只讨论【个人电脑】（桌面 PC）。

（注：本文中所说的“PC”，既包括“台式机”，也包括“笔记本电脑”）

◇本文主要介绍【通用】的方法论

为了叙述方便，在涉及“操作系统”时，俺会拿 Linux 来说事儿。用 Windows 或 Mac OS 的同学，请依样画葫芦。

为了叙述方便，在谈到“磁盘加密软件”时，俺会以 TC（TrueCrypt）/VC（VeraCrypt）来举例。但本文介绍的方法，也能适用于其它磁盘加密工具（前提是——要支持【key files】这种机制）。

★本文的特色

网上已经有很多关于【磁盘加密】的教程。包括俺自己，也已经写了好多这类教程。

为了避免读者说俺炒冷饭，先说说本文的【特色】。

◇本文会更多地讨论【物理安全】

关于其它层面的防御，在《为啥朝廷总抓不到俺——十年反党活动的安全经验汇总》一文中已经谈了很多。所以，本文会更多地讨论：PC 的【物理安全】。

与“网上的骇客”不同，警方不光能【物理接触】你的 PC，甚至可以没收你的 PC（拿回去做“数据取证分析”）。因此，当你面对警方的时候，【物理安全】很重要！

（由于主要讨论“物理安全”）在本文中，如果没有特别注明，关于【操作系统】的讨论，针对的是【物理系统】（Host OS）。

◇本文会更多地讨论【紧急情况】的应对

警方不光可以把你的电子设备（电脑、手机）拿去做“数据取证分析”，还可以限制你的行动自由（拘捕/逮捕）。

所以本文会花一些篇幅讨论——如何应对【紧急情况】。也就是说，当你即将丧失行动自由【之前】，该做哪些操作，对你最有利。

请注意：在你被捕之前，**你对敏感数据销毁越彻底，警方就越拿你没办法（因此也就对你越有利）**。

★警方如何【破解】加密盘?

《孙子兵法》说【知己知彼 百战不殆】。你要防范对方，首先要知道对方有哪些招数。

关于“攻击者如何破解加密盘”这个话题，俺在5年前（2013）已经写了一篇：《[TrueCrypt 使用经验3]：关于加密盘的破解和防范措施》。

虽然那篇是针对 TrueCrypt 加密盘，但其中提到的几种破解手段，对其它格式的加密盘（VeraCrypt，LUKS，BitLocker ……）也适用。

考虑到某些新读者没有看到那篇旧博文，俺把那篇中提到的几种破解手段再【简述】如下。

◇利用“加密算法本身的漏洞”——基本上【不可能】

【成熟的】磁盘加密软件，使用的“对称加密算法”肯定也是【成熟】滴。

以 TC（TrueCrypt）/VC（VeraCrypt）为例，它支持的“对称加密算法”是“Rijndael（AES）、Twofish、Serpent”。

当年美国国家标准局（NIST）公开招标21世纪的新一代加密算法标准（叫“高级加密标准”，简称“AES”）；经过多次淘汰，有5个算法成为【最后一批】候选者，上述的3个算法也在这5个当中；最终是 Rijndael 正式被选中成为 AES（所以，现在所说的 AES 也就是指 Rijndael）。

由于这三个算法是历经淘汰的最后一批候选者，算法本身已经被世界各国的密码学专家仔细检查过啦，【几乎不可能】出现算法层面的漏洞。

引申阅读：在如下博文中介绍了磁盘加密常用的几种“对称加密算法”。

《[TrueCrypt 使用经验1]：关于加密算法和加密盘的类型》

◇对密钥的穷举——基本上【不可能】

刚才说啦——成熟的磁盘加密软件，使用的加密算法肯定也是【成熟】滴。

既然如此，算法使用的密钥，其“密钥空间”肯定足够大。因此在足够长时间内（几代人），并考虑到摩尔定律带来的算力增长，都【不可能】对密钥进行穷举。

以 AES256 为例，其密钥有【256 比特】，所有可能的密钥数量是【2的256次方】。这个数有多大捏?

```
2^256 =
115792089237316195423570985008687907853269984665640564039457584007913129639936
```

◇对【弱】密码/口令的【暴力猜解】——常见招数

这是最常见的手段，专门针对【弱密码】（weak password）。

由于本文要对付的是【警方】。他们比一般的骇客具备更多的资源。比方说，天朝的警方已经建立了专门用于“暴力猜解密码”的【服务器集群】，可以大大提升密码猜解的效率。普通骇客用“单机”无法暴力猜解的密码，有可能被警方搞定。

但是，**这招很好对付**——你只需要引入【key files】作为加密盘的“认证因素”，就可以让警方的暴力猜解变得【不可能】。

在《[TrueCrypt 使用经验2]：关于加密盘的密码认证和 KeyFiles 认证》一文中，俺已经谈过“key files”的“用途”和“注意事项”。关于“注意事项”，再说一次：

\1. key files 要使用“二进制文件”，【不要】用“文本文件”

\2. key files 最好是【随机生成】（TC 和 VC 自带了“随机生成 key file”的功能）

\3. key files 的文件尺寸【至少64字节】

外行的读者可能会觉得“64字节”太小了。其实“64字节”的随机文件，已经足够对抗穷举。一个字节有8比特，64字节有512比特。因此，64字节的随机内容，其可能的数量是“2的512次方”——这已经大大超过刚才提到的“AES 256 密钥空间”了。

给大伙儿秀一下这个数有多大。

```
2^512 =
134078079299425970995740249982058461274793658205923933772356144372176403007354697680187429816690342769003185818648605085375388281194656994643364900608409
6
```

另外，像 TC/VC 这两款磁盘加密软件，最多只提取【每个】key file 开头的一兆字节（1MB）参与密钥生成。所以，随机生成的 key file，每个都【不必】超过 1MB。

◇【窃取】加密盘的【密钥】——常见招数，本文的重点

前面提到的几种攻击手法，都比较好解决（容易防御）。比较难对付的是这招——“攻击者盗取密钥”。所以，对这招的防范是本文的重点内容之一。

下面，俺分多个章节，对这个话题进行展开。

◇用【审讯】的方式获得“密码和 key files”——常见招数，本文重点

前面的“攻击手法”都属于【技术手段】。如果警方尝试了技术手段而不可得，那自然会采用【审讯】的方式。

在《为啥朝廷总抓不到俺——十年反党活动的安全经验汇总》一文，俺已经谈到了【如何对付审讯】（包括酷刑）。

原理很简单——如果你的加密盘采用了【随机生成】的 key files 作为“认证因子”。你只需要在“丧失行动自由（被捕）”之前【彻底销毁】key files；如此一来，连你自己都【不可能】打开加密盘。酷刑也就失去意义啦。

原理说起来蛮简单，但落实起来会涉及到很多细节。比如说：如何【既彻底又快速】地销毁 key files？下面单列一个章节来谈。

★为啥加密盘的【密钥】如此关键?

◇【密钥】是啥玩意儿?

（考虑到大部分读者基本不懂密码学，俺尽可能通俗地解释一下【密钥】是啥玩意儿）

密钥说白了就是一段数据，可以用来加密或者解密。磁盘加密软件用的算法都是“对称加密算法”。这种算法的特点是：“加密的密钥”和“解密的密钥”是【同一个】。假设你用“密钥 K”把一段“明文 P”变成“密文 C”，那么你同样可以用 K 把 C 变为 P。

除了“对称加密算法”，还有一类算法叫做“非对称加密算法”——“加密的密钥”和“解密的密钥”是【不同】滴。这类算法与本文无关，就不展开介绍啦。

◇“密钥”（key）与“密码/口令”（password）有啥【区别】?

很多不懂技术的网友经常混淆“密钥”和“密码”，其实这两者是【完全不同】滴：

\1. 密钥（key）直接参与加密/解密运算滴。

\2. 密码（password）【不】参与“加密/解密”运算，它的作用是保护“密钥”。

◇为啥说密钥【很重要】?

首先，

如果攻击者能通过某种方式【窃取】密钥（key），就可以直接用密钥（key）【解密】加密盘存储的数据。

也就是说——在【不知道】你的密码和 key files 的情况下，也能打开加密盘。

其次，

一旦创建了加密盘，其密钥就固定了。即使你修改了加密盘的“认证因子”（“密码”或“key files”），密钥依然【不变】。

一旦攻击者拿到某个加密盘的密钥，不论你如何修改密码，攻击者依然可以打开这个加密盘。

综上所述，

“密钥”的重要性【超过】“密码”和“key files”。

◇密钥如何【生成和存储】？

（本小节谈【技术原理】。比较懒的同学，可以跳过这个小节。不影响后续阅读）

密钥的生成

简单地说，是通过某种复杂的数学运算产生出来（密码学术语叫“密钥生成函数”）。在“密钥生成函数”的生成过程中会用到【哈希运算】和【多次迭代】。（注：“哈希运算”也叫“散列运算”，洋文叫“hash”，这篇博文有 hash 的扫盲）

如果你用图形界面的 TC 或 VC 创建一个新的加密盘，在创建过程中，软件会提示你：尽可能快速并随机地移动鼠标。这么干是通过你随机移动鼠标，让软件能收集到足够多的随机数据（足够的【熵值】）

为啥要搞如此复杂的数学方式来创造密钥捏？（通俗地说）要达到——为了让密钥尽可能随机，使得攻击者无法猜测密钥，也无法缩小密钥的分布范围。

密钥的存储

加密盘的密钥，会以【加密】的形式存储在加密盘的“头部或尾部”。而你设置的“密码”或“key files”，就是用来加密密钥的。

所以，当你要打开加密盘时，如果输错了“密码”或“key files”，加密盘的密钥解不出来，自然就打不开。

★如何防止攻击者【窃取】密钥?

◇确保物理系统（Host OS）的【纯洁】

关于这点，无需多言。

如果你的物理系统本身已经中招（感染了恶意软件），那么，其它所有的防范措施皆是空中楼阁。

关于这个话题，可以参见之前的系列教程：

《如何防止黑客入侵》

◇在电脑的 BIOS 设置“开机密码”和“硬盘锁密码”

这2点也是老调重弹了——俺刚在《为啥朝廷总抓不到俺——十年反党活动的安全经验汇总》中已经提到过。

设置这两个密码，是为了防止攻击者（警方人员）在物理接触你的电脑之后，在（主引导扇区的）boot loader 中植入恶意软件，从而盗取你全盘加密的“密码或密钥”。

（注：在【不】依赖外部启动介质的情况下，主引导扇区是【不可能】加密滴。因为至少要有一个【明文】的程序，让你输入“全盘加密的”解锁密码，并执行解锁的动作）

从技术上讲，你也可以把 boot loader 放到某个外置的【可启动 U 盘】中，每次开机都先插入该 U 盘。但这么干太麻烦，而且会引来额外的风险。因为 U 盘没法像笔记本那样设置【硬盘锁】。警方的技侦人员如果能偷偷拿到你的这个启动 U 盘，同样可以植入恶意软件。

◇【禁用】“虚拟内存”

现代操作系统都有“虚拟内存”的机制。对 Windows 而言，叫做“页交换文件”，对 Linux 而言就是“swap 分区”。

“虚拟内存”的作用在于——当物理内存比较紧张的时候，把一些不经常访问的“内存页”转储到硬盘，就可以腾出物理内存来存放新的内容。

一旦启用了“虚拟内存”，也就存在某种可能性——【敏感加密盘的密钥】有可能会残留在“虚拟内存”（也就是硬盘上）——显然，这增加了你的风险。

有些同学可能会问：既然已经“全盘加密”，不论是“swap 分区”还是“页交换文件”，都被加密了。为啥还要担心这个风险？

下面，俺来回答一下——

前一篇博文之所以建议：在“全盘加密”的基础上再搞“敏感加密盘”，就是因为这两个东西的【密级不同】——“敏感加密盘”的密级【更高】。

如果你在物理系统（Host OS）中启用了“虚拟内存”，你就把“敏感加密盘”的安全等级【降低到】与“全盘加密”一样的水平。换句话说，一旦攻击者能够突破全盘加密，也就【有可能】从“虚拟内存”中找到“敏感加密盘的密钥”，从而突破“敏感加密盘”。

◇谈谈“关机、休眠、待机”三者的安全差异

在《为啥朝廷总抓不到俺——十年反党活动的安全经验汇总》一文中，俺提到：

> 当你要【长时间】离开自己的电脑——应该【关机】（shutdown）；而【不要】“休眠”（hibernation）或“待机”（suspend，stand by）

现在来解释一下：

休眠（hibernation）

所谓的“休眠”（hibernation），说白了就是把【整个物理内存】保存到硬盘中的“休眠文件”。**这非常危险！**危险的原因参见“虚拟内存”那个小节的讨论。另外，如果你在【未卸载】“敏感加密盘”的情况下让系统休眠，那么“休眠文件”中【必定包含】“敏感加密盘”的【密钥】。

待机（suspend，stand by）

如果你已经禁用了“虚拟内存”和“休眠功能”，仅仅是让系统“待机”，这种情况下，加密盘的密钥【不会】出现在硬盘上，但还是有风险——如果你在待机之前【没有】卸载加密盘，加密盘的密钥还在物理内存（RAM）中。

如今的 DRAM 内存条在断电后依然有【数据残留】。在你离开电脑的这段时间里，如果警方的技侦人员对（正在待机的）电脑进行【冷启动攻击】（洋文叫“cold boot attack”，维基百科的链接在“这里”），可以直接拿到【内存条】中的数据，再从中分析出“加密盘密钥”。

（注：断电之后，内存条中的数据【不会】立即消失，会有一个残留时间。温度越低，残留时间越长。在正常的室温下，内存条的数据残留时间【不超过】10分钟）

引申阅读：

在2013年的旧博文，俺已经介绍了【冷启动攻击】的原理和手法。

《[TrueCrypt 使用经验3]：关于加密盘的破解和防范措施》

“冷启动攻击”这个手法已经有10年以上的历史。很多笔记本厂商在硬件和 BIOS 方面做了一些防范措施。可惜的是：前几个月（2018年9月），国外安全研究人员曝光了“冷启动攻击”的新进展——可以突破所有主流 PC 厂商（与“待机”相关）的防御。

《研究人员警告现代计算机都易受冷启动攻击 @ Solidot》

《New modification of the old cold boot attack leaves most systems vulnerable @ Ars Technica》

关机（shutdown）

既然“休眠”和“待机”都靠不住，你只剩下一个选择——【关机】。

前面提到——正常的室温下，内存条的数据残留时间【不超过】10分钟。也就是说，只要关机超过10分钟，就【不可能】进行“冷启动攻击”了——因为内存条中的数据全都消失了。

补充：锁定屏幕

如果你仅仅是“锁定屏幕”，其风险情况与“待机”类似——你的 PC 会被【冷启动攻击】搞定。

补充：混合睡眠模式

经热心读者提醒：Windows8 开始引入了【混合睡眠模式】。即使在“待机”状态下也会保存一部分内存到硬盘上。

因此，你应该彻底禁止“休眠功能”（具体参见微软官网的“这个链接”）。

◇禁用 FireWire DMA

“FireWire”是某种硬件接口规格（俗称“火线接口”，也叫“IEEE 1394 接口”）。这玩意儿支持 DMA（direct memory access）方式直接操作【物理内存】。所以攻击者如果能物理接触你的电脑，可以在你的电脑上插入某个特定 FireWire 设备，然后利用 DMA 的方式直接读取整个系统的内存。因此，这招也叫做“DMA attack”

这种攻击方式不仅仅是理论上可行，前几年就已经有人搞了傻瓜化的“DMA 攻击工具”。

如果 BIOS 有 FireWire DMA 相关选项，应该禁掉；或者在操作系统层面禁用相关的功能（通过“重编译 Linux 内核”）。

由于这个方式也是针对【运行状态下】盗取内存。所以，【关机】可以对付这种攻击方式。

★“机械硬盘”与“固态硬盘（闪存）”【删除数据】的差异——损耗均衡技术

◇“损耗均衡技术”是啥玩意儿?

为了介绍后面的章节，需要先扫盲一个小知识——损耗均衡技术（洋文叫“wear leveling”）。

考虑到本文的大部分读者不是 IT 硬件方面的专业人士，俺尽量通俗地扫盲一下。

所谓的“固态硬盘”（SSD），是以【闪存】作为永久性存储介质。平时咱们用的“U盘”也是以【闪存】作为介质。（为了打字省力，本文以下章节凡是提到“闪存介质”，就是指——“采用闪存作为存储介质”的各种东西的【总称】）

“闪存介质”相比“机械硬盘”的一大【缺点】是——写入次数的上限比较低。如果某个存储单元的数据反复更改（反复写入），达到上限后，这个存储单元就坏掉（变为“不可用”）。

“闪存介质”的厂商为了解决这个问题，采用了“损耗均衡技术”（详细介绍请看维基百科的“这个链接”）。比如某个文件，原先存储在“a单元”，当你更改了文件内容并保存，新的内容就不在“a单元”了，而是把新内容存到一个用得最少的“b单元”。此时，“a单元”里面依然有该文件的【旧内容】。

请注意，这个技术是由【硬件层面】（闪存介质的控制器）实现的，对操作系统【不可见】。

◇“损耗均衡技术”【不利于】彻底删除数据

聪明的读者，看完前一个小节，已经发现问题所在了。

很多时候，为了彻底删除文件，要用“垃圾数据（随机数据）”去覆盖文件的原有内容，从而让原有内容【不可恢复】。这个招数对“机械硬盘”是 OK 的，但对“闪存介质”就不灵了。因为“损耗均衡技术”使得你覆盖的内容存到的【别的】单元，根本【没达到】你想要的目的。

而且俺刚才也说了——“损耗均衡技术”是由硬件层面的存储控制器实现，对操作系统不可见。所以，用【软件方式】难以证明某个存储单元是否【真正被覆盖了】。

◇对【闪存介质】，如何彻底删除文件?

综上所述，要想在“闪存介质”中彻底删除文件，你需要换一种思路，大致如下：
\1. 先简单删除该文件
\2. 用垃圾数据（随机数据）填满该存储介质的剩余空间——要 100% 填满。
（这样才能确保——原先保存过敏感数据的单元，已经被垃圾数据覆盖掉）

注意事项：要填满的是【物理硬盘】的“剩余空间”，而不是【分区】的“剩余空间”。
举个例子：
假设你的固态硬盘有多个分区。敏感文件在“分区1”，你把敏感文件删除后，光填满“分区1”的剩余空间是【不够】滴。
正确做法是——你要把【所有】分区的剩余空间都填满，才能确保之前保存过敏感数据的单元，确实被垃圾数据覆盖掉了。

◇闪存厂商提供的工具，是否可信/可靠?

某些闪存介质的厂商会提供一些配套的工具/软件，据说可以提供【彻底删除】的功能。
对此，俺表示谨慎的怀疑。因为无法验证其效果。
所以，为了保险起见，还是老老实实去【填满剩余空间】吧。

◇小结

正式因为“闪存介质”要彻底删除文件，如此之麻烦。所以那些看重隐私保护的同学，更加应该早早用上【全盘加密】。
一旦你在“闪存介质”上使用【全盘加密】，每一个物理存储单元中的数据都是【密文】——“损耗均衡技术”就不再是障碍啦。

★如何【快速且彻底】销毁加密盘的数据?

◇为啥要强调“快速且彻底”?

为啥俺强调【彻底】？——只有彻底地销毁数据，才能对付警方的【取证软件】；
为啥俺强调【迅速】？——警方想要拘捕你，当然不会留给你从容的时间。比如说：当警方人员正在撞门的时候，留给你的时间可能连1分钟都不到。

◇“销毁”的含义

本章节所说的“销毁”是指——让任何人【包括你自己】再也无法得到加密盘里面的数据。

◇为啥“哄骗”不可行?

本文发出后，某些热心读者提到了“哄骗”的招数。简单说就是：你假装销毁，但其实并没有。这样做，将来你自己还能继续打开加密盘。

俺认为：这种方法【不够】可靠，**甚至是危险的想法**。理由如下：

如果你是警方的重点关注对象，或者你是重大案件的关键人物，警方肯定会动用【刑侦和审讯】方面的高手来参与办案。

在这种情况下，你【别想】太容易哄骗对方。除非你自己也受过【严格的】“反侦查和反审讯”方面的训练。

但试问：有几个人具备这个条件?

基于上述理由——只有当你在紧急情况下，无法彻底销毁敏感数据，再考虑“哄骗”的招数。

◇【错误】方式举例

首先来说说【反面】教材。所谓“错误的方式”指的是——【不够彻底】或者【不够快速】的方式。

错误方式1——用“普通的删除命令”删除加密盘里面的文件

（所谓的“普通删除命令”，比如：Linux 下的 `rm` 命令，Windows 下的 `del` 命令）

首先，这种做法【不】彻底（会被“取证软件”恢复出来）；其次，如果加密盘中的文件很多，这种做法太慢。

错误方式2——用“专门的【擦除】命令”彻底删除加密盘里面的文件

（所谓的“专门的擦除命令”，比如 linux 下的 `shred` 命令）

这种方式比较彻底，但是【慢】。如果文件很多，就非常慢。

（注：shred 支持【多轮反复擦除】，可以防范专业取证人员对机械硬盘的“剩磁分析”，但也导致其速度很慢。即使你把 shred 设置为“只擦除1轮”，速度还是慢）

错误方式3——快速格式化

这种方式速度快，但是不彻底——被“快格”的分区，数据全都在。

错误方式4——彻底格式化

这种方式算是比较彻底，可惜【太慢】了。如果格式化的分区有好几个 GB，你就慢慢等吧。

◇【正确】方式——彻底删除 key files

关于这招，已经提到过多次了。不但本文提到，之前在多篇博文中也提到。

这种方式有两种实现——“软删除”和“硬删除”。

如果 key files 存储在 PC 的硬盘上，用软件方式（比如：shred 命令）彻底擦除内容。为了讨论方便，以下称之为：“软删除 key files”。

如果 key files 放在【外部】的存储介质（比如：U盘、MMC/SD 卡、...），你可以先把存储卡插到 PC 上，然后用软件干掉 key files（依然是“软删除”）；但你还可以用【物理方式】直接破坏存储介质（这种情况称为——“硬删除 key files”）。

由于 key files 都很小（通常小于 1MB），所以“软删除 key files”肯定是【既快速又彻底】。

提醒：“软删除”的【局限性】

如果是【机械硬盘】，“软删除”是 OK 的；但如果是【固态硬盘】，“软删除”【不】保险。原因请参见本文前面章节介绍的【损耗均衡技术】。

至于“硬删除”的方式

由于存在几种不同情况，后面俺用一个单独的章节来讨论。

◇【正确】方式——破坏加密盘的【密钥存储区】

前面俺聊“密钥的存储”，已经提到——磁盘加密软件为了能打开加密的数据，必须把密钥（以【加密形式】）存储在某个地方。通常是存储在加密盘的【头部或尾部】。

当你输入认证因子（“密码”和“key files”），加密软件根据“认证因子”进行一系列数学运算，然后从密钥存储区中解密出【加密盘的密钥】。如果你的认证因子输错了，密钥就解不出，加密盘自然打不开。

所以，如果你用“随机数据”把加密卷的【头部和尾部】两者都覆盖，就足以【彻底破坏】整个加密盘。

密钥存储区通常很小（不到 1KB）。为了保险起见，咱们把覆盖范围扩大一千倍（从 1KB 变为 1MB），彻底覆盖加密盘最开头的 1MB 和最末尾的 1MB，肯定就能毁掉“密钥存储区”了。

再来看“速度”——哪怕老式的机械硬盘，对头尾各写入 1MB 的数据，也可以在1秒内完成。所以这招属于【既彻底又快速】。

提醒：本招数的【局限性】

如果是【机械硬盘】，可以用这招；但如果是【固态硬盘】，这招【不】保险。原因请参见刚才介绍的“损耗均衡技术”。

◇【运行状态】下的权衡

（所谓的【运行状态】指的是——出现“紧急情况”时，你的 PC 处于【开机运行】，并且你正在操作它）

在这种情况下，显然【可以用】“破坏加密盘”的方式。

至于“删除 key files”是否可用，取决于你的 key files 存储放在哪里?

\1. 如果你的 key files 存储在 PC 上，也可用“【软】删除 key files”的方式。

\2. 如果你的 key files 位于【外部】存储介质（比如：U盘、MMC/SD 卡、...），并且这个存储介质【没有】插在电脑上，就应该用“【硬】删除 key files”的方式。

◇【关机状态】下的权衡

当你的 PC 处于【关机状态】，一旦碰到紧急情况，你已经来不及开机并启动系统。所以，“破坏加密盘”显然不可行；同样的道理，此时“软删除 key files”也不可行。

因此，在【离线状态】下，你只有唯一的选项——“硬删除 key files”。

◇结论

从上述两种状态的权衡，很自然就可以得出结论——**key files 必须位于你【身边】的存储介质中。**只有这样才能保证——你在【各种情况】下都能【快速且彻底】地销毁加密数据。

为了让大伙儿一目了然，放一个对照表：

	运行状态，机械硬盘	**运行状态，固态硬盘**	**关机状态**
【硬】删除 key files	YES	YES	YES
【软】删除 key files	YES	不保险	NO
破坏加密盘的【密钥存储区】	YES	不保险	NO

★如何【物理破坏】存储卡?

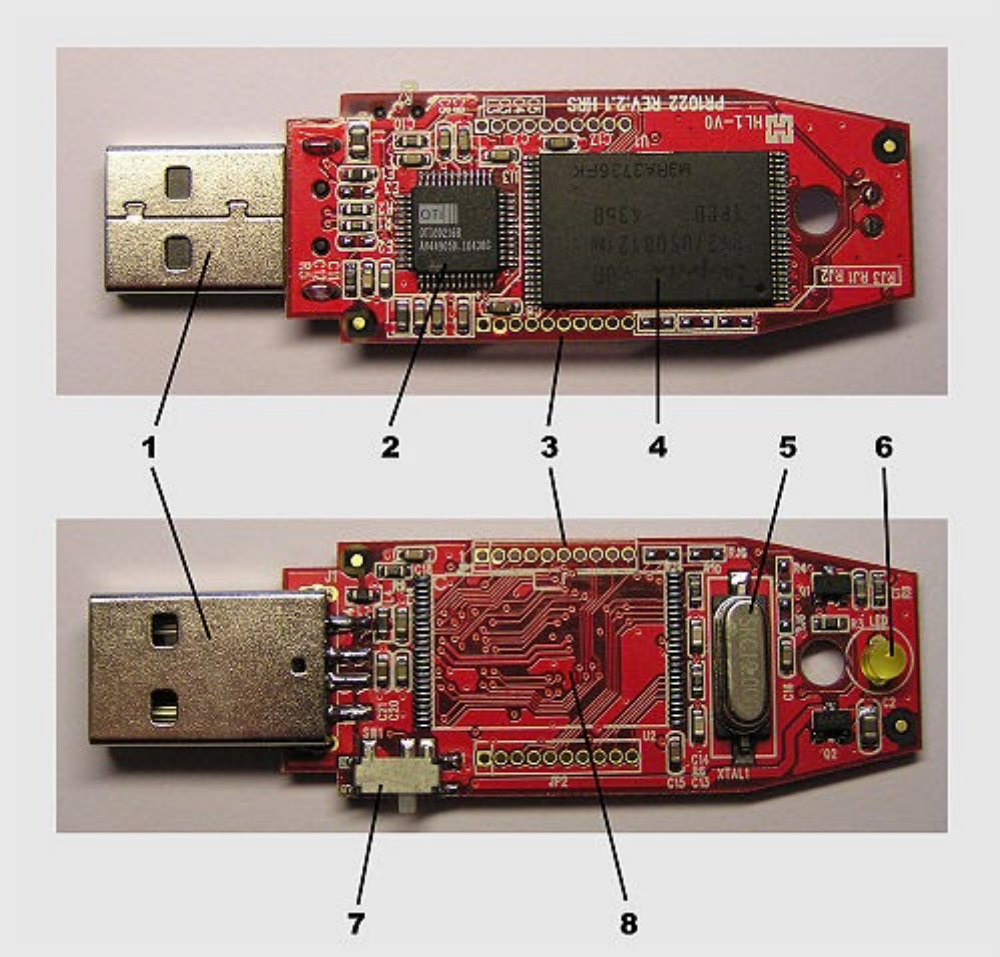
好，现在来谈“硬删除 key files”的方式——也就是“从物理上毁掉存储卡”。

（注：本章节所说的“存储卡”包括：U盘、SD卡、MMC卡...）

再次唠叨：这种“物理破坏”的方式比较粗鲁，是【紧急情况下】的无奈之举。如果你的时间比较从容，应该用前面提到的软件方式——删除敏感文件后，通过填充垃圾数据，塞满整张存储卡的剩余空间。

◇U 盘的内部结构

为了方便讲解，从维基百科剽窃了一张照片，并附上相应的说明。



1 插头
2 存储控制器
3 测试接点
4 **【闪存芯片】**
5 石英振荡器
6 发光二极管（LED）
7 写入保护开关
8 预留给第二颗存储器芯片的空间

◇物理摧毁的关键——破坏【闪存芯片】

对存储卡而言，除了【闪存芯片】，其它的都是浮云。

因为其它所有的部件全被毁掉，只要【闪存芯片】还完好，就可以通过专用的设备，读取出里面保存的数据。

◇破坏“闪存芯片”——【彻底】的方式

关于这个话题，俺看过一些资料，以及网上的讨论。权衡下来，能够在“短时间”（1分钟内）物理破坏“闪存芯片”的方式，大致有两种：

电磁方式——微波炉

把存储卡放入微波炉并启动。

这么干，不但能毁掉存储芯片，可能你的微波炉也会跟着报废。但在紧急情况下，一个微波炉又算得了什么？

机械方式——砸烂

如果你身边有锤子、扳手（或诸如此类的工具），对准结构图当中那个“傻大黑粗”的家伙狠狠砸下去，砸烂为止。

由于“闪存芯片”比较硬，还可以考虑用某个尖锐的东西作为辅助（学过基础物理学的应该明白——这可以增加压强）。就比如说，把一个钉子架在“闪存芯片”上，然后再用锤子敲击钉子，更容易击穿芯片的外壳。

但如今现代化的家庭或办公室中，要想找一个钉子还真不太容易。那么，啥东西可以作为钉子的替代品捏？俺列几个替代品作为参考（欢迎大伙儿补充）

> 螺丝刀（尤其是【小型的十字】螺丝刀）
> 瑞士军刀中某些尖锐的部件（如下图）

（某款瑞士军刀的示意图——几乎每种款式的瑞士军刀都带有尖锐的工具）

◇破坏“闪存芯片”——【不太彻底】的方式

现在介绍【不】那么彻底的方式。也供高风险人士参考。

在紧急情况下，如果你身边没有微波炉也没有砸烂芯片的工具，还有一招是——用冲水马桶把存储卡冲掉。如今的存储卡都比较小，至少不会卡在马桶里。

但要强调的是——这招【不彻底】，因为存储卡浸泡在普通液体（只要不是强酸），即使长达几天

时间，还是有可能恢复出数据。

如果你用了这招，那主动权就转到警方这边——考验他们是否愿意到粪坑里去找存储卡了。

★预备和演习

如果你确实认为自己是【高风险人士】，本文讲述的这些东西，你【不要】光看看而已。要先做好准备工作，甚至来一次演习（彩排）。

否则真的到了紧急情况，你在慌乱之中很可能会出错。（这是本文最后的忠告）

俺博客上，和本文相关的帖子（需翻墙）：

《为啥朝廷总抓不到俺——十年反党活动的安全经验汇总》
《文件加密的扫盲介绍》
《扫盲文件完整性校验——关于散列值和数字签名》
《TrueCrypt 使用经验》（系列）
《扫盲 VeraCrypt——跨平台的 TrueCrypt 替代品》
《扫盲 dm-crypt——多功能 Linux 磁盘加密工具（兼容 TrueCrypt 和 VeraCrypt）》
《扫盲 Linux 逻辑卷管理（LVM）——兼谈 RAID 以及“磁盘加密工具的整合”》
《文件备份技巧：组合“虚拟加密盘”与“网盘”》
《如何保护隐私》（系列）
《如何防止黑客入侵》（系列）
《扫盲 Linux＆UNIX 命令行——从“电传打字机”聊到“shell 脚本编程”》
《扫盲操作系统虚拟机》（系列）

[如何保护隐私0]：为啥写这个话题？

文章目录

★引子

前不久美国爆发了“棱镜门”丑闻。这事儿再度激起了大伙儿对隐私的关注。应多位读者的要求，俺今天来聊一下 IT 领域的隐私防范。本来想一篇博文搞定，写着写着发现这个话题挺大的，只好再挖一个坑。

俺挖的坑有点多，还望大伙儿谅解哦。一方面是要照顾到不同需求的读者；另一方面是俺想挑战自己写博的能力。另外，最近2周有大量职业网评员到俺博客留言，这是对俺的巨大鼓励 :)

★本系列的目标读者群

本系列面向的是不太懂技术的读者，尽量不涉及太深奥的技术细节。

本系列的重点是“隐私保护”，而不是“隐匿性保护”（关于“隐匿性保护”，请看俺另一个系列《如何隐藏你的踪迹，避免跨省追捕》）。

对于大部分普通网友，只需要注重“隐私保护”就够了；“隐匿性保护”是面向那些从事危险网络行为的网民（比如像俺这样，**公开发布**敏感政治言论，**公开**抹黑党国）。

★本系列的目录

为了方便阅读，把本系列帖子的目录整理如下（需翻墙）：

[如何保护隐私1]：如何选择软件和服务？

文章目录

★IT 机构的类型

本节所说的“IT 机构”指的是跟软件相关或者跟互联网相关的组织或机构（包括“商业公司”和“非营利组织”）。IT 机构可以根据如下几个维度进行分类。

◇提供服务 VS 提供软件

根据这个维度，可以分为如下三类。

第1类

仅仅提供基于互联网的服务，不提供软件。

因为不提供软件，所以这类 IT 机构提供的服务往往是纯 Web 服务（比如专门提供恶意软件扫描的 VirusTotal）。

第2类

仅仅提供应用软件，不提供基于互联网的服务（比如某些开发单机游戏的公司）

第3类

两者都提供（比如：Google、Apple、微软……）

按照收集信息的能力排序：第3类 > 第2类 > 第1类

为啥 第2类 大于 第1类？因为软件是安装到你的操作系统中的，（从技术上讲）不但可以访问你的文件系统，而且可以获取你操作系统中的很多信息。相对而言，纯 Web 的应用就安全得多。

所以，**能用纯 Web 搞定的，尽量不要装软件。**

举例：

比如俺用微软的 One Drive 网盘分享电子书。OneDrive 网盘同时支持客户端软件和纯 Web。因为有了纯 Web 支持，而且功能够用，所以俺从来不装微软的网盘客户端。

◇非商业（非盈利性） VS 商业（盈利性）

如果某个商业公司具有很大的用户群，那么该公司【很有可能】收集用户的行为。

原因在于：数据挖掘技术已经非常成熟，收集大量用户的行为，有可能带来商业利益，这对商业公司具有很大的诱惑力。

举例——Netflix

可能很多国内的网友没听说过这家公司。它是世界上最大的在线影片租恁服务商。

Netflix 很擅长数据收集，也很擅长数据挖掘。它不光记录每一个用户看了哪些影片，而且会记录用户看某个影片时，在哪个时间点按了“暂停”，在哪个时间点按了“快进”。

因为它的用户数足够多，再加上它有足够好的数据挖掘算法。就可以预测某个还没有上映的美剧是否会火爆。

比如 Netflix 的管理层连视频的内容都没看到，就砸下1亿美元购买《纸牌屋》的版权。因为 Netflix 的数据挖掘算法预测，此片必火（如果你觉得很神奇，可以看《Wired》的详细报道）。

说了这么多，就是想表明一点：商业公司有盈利压力，所以对收集用户信息具有天然的偏好。相对而言，非营利组织就好很多——它们或许也会收集，但肯定没商业公司这么大的热情。

所以，**尽量用非营利机构的软件和服务。**

有些读者可能有一个错觉——以为非营利机构搞出来的东西不如商业公司。

其实这是不一定滴！

比如非营利组织 Mozilla 开发的 Firefox 在功能上要好于微软的 IE。
比如非营利的维基百科是全球最好的在线百科（远远好于百度百科，百度百科的【中立性】很差）。

◇国外 VS 国内

再来说说最后一个维度。
俺博客的大部分读者都是天朝网民。所以大伙儿还需要考虑 IT 机构是国内还是国外。
大伙儿都知道，天朝是个一党专制的国家。所以，国内的 IT 机构会受到朝廷的胁迫。朝廷让他们干啥，他们就必须干啥（否则就别想在天朝混）。
而且咱们朝廷搞了一个金盾工程（维基百科的词条在"这里"）。这个金盾工程会收集并监控国内网民的各种网络行为（比如：论坛、邮件、聊天、网盘、等等）。

举例——QQ 聊天
腾讯作为 IM 市场的长期垄断者，早就被朝廷盯上了。据说腾讯的聊天服务器上部署了专门的监控模块。如果你经常在 QQ 群中发布一些不和谐的言论，就会被朝廷盯上。
顺便插一句：经常有读者来信，询问俺的 QQ 号。在此郑重声明：俺一直不用 QQ 的。像俺这种长期抹黑党国的危险分子，用 QQ 简直是找死。

有些读者会反问：那几个美国大公司不是也卷入到"棱镜门"丑闻吗？
俺的观点是：如果你是天朝的网民，你不用担心美国政府的监控。
首先，美国政府对你【没有】司法管辖权；其次，美国政府关注的重点是不同的——对于天朝网民发表的敏感政治言论，美国政府通常不感兴趣。
引申阅读：
《中美政府信息监控的差异——"棱镜门"丑闻随想》

显然，**国内 IT 机构的危险性远远大于国外的。**

◇小结

根据上述的对比，可以得出如下结论——尽量使用国外的、非营利的 IT 机构提供的软件和服务
举例：
比如选浏览器的话，Firefox（在隐私保护方面）比"Chrome、IE、Safari"更靠谱，因为 Mozilla 是非营利机构，而"Google、微软、苹果"都是商业公司。

★应用软件的类型

刚才说了，如果某个功能可以用纯 Web 搞定，就尽量不要装软件。但是有很多东西是纯 Web 搞不定，你不得不装软件。这时候如何防范捏？请看如下的对比。

◇开源软件 VS 闭源软件

所谓的"开源"（洋文叫 Open Source），就是说该软件的源代码是公开的，可以被网民获取。
通常而言，"开源软件"好于"闭源软件"。因为源代码公开，如果软件带有后门或偷窥隐私的行为，就**比较容易**被发现。（但是也有例外，曾经发生过开源软件的后门长期未被发现的案例，比如 Borland 的 InterBase 后门）
相对而言，"闭源软件"由于没有公开源代码，要发现其后门或偷窥隐私的行为，就比较难（只能通过监控软件行为来发现）。
所以，**尽量使用"开源软件"以防止后门和偷窥隐私。**

对比举例——磁盘加密软件

俺拿两款比较有名的磁盘加密工具（TrueCrypt 和 BitLocker）来说事儿。

TrueCrypt 是开源软件，而 BitLocker 是微软提供的商业软件（不开源）。由于 BitLocker 不开源，是不是内置了后门，就说不清楚啦。微软自己肯定不承认有后门。但是 N 年前就有安全专家怀疑，NSA（美国国安局）已经在微软的加密工具中设置了后门。

显然，TrueCrypt 在保护隐私方面要好于 BitLocker。

引申阅读：

对 TrueCrypt 感兴趣的读者，可以看俺写的扫盲教程：

《TrueCrypt——文件加密的法宝》

《扫盲 VeraCrypt——跨平台的 TrueCrypt 替代品》

◇单机软件 VS 网络软件

所谓的“单机软件”就是说：该软件不访问网络，使用该软件不需要联网；反之，“网络软件”在使用的时候会访问网络。

在保护隐私方面，单机软件好于网络软件。

对比举例——输入法

早期的输入法，大都是单机软件；如今的输入法很多都成为网络软件（一个很流行的功能是“在线同步个性化词库”）。

那些在线同步词库的输入法就会泄露你的隐私。因为输入法的提供商可以根据词频分析，推测你平时经常输入哪些内容。如果他们愿意，还可以进一步分析你的职业、你的喜好、等等。

反之，单机版的输入法，软件提供商就无法拿到你的“词频”信息。

所以，**如果软件本身的用途跟网络无关，那就尽量选“单机软件”。**

◇绿色软件 VS 非绿色软件

所谓的“绿色软件”，就是无需安装，也无需依赖管理员权限的软件。

反之，“非绿色软件”要么需要安装，要么需要管理员才可以运行。

前几年写过一个《如何防止黑客入侵》的系列，第一篇的标题就是《避免使用高权限用户》。在那篇博文中，俺详细介绍了“高权限用户的危害”，此处就不再啰嗦了。

要避免使用高权限用户，技巧之一就是：尽量用绿色软件。

◇小结

根据上述的对比，可以得出如下结论：优先使用开源的，绿色的软件。如果软件本身的用途跟网络无关，那就尽量选“单机软件”。

★上网的类型

说完了 IT 机构类型和应用软件类型，再来说说上网的类型。

◇有中心——C/S 型（Client-Server）

所谓的 C/S 型，也就是你的电脑充当 Client，你通过跟 Server 通讯来进行信息的交互。大部分互联网行为都属于这一类。

举例：

浏览网页，Server 就是网站的 Web 服务器
收发邮件，Server 就是邮件提供商的邮件服务器
传统 VPN 翻墙，Server 就是 VPN 服务器
……

C/S 型的缺点在于，中央服务器知道太多用户的信息。

◇无中心——P2P 型（Peer to Peer）

自从 P2P 下载普及之后，很多网友都开始听说 P2P 一词。P2P 和 C/S 的主要差别在于——不需要固定的中央服务器。所以 P2P 又可以称为“无中心”或“去中心化”。

举例：

最近两年开始流行的比特币（BitCoin）就是典型的无中心化。它把所有的货币交易历史都存储在每一个客户端上。

相比 C/S 过度依赖中央服务器，“去中心化”的好处在于，你的信息分散在许许多多不同的网络节点中——这就增加了收集隐私的难度。

◇半中心——P2P 与 C/S 混合型

所谓“半中心”的混合型，就是既有 P2P 也有 C/S。

举例：

Skype 聊天。当你登录的时候，是基于 C/S 型（需要从 Skype 的服务器上验证你的帐号）。在语音聊天的时候是基于 P2P 型（可以直接跟对方进行语音传输，无需经过 Skype 的服务器）。

◇小结

按照收集信息的能力排序：“有中心”>“半中心”>“无中心”。

对比举例——翻墙工具

传统的 VPN 翻墙和代理翻墙是“有中心”的。假设你长期使用同一个 VPN 提供商，万一该提供商记录你的上网历史，你的隐私就泄露啦。

而 TOR 跟 I2P 是无中心的。而且中转节点会随着时间频繁变化。因此，即使中转节点偷窥你的网络流量，看到的也是残缺不全的片段。

顺便补充一下：TOR 和 I2P 本身都是多重代理。只有最后一个节点（术语叫“出口节点”）可以看到你的访问的网站；其它节点看到的都是强加密的流量。

虽然“无中心”很好，而且也很符合互联网精神（互联网当初的设计，就是“去中心化”的）。可惜的是，如今的很多网络服务（尤其是 SNS）缺乏成熟的“无中心”替代品。

“棱镜门”丑闻曝光之后，某个热心的老外搞了个“粉碎棱镜”的网站，里面列举了不少“去中心化”的网络服务（包括：Email、IM、SNS、等）。有兴趣的同学可以去瞧一瞧。

[如何保护隐私2]：关于浏览器的基本防范（上）

文章目录

本系列的前一篇博文，俺介绍了选择“软件”和“服务”的一般性原则。

从这篇开始，俺会花3篇博文的口水来介绍一下浏览器的基本防范。因为在个人软件中，虽然有很多软件跟隐私关系密切，但“浏览器”的牵扯的面最广。

★浏览器本身导致的隐私问题

◇浏览器自身如何泄漏隐私？

先来谈浏览器**【自身】**是如何泄漏隐私的。这里强调的是**【自身】**，也就是说，跟浏览器外围的插件、扩展、cookie 等等无关，是浏览器软件自己把用户隐私给泄漏了。

很多国产的浏览器（至少包括：奇虎的“360浏览器”、腾讯的“QQ浏览器”、百度的“百度浏览器”）自身都存在严重的隐私问题。它们会收集用户的上网行为（比如：你在啥时候访问了啥网站），并且把你的上网行为保存到它们自家的服务器上。

一旦你用了有危险的浏览器，即使你在其它方面的防范做得再好，也是白搭。因为浏览器是上网冲浪的关键，浏览器出问题会导致你的隐私防御全线崩溃。

稍微跑题，说一下当年轰动武林的“3Q大战”。（因为前两天有读者在博客留言中争论此事）

这件事说白了就是【狗咬狗 一嘴毛】，“奇虎、腾讯”都不是好东西，它俩不但充当朝廷的走狗，而且不择手段收集用户隐私。但是“3Q大战”的积极意义在于：两只癞皮狗互咬，把两边的家丑都曝光了。咱们普通网民作为旁观者，可以更清楚这些国产软件厂商的嘴脸。

再强调一下，有问题的国产浏览器，绝不止这两家。

◇应该如何选择？

刚才说了，国产浏览器猫腻很多。那么国外的浏览器，是不是就很安全捏？也未必。

大伙儿不妨复习一下前一篇博文。在那篇博文中，有两个原则可以参考：

原则1：开源 好过 闭源

（开源软件，即使有后门或流氓行为，也比较容易被发现）

原则2：非营利组织 好过 商业公司

（对于大型商业公司，收集用户信息会带来商业利益，所以商业公司有收集隐私的热情）

国外的浏览器，名气大且使用广泛的，主要有三款：Firefox、Chrome（Chromium）、IE。其中，Firefox 和 Chromium 是开源的，Chrome 和 IE 是闭源的。

很多人都误以为 Chrome 是开源的。虽然 Chrome 是基于开源的 Chromium，但是 Chrome 包含闭源的模块。所以严格来讲，Chrome 只能算【部分开源】。

根据“原则1”（开源角度）：这几款浏览器的隐私安全性如下

Firefox = Chromium > Chrome > IE

再来看这几个浏览器的后台。IE 的 后台是微软**公司**、Chrome/Chromium 的后台是 Google **公司**，Firefox 的后台是 Mozilla **组织**。

根据“原则2”（商业角度）：这几款浏览器的隐私安全性如下

Firefox > Chromium = Chrome = IE

◇隐私方面，为啥 Firefox 优于 Chrome?

先声明，本小节介绍的是两款浏览器在“隐私保护”方面的对比；至于安全性（防范骇客入侵）方面的对比，请看俺的另一个系列《如何防止黑客入侵》。

俺博客的读者中，有相当数量的 Chrome 用户。估计很多人会为 Chrome/Chromium 打抱不平。所以，再从商业模式的角度分析一下，为啥 Chrome/Chromium 的**隐私保护**不如 Firefox。

稍微熟悉 Google 的同学应该知道，Google 的大部分利润（90% 以上）来自于“在线广告”这个业务。要想做好“广告”这门生意，最关键的是——做到“精准投放广告”。而要想实现精准投放，自然要收集用户信息（只有当 Google 对某网民足够了解，才能知道该网民会对哪些广告感兴趣）。而这就涉及到隐私问题。

反之，Mozilla 是非营利组织，不像 Google 那么依赖广告客户。所以 Firefox 就可以放开手脚去防范隐私泄露。

举例——浏览器对 Do Not Track 的支持

Do Not Track（简称 DNT，中文叫“请勿追踪”），维基百科的词条在“这里”。是一项浏览器功能，用来告知网站，用户不希望被追踪。

以下是各个知名浏览器加入 DNT 功能的时间表。

Firefox	2011年1月	第一个支持 DNT 的浏览器
IE	2011年3月	
Safari	2011年4月	
Opera	2012年2月	
Chrome	2012年11月	在主流浏览器中，支持最晚！ 比 Firefox 晚了将近2年

用 Chrome 的同学都知道，Chrome 是频繁发布新版本的，而且 Google 的研发力量是很强大滴。那么，为啥 Chrome 拖了这么长时间才支持 DNT 功能，显然不是因为技术原因，而是因为商业原因——Google 不愿意得罪广告客户。

◇小结

综上所述，在隐私保护方面，Firefox 是首选。

补充说明：在本文发布5年之后，俺又写了另一篇博文（如下），继续讨论 Firefox 与 Chrome/Chromium 的对比。

《弃用 Chrome 改用 Firefox 的几点理由——关于 Chrome 69 隐私丑闻的随想》

★“浏览历史”导致的隐私问题

本章节提及的“浏览历史”，至少包括如下几个方面的信息：

网址的历史

下载的历史

页面缓存

各种 Cookies

◇“浏览历史”如何泄漏隐私？

举例：

假如你是一个年轻男网民，喜欢访问成人色情网站。然后捏，你又不注意清除浏览器的历史缓存。或许有一天，你的女朋友或者你的父母无意中打开你的浏览器，就会发现你的癖好。那你就尴尬了。

◇“隐私浏览模式”的用处

如今，几款主流的浏览器（Firefox、Chrome、IE）都已经支持“隐私浏览”功能。

你在“隐私浏览模式”下进行的上网行为，浏览器【不会】保存相应的“浏览历史”。当你退出“隐私浏览模式”或者关闭了浏览器之后，这些信息就不见了。

◇“隐私浏览模式”的局限性

插件导致的问题

但是，“隐私浏览模式”并【不是】足够安全滴！如果你的浏览器安装了插件（比如 Flash），可能会导致“隐私模式”【部分失效】。

因为浏览器插件不受浏览器的控制。所以，即使在“隐私模式”下，某些插件还是可能会留下上网痕迹。

本文的下一章节会介绍，插件如何导致“隐私模式”部分失效。

书签导致的问题

另一个局限性是：对于大部分浏览器（比如 Firefox、Chrome），隐私浏览期间对书签的修改，会被保留下来。可能有些同学没有意识到这里面的风险。俺举个例子。

举例：

你在隐私浏览模式下，访问某个敏感的网站。浏览网页的时候，你不下心点了 `Ctrl + d` 组合快捷键，然后浏览器就把当前页面加入书签。退出“隐私浏览模式”之后，这个书签还在哦。然后，假如你周围的人一不留神看了你的书签，就看到你曾经上过某敏感网站。

◇更彻底地解决浏览历史——使用虚拟机

假如你对“隐私浏览模式”的局限性，很在意。解决方法之一就是：使用操作系统虚拟机。

如果你不熟悉操作系统虚拟机，请先看俺写的《扫盲操作系统虚拟机》系列博文。

使用虚拟机，大致的操作步骤如下：

\1. 先安装虚拟机软件，然后装一个虚拟操作系统（Guest OS）
\2. 在 Guest OS 中安装好上网相关的软件（比如浏览器、插件、等）
\3. 在没有访问任何网站之前，先做一个 Guest OS 的快照（这是一个干净的快照）
\4. 在这个 Guest OS 里面上网
\5. 上网结束后，回退到这个干净的快照

通过上述步骤，就可以避免在电脑中留下任何上网的痕迹。由于虚拟机是操作系统级别的，即使是浏览器插件，也不会留下痕迹。

★浏览器插件导致的隐私问题

◇“插件”与“扩展”的区别

先来扫盲一下插件和扩展的区别（连很多 IT 技术人员都把这两者混为一谈）。所谓的插件，洋文叫“plugin”；所谓的扩展，洋文叫“extension”。两者的区别如下：

插件

在功能上，插件通常是用来渲染页面里的 `<object>` 或 `<embed>` 标签。

插件通常用操作系统的本地代码（也叫“原生代码”）编写，可以调用操作系统的 API。形式上，插件以动态库（Windows 上就是 DLL 文件）的方式，加载到浏览器的进程内。由于使用本地代码编写，插件通常依赖于特定的操作系统（不同系统的插件不能混用）。

举例：
Flash 插件
媒体播放器插件
PDF 插件
Java 插件
各种网银控件

扩展

扩展，顾名思义，是用来扩展浏览器自身的功能。所以，扩展可以调用浏览器自身的 API，但是扩展通常不能调用操作系统的 API。

一般来说，扩展是跟操作系统无关的。比如 Firefox 的大部分扩展，既可以用于 Windows 平台的 Firefox，也可以用于 Linux 和 Mac OS X 的 Firefox。

举例：
俺推荐的 GreaseMonkey，就属于扩展。

◇插件的隐私问题

刚才说了，插件是用本地代码编写的，调用的是操作系统的 API。所以，插件的行为浏览器是无法控制的。相对而言，扩展调用的是浏览器的 API，所以扩展的行为，浏览器是比较可控的。

下面，俺就拿 Flash 的 cookie 来举例，让大伙儿看看插件导致的隐私问题。

举例——Flash 的 cookie

平时大伙儿提到 cookie，说的都是浏览器的 HTTP Cookies；但是除了浏览器，插件也可能有自己的 cookie——比如最流行的插件 Flash 就有自己的 cookie 功能。

Flash Cookies，专业术语叫做“Local shared object”。网页中加载的 Flash 对象可以利用这个功能，在你的操作系统中保存一些信息。所以，Flash Cookies 跟“HTTP Cookie”一样，都可能带来某些隐私问题。

从上述例子可以看出，插件也可以在操作系统中留下自己的一些痕迹（不妨称之为“插件的 Cookie”）。而且“插件的 Cookies”比“HTTP Cookies”更讨厌的地方在于——它是独立于浏览器 cookie 的。这就导致如下两个隐私问题：
\1. 当你在浏览器中清空 HTTP Cookies 的时候，插件的 cookie 【不会】受影响；
\2. 浏览器的“隐私模式”会限制 HTTP Cookies 的永久保存，但是浏览器无法限制插件的 cookie（这就是俺刚才提到的“隐私浏览模式”局限性）

注：从 Flash 10.3 开始，Adobe 已经通过技术手段，解决了 Flash Cookie 与浏览器的整合问题。也就是说，浏览器在清除 HTTP Cookies 的同时，也清除了 Flash 的 Cookies。**但是，其它的插件【有可能】依然存在上述问题。**

◇如何防范插件留下的痕迹?

办法由两个:

办法1

最直接的办法就是:你的浏览器不要装任何插件。

办法2

如果你不得不装一些插件,那么还有一招——利用虚拟机的快照功能。这个方法刚刚讲过,就不再啰嗦了。

今天先聊到这里,下一篇接着说说浏览器的 HTTP Cookie。

[如何保护隐私3]:关于浏览器的基本防范(中)

文章目录

今天这篇,主要介绍浏览器 Cookie 方面的防范。

★“cookie”是啥?

本文所说的“cookie”,指的是浏览器相关的 cookie(有时候也叫“HTTP cookie”)。

浏览器 cookie 的主要功能是:帮助网站保存一些小片段的信息。比如,你曾经在自己的浏览器上登录过某个论坛,下次你再打开论坛的登录页面,你会发现用户名已经帮你填好了,你只需要输入口令即可。那么,这个登录页面是如何知道你上次登录用的账户名捏?奥妙就在于:该网站在你的浏览器端保存了一个 cookie,里面包含了你上次登录使用的帐号名称。

★Cookie 的技术实现

本章节面向【懂技术】的读者。如果你不太懂技术,可以略过本节,直接进入下一章节,以免浪费时间。

◇网站如何设置 cookie(写操作)

步骤1

当你在浏览器中点某个书签、或者在浏览器地址栏输入某个网址,浏览器会向对应的网站发起一个 HTTP 请求(术语是 HTTP Request)。

步骤2

然后,网站的服务器收到这个 HTTP 请求之后,会把相应的内容(比如网页、图片、等)发回给浏览器(这称为 HTTP 响应,术语是 HTTP Reponse)。

如果网站想设置 cookie，就在发回的 HTTP Response 中，包含一个设置 cookie 的指令。举例如下：

> Set-Cookie: user=xxxx; Path=/; Domain=www.example.com

在上述例子中，设置了一个 cookie。这个 cookie 的【名】是 `user`；cookie 的【值】是 `xxxx`；cookie 绑定的【域名】是 `www.example.com`

步骤3

浏览器在收到这个指令后，就会在你的电脑中存储该 cookie 的信息。

◇网站如何获取 cookie（读操作）

假设过了几天之后，你再次访问上述的 `www.example.com` 网站（在上次的访问中，已经被设置过 cookie 了）。这时候，浏览器发现该网址已经有对应的 cookie，就会把 cookie 的信息放在 HTTP Request 中，然后发送到网站服务器。具体的指令如下：

> Cookie: user=xxxx

网站服务器拿到这个 HTTP Request 之后，就可以通过上述信息，知道 cookie 的【名】与【值】。

★Cookie 的特点

◇存储信息量小

cookie 在洋文中的意思就是：小甜饼、曲奇饼。这个单词其实已经暗示了 cookie 技术所能存储的信息量是比较小滴。

从刚才的技术实现机制可以看出，cookie 只能用来存储纯文本信息，而且存储的内容不能太长——因为 Cookie 的读写指令受限于 HTTP Header 的长度。

但是，cookie 的信息量虽小，能耐却很大哦。请看下面的例子。

举例

比如某个网站上有很多网页，每个网页上有很多广告。该网站想要收集：每一个访客点击了哪些广告。

由于这些信息量比较大，直接存储在 cookie 里可能放不下。所以，网站通常是在 cookie 中保存一个**【唯一的用户】**标识。然后把用户的点击信息（包括在哪个时间点击哪个广告）都存储在服务器上。

下次你再访问该网站，网站先拿到 cookie 中的用户标识，因为这个标识具有唯一性，那么就可以根据该标识，从网站服务器上查出该用户的详细信息。

◇绑定到域名和路径

从上述的实现机制可以看出，cookie 是跟 HTTP Request 对应的网址（域名和路径）相关的。

所以，不同域名的网站设置的 cookie 是互相独立的（隔离的）。这一点由浏览器来保证，以确保安全性。

补充一下：cookie 绑定的域名可以是**【小数点开头】**的。举例如下：

> Set-Cookie: user=xxxx; Path=/; Domain=.example.com

这个指令设置的 cookie，可以被 `example.com` 的**【所有】**下级域名读取（比如：`www.example.com` 或 `ftp.example.com`）。

★Cookie 的类型

◇第一方 Cookie VS 第三方 Cookie

首先来说说“第一方 Cookie”与“第三方 Cookie”的区别，因为这跟隐私的关系比较密切。
要说清楚这两者的差异，俺来举个例子。

举例
打个比方，你上新浪去看新闻，并且新浪的网页上嵌入了阿里巴巴的广告（假设新浪的页面和嵌入的广告都会设置 cookie）
那么，当你的浏览器加载完整个页面之后，浏览器中就会同时存在新浪网站的 cookie 和 阿里巴巴网站的 cookie
这时候，新浪网站的 cookie 称为“第一方 Cookie”（因为你访问的就是新浪嘛）
相对的，阿里巴巴的 cookie 称为“第三方 Cookie”（因为你访问的是新浪，阿里巴巴只是不相干的第三方）

◇内存型 VS 文件型

根据存储方式的不同，分为两类：基于内存的 Cookie 和 基于文件的Cookie。基于内存的 cookie，当浏览器关闭之后，就消失了；而基于文件的 cookie，即使浏览器关闭，依然存在于硬盘上。和隐私问题相关的 cookie，主要是第二类（基于文件的Cookie）。

★Cookie 有啥正经用途?

今年的315晚会，央视猛烈抨击了 cookie 的隐私问题，搞得好像 cookie 是洪水猛兽一般。CCAV 对 cookie 的妖魔化宣传，典型是用来吓唬不懂技术的外行。
其实捏，cookie 是有利有弊的。cookie 之所以应用这么广泛，因为它本身确实是很有用的。请看下面的几个例子。

◇举例1——自动登录

目前很多基于 Web 的邮箱，都有自动登录功能。也就是说，你第一次打开邮箱页面的时候，需要输入用户名和口令；过几天之后再来打开邮箱网页，就不需要再次输入用户名和口令了（比如 Gmail 和 Hotmail 就是这样的）。
为啥邮箱可以做到自动登录，就是因为邮箱的网站在你的浏览器中保存了 cookie，通过 cookie 中记录的信息来表明你是已登录用户。

◇举例2——提供个性化界面

比如某个论坛允许匿名用户设置页面的字体样式和字体大小。
那么，该论坛就可以把匿名用户设置的字体信息保存在 cookie 中，下次你用同一个浏览器访问该论坛，自动就帮你把字体设置好了。

◇小结

一般来说，有正经用途的 cookie，大都是“第一方 Cookie”；至于“第三方 Cookie”，大部分是用来收集广告信息和用户行为的。

★Cookie 如何泄漏隐私?

cookie 就像一把双刃剑，有很多用途，但也有弊端。一个主要的弊端就是隐私问题。

◇举例1

假如你同时使用 Google 的 Gmail 和 Google 的搜索（很多 Google 用户都这么干）。当你登录过 Gmail 之后，cookie 中会保存你的用户信息（标识你是谁）；即使你在 Gmail 中点了注销（logout），cookie 中还是会有你的用户信息。

之后，你再用 Google 的搜索功能，那么 Google 就可以通过 cookie 中的信息，知道这些搜索请求是哪个 Gmail 用户发起的。

可能有些同学会问：Gmail 和 Google 搜索，用的是不同的域名，如何共享 cookie 捏?

俺前面有介绍过，某些 cookie 绑定的域名是以小数点开头的，也就是说，这类 cookie 可以被所有下级域名读取。

因为 Gmail 的域名是 `mail.google.com`，而 Google 搜索的域名是 `www.google.com`。所以这两者都可以读取绑定在 `.google.com` 的 cookie!

注：俺拿 Google 来举例是因为俺博客的读者，大部分都是 Google 用户。其实不光 Google 存在此问题，百度、腾讯、阿里巴巴、奇虎360、等等，都存在类似问题（这几家都有搜索功能，也都有自己的一套用户帐号体系）。

◇举例2

很多网站会利用 cookie 来追踪你访问该网站的行为（包括你多久来一次，每次来经常看哪些页面，每个页面的停留时间）

这样一来，网站方面就可以根据这些数据，分析你的个人的种种偏好（这就涉及到个人隐私）。

请注意：利用 cookie 收集个人隐私的把戏有很多，俺限于篇幅，仅列出上述两例。

★浏览器禁用 Cookie 的几种方法

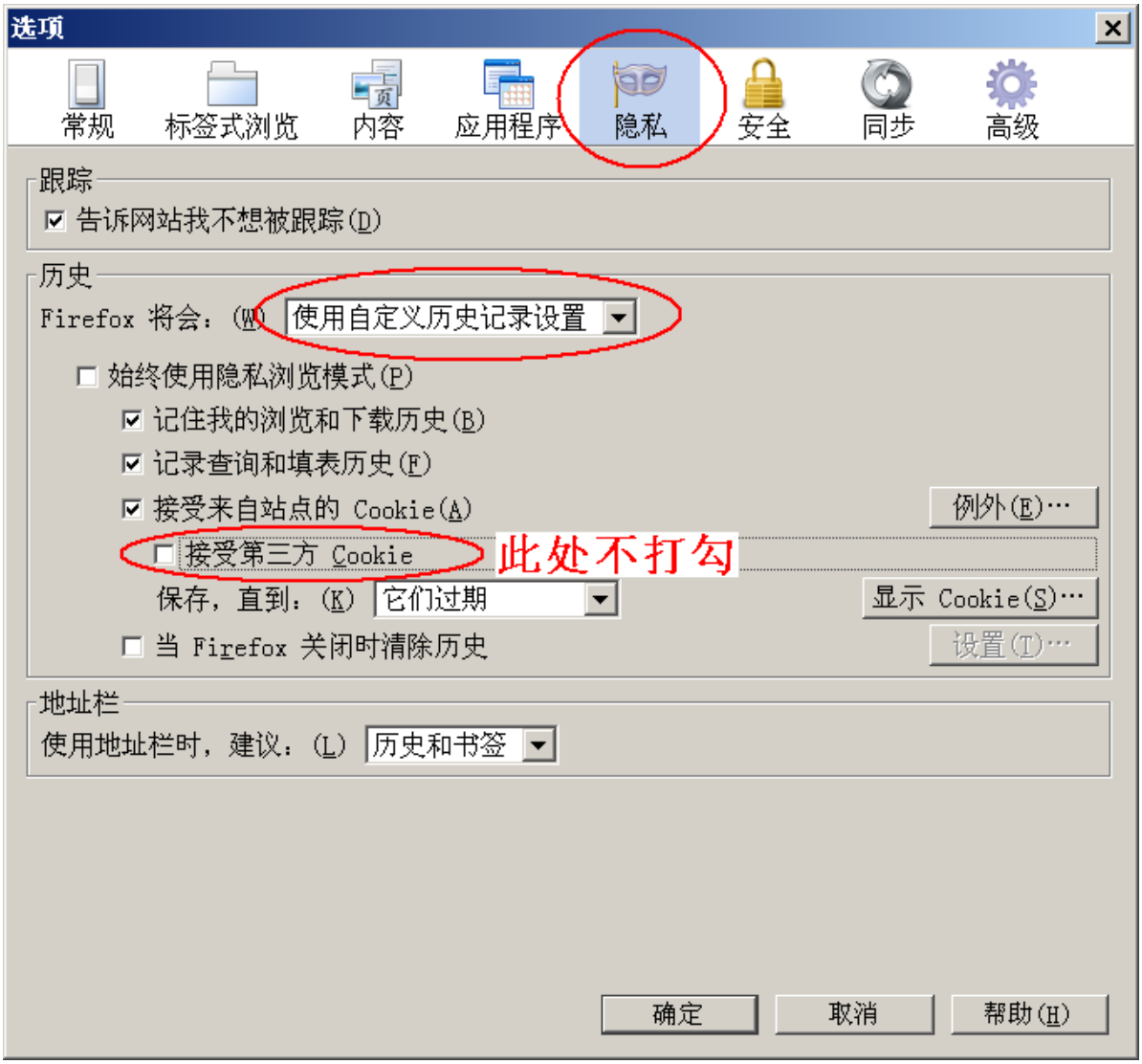
在本系列的前一篇，俺已经分析了：Firefox 是主流浏览器中，隐私方面比较靠谱的（完全开源、没有商业背景）。相对于 Firefox，Chrome 只是【部分开源】，而且 Google 的商业模式太依赖广告，不会为隐私保护而得罪广告主；至于 IE，根本不开源，自身的安全性也不够。

所以，本小节下面的内容，主要拿 Firefox 来说事儿。如果你是 Chrome 的粉丝，不想换 Firefox，也没关系。这两款浏览器的某些功能大同小异，你可以参考本章节的介绍，然后在 Chrome 上依样画葫芦。

◇只禁用【第三方】Cookie

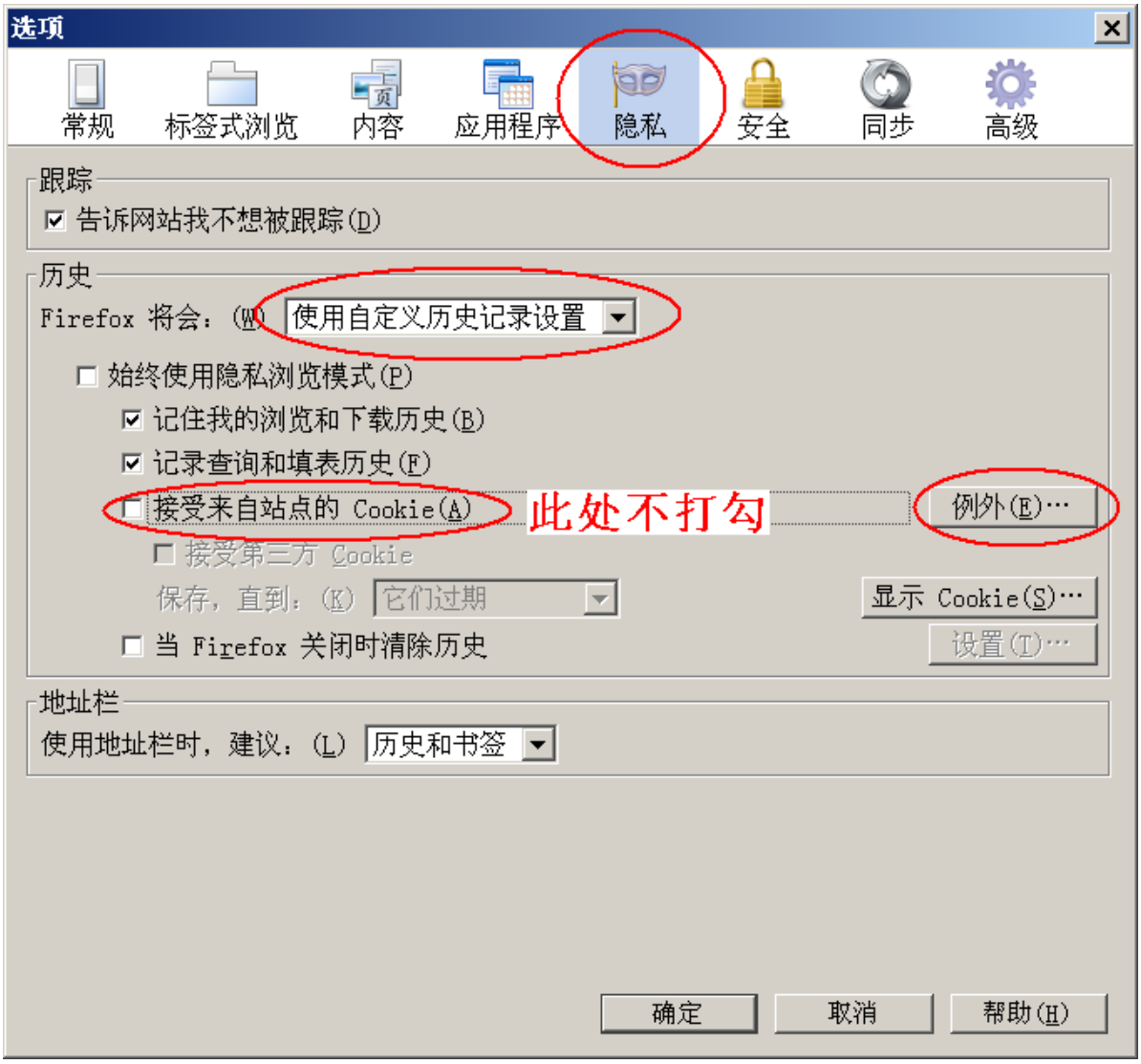
前面说了，大部分“第三方 Cookie”都【不是】干正经事儿的。所以，最简单也是最宽松的设置，就是光禁用“第三方 cookie”。

配置方法和截图如下：
在 Firefox 主菜单点“工具”，再点“选项”。在弹出的选项对话框中，选“隐私”标签页。
在界面的第一个下拉框中选择“使用自定义历史记录设置”
选完下拉框，你会看到一个复选框叫“接受第三方 Cookie”，请**【去掉】**这个复选框的打勾。



◇禁用【所有】cookie，但保留【例外】（白名单）

这种方式比前一种更严格。你只保留某些你信任的网站的 cookie，其它网站统统禁用。
配置方法和截图如下：
在 Firefox 主菜单点“工具”，再点“选项”。在弹出的选项对话框中，选“隐私”标签页。
在界面的第一个下拉框中选择“使用自定义历史记录设置”
选完下拉框，你会看到一个复选框叫“接受来自网站的 Cookie”，请**【去掉】**这个复选框的打勾。
然后到该复选框右侧，点“例外”按钮，把你信任的网站域名输入进去。



◇始终启用“隐私浏览模式”

关于“隐私浏览模式”，在本系列的前一篇已经介绍过了，此处不再啰嗦。

相比前两个招数，这招最严格。在隐私浏览模式下，浏览器关闭之后，期间所有的 cookie 都消失。

但是，这样设置也可能带来一些不方便之处（安全性和方便性通常是截然对立）。你可能要先尝试一段时间，看看自己能否忍受这种模式。

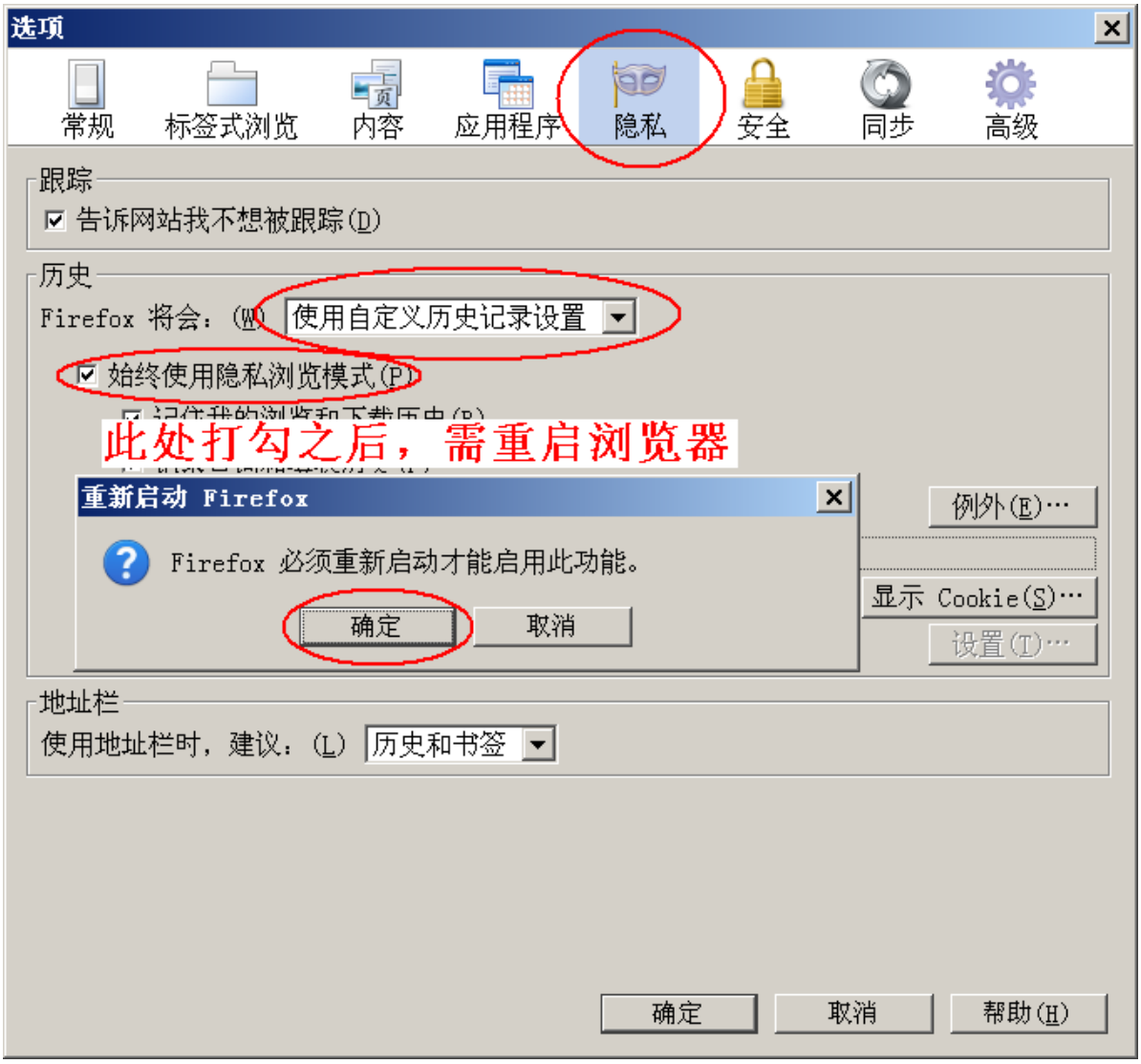
配置方法和截图如下：

在 Firefox 主菜单点“工具”，再点“选项”。在弹出的选项对话框中，选“隐私”标签页。

在界面的第一个下拉框中选择“使用自定义历史记录设置”

选完下拉框，你会看到一个复选框叫“始终使用隐私浏览模式”，请**【勾上】**这个复选框。

打勾之后，Firefox 会提示你重启浏览器。



◇小结

刚才介绍的几招，都是针对单个浏览器 。大部分情况下是够用了。但是某些特殊情况，还是会搞不定。

比如：你经常用 Gmail，而且依赖于 Gmail 的自动登录。这时候，你就【不应该】禁用 `.google.com` 域名下的 cookie（禁用了就无法自动登录 Gmail）。

但是，你在用 Google 搜索的时候，又不希望让 Google 知道你是谁。咋办捏？请听下回分解——如何隔离浏览器。

[如何保护隐私4]：关于浏览器的基本防范（下）

文章目录

本系列的上一篇介绍了 Cookie 相关的知识。今天这篇把浏览器剩余的几个常见话题聊一下。

★如何隔离浏览器?

◇为啥要隔离浏览器?

举例1

你经常使用 Gmail，并且依赖于 Gmail 的自动登录功能。这种情况下，你就【不能】禁用 `google.com` 域名对应的 cookie（禁用了就无法自动登录）。但是，如果你不禁用 `google.com` 域名对应的 cookie，当你使用 Google 的搜索功能时，Google 会知道你的搜索偏好（这涉及到隐私）。

举例2

虽然浏览器插件会导致隐私问题（本系列前面的博文有提到），但有些浏览器插件是你不得不安装的。比如你想看在线视频，不得不装 Flash 插件；比如你使用网银，不得不安装某些登录的控件。

当你学会了【浏览器隔离】这个招数，就可以搞定上述难题，鱼和熊掌兼得。下面，俺分别介绍几种不同的隔离方法。

◇多浏览器

这是最简单的隔离方法——使用不同类型的浏览器（比如同时 Firefox 和 Chrome）。

对于刚才的第一个例子。你可以在“A浏览器”里面登录 Gmail，并允许“A浏览器”存储 `google.com` 域名的 cookie；然后你用“B浏览器”进行 Google 搜索。

只要你能确保——从来不在“B浏览器”中登录任何 Google 帐号，那么“B浏览器”的 cookie 中就【不会有】你的用户标识。所以你用“B浏览器”进行 Google 搜索的时候，Google 不知道你是谁。

◇多实例

但是有些用户就喜欢某款浏览器，不喜欢混用。那就可以考虑“多实例”的招数。目前的三大浏览器中，Firefox 和 Chrome 支持多实例，IE 不支持。啥是多实例捏？稍微解释一下。

所谓的浏览器多实例，洋文称之为“Multiple Profiles”。

不论是 Firefox 还是 Chrome，默认安装的时候，只有一个实例（Profile）。和浏览器相关的各种信息，包括：插件、扩展、外观（皮肤）、页面缓存、cookie、等等，都存储在这个实例中。

反之，如果使用多实例，每个实例都具有独立的插件、独立的扩展、独立的外观（皮肤）、独立的页面缓存、独立的 cookie、等等。不同实例之间是相对隔离的，不会互相影响。

对于 Chrome，要特别提醒一下：

Chrome 同时支持“Multiple Profiles”和“Multiple Accounts”。千万别把这两者搞混了。即使你配置了多个 Accounts，依然在同一个实例里（有兴趣的同学可以看 chromium 官网的“这个链接”）

关于多实例的配置，一年前写《如何防止黑客入侵》系列的时候，已经介绍过了（请翻墙看“这里”），今天就不再浪费口水了。

◇多用户

万一你喜欢的浏览器是 IE，而 IE 又不支持“多实例”，咋办捏？招数还是有滴，那就是【多用户】。

你可以创建多个操作系统用户，然后在不同的操作系统用户中分别运行同一款浏览器。所有主流的操作系统都会对系统用户的资源进行隔离。所以，你在不同的操作系统用户中运行的浏览器，也是互相隔离的（包括“插件、扩展、书签、浏览历史、cookie、缓存”，都不会互相影响）。

如果你用的是 Windows 操作系统，可以使用“快速用户切换”的功能，在不同用户的桌面之间切换。但是有个缺点：一次只能看到某一个用户运行的桌面，其它用户运行的软件看不到。其实有一个小技巧，可以在同一个桌面运行多个用户的软件。具体请看这篇博文中的“多用户浏览器共享同一个桌面的技巧”。

◇多虚拟机

大部分场合用上述三招基本就能搞定了。但有捏，有时你会碰到一些比较变态的网银控件，一定要用系统管理员才能安装，有的控件甚至做到了“驱动级”。对于这种变态的控件，即使用“多用户”的隔离方案，也行不通了。这时候只能用【多虚拟机】来隔离。

“多虚拟机”的方案，说起来挺简单——就是安装多个操作系统虚拟机，把浏览器安装到虚拟机中。利用虚拟机来进行隔离，每个虚拟机就如同一台单独的电脑，这种隔离性，比前面三个方案更加彻底。如果你不熟悉操作系统虚拟机，请先看俺写的《扫盲操作系统虚拟机》系列博文。

★DNT（Do Not Track）

说完了浏览器的隔离，再来说几个杂项。首先说说 DNT 这玩意儿。

◇DNT 是啥?

DNT 是洋文 Do Not Track 的缩写，中文译作“请勿追踪”。它最早是由 Mozilla 的一个工程师在 2009 发明的，如今已经成为 W3C 的标准。截至2012年底，所有主流的浏览器都已支持 DNT 标准。

◇DNT 的原理

这玩意儿说白了没啥技术含量，其原理大致如下：

如果你在浏览器中启用了 DNT，那么浏览器每次访问网站的时候，会在 HTTP 请求的 header 部分加入一个 DNT 的标识。网站的服务器接收到这个 HTTP 请求，看到此标识，就知道该用户不希望被追踪。

如果这个网站遵循 W3C 的规范——当它看到这个 DNT 标识，就【不】应该使用 cookie 或诸如此类的手段来追踪用户的行为；反之，如果这个网站【不】遵循 DNT 规范，它就会直接无视 HTTP 请求中的 DNT 标识。

◇DNT 有用吗?

通过上述介绍你应该可以看出：DNT 技术并不是一项很保险的防范措施。DNT 要起作用需要靠网站方面【自觉配合】。如果你访问的网站比较流氓，不愿意配合这个规范，那你的 DNT 设置就形同虚设。

所以，DNT 是一个有点鸡肋的功能，不能全指望它，但“启用”总比“不启用”要好（至少没啥坏处）。虽然遵循该规范的“老实”网站很少，但如果你访问的网站正好是“老实”的，启用 DNT 就能禁止该网站对你的追踪。

★User Agent

◇“User Agent”是啥?

浏览器方面还有一个涉及到隐私的因素，而且不太为人所知，那就是“User Agent”。

浏览器的 UserAgent 是用来标识客户端的信息（包括 浏览器的类型和版本、操作系统类型和版本、等等）。浏览器向网站发起 HTTP 请求时，会在 HTTP header 中加入 User Agent 信息。

◇“User Agent”有啥用?

因为 User Agent 标识了浏览器客户端的信息，网站拿到这些信息之后，就可以针对客户端的不同，发送针对性的网页。比方说，如果客户端是移动设备（屏幕较小），就发送针对小尺寸屏幕的网页。

从这里的介绍可以看出，User Agent 本身是有用的。

◇“User Agent”有啥隐私问题?

但是，有些浏览器的 User Agent 写得太详细了。这就导致：很多额外的客户端信息也通过 User Agent 提交到 Web 服务端。网站拿到这么详细的信息，就可以知道你操作系统和浏览器的很多细节。另外，User Agent 越详细，“独特性”就越明显。那么网站就可以利用 User Agent，**【大致猜测】**某些页面访问是否来自同一个人。

放几个比较详细的 User Agent 给大伙儿瞻仰一下：

(某个 IE)

> Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; Zune 4.0; Tablet PC 2.0; InfoPath.3; .NET4.0C; .NET4.0E)

(某个 Opera)

> Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.13) Gecko/20101213 Opera/9.80 (Windows NT 6.1; U; zh-tw) Presto/2.7.62 Version/11.01

(某个腾讯浏览器)

> Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; QQPinyin 686; QQDownload 661; GTB6.6; TencentTraveler 4.0; SLCC1; .NET CLR 2.0.50727; Media Center PC 5.0; .NET CLR 3.0.04506)

想知道自己浏览器的 User Agent 长啥样？猛击下面这个网址就可以看到了。

http://www.useragentstring.com/

◇要不要修改“User Agent”?

对于大部分网友而言，如果对隐私方面的要求不高，没必要修改 User Agent。目前为止，User Agent 的隐私问题不如 cookie 严重。所以隐私要求不高的网友，先学会清除 cookie 的招数。

但如果你对隐私的要求比较高，可以考虑修改自己浏览的默认 User Agent，伪造一个假的。

◇如何修改“User Agent”?

要想避免 User Agent 泄漏隐私，简单的办法就是修改浏览器的“默认 User Agent”
俺简单说一下三大浏览器如何修改默认的 User Agent：

Firefox
通过定制 Firefox，创建一个新的配置项，其“名称”是 `general.useragent.override`，其“值”就是“新的 User Agent”。
定制 Firefox 的方法参见博文：《扫盲 Firefox 定制——从“user.js”到“omni.ja”》。

Chrome
在 Chrome 的启动参数中加上 `--user-agent="XXX"`
这个 XXX 就是新的 User Agent。（可以在快捷方式中追加命令行的启动参数）

IE
用 regedit 打开注册表，编辑键值 `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\User Agent`
输入新的 User Agent

◇“User Agent”改成啥样比较好?

用上述方式，可以伪造你浏览器的 User Agent。那么，伪装的 User Agent 该如何写捏?
前面俺说了：“User Agent”越【独特】就越容易被追踪。所以，当你要伪造自己的“User Agent”，当然是改成比较【大众化】的。下面这3个链接包含了三大浏览器【常见的】User Agent，供大伙儿参考。
Firefox 的 User Agent
Chrome 的 User Agent
IE 的 User Agent
你可以把自己浏览器的 User Agent 伪装成另外的两种之一。

★结尾

用了三篇博文，大致把浏览器【常见的】隐私问题介绍了一遍。大伙儿如果还有其它补充，或者有其它疑问，欢迎到本文留言。俺会尽量给予解答。
本系列的下一篇，俺来扫盲一下“浏览器指纹”。

[如何保护隐私5]：扫盲“浏览器指纹”

文章目录

去年4季度，俺写了4篇《TrueCrypt 使用经验》，又写了3篇 Linux 方面的扫盲，导致本系列又中断了几个月。经热心读者提醒，现继续补上。

本系列的前面三篇，咱们聊了浏览器的基本防范。对于“隐私要求不高并且技术水平也不高”的同学，那三篇基本上够了。后面俺要继续聊浏览器方面的问题，面向的是那些“对隐私要求较高，同时也具有一定折腾能力”的同学。今天这篇谈谈浏览器的“指纹”是如何暴露你的隐私，顺便分享一些防范技巧。

★啥是“指纹”?

看过警匪片或者破案小说的同学，应该都知道“指纹”在刑侦中的作用——警方虽然没有直接看到犯罪现场的作案人员，但可以根据现场留下的指纹来猜测/判断作案人员。

如果你时常接触信息安全领域的一些资料，也会听到“指纹”这个形象的说法（比如：操作系统指纹、网络协议栈指纹、等等）。IT 领域提到的“指纹”一词，其原理跟“刑侦”是类似的——“当你需要研究某个对象的类型/类别，但这个对象你又无法直接接触到。这时候你可以利用若干技术来获取该对象的某些特征，然后根据这些特征来猜测/判断该对象的类型/类别。”

★啥是“指纹”的“信息量”?

在 IT 领域有各种各样的特征可以用来充当“指纹”。这时候就需要判断，用哪个特征做指纹，效果更好。为了讨论这个问题，就得扫盲一下“指纹的信息量”。

为了帮助大伙儿理解，先举一个例子：

假设你要在学校中定位某个人，如果你光知道此人的性别，你是比较难定位的（只能排除 1/2 的人）；反之如果你不知道性别，但是知道此人的生日，就比较容易定位（可以排除掉大约 364/365 的人，只剩大约 1/365 的人）。为啥捏？因为“生日”比“性别”更加独特，所以“生日”比“性别”能够提供更多的信息量。

从这个例子可以看出：某个特征越独特，则该特征的信息量越大；反之亦然。信息量越大的特征，就可以把对象定位到越小的范围。

★“指纹”的“信息量”如何度量——关于指纹的比特数?

（本节涉及到中学数学，数学很差的或者对数学有恐惧感的读者，请直接无视）

在 IT 领域中，可以用【比特数】来衡量某个指纹所包含的信息量。为了通俗起见，先以前面提到的“性别”来说事儿。性别只有两种可能性——“男”或者“女”，并且男女的比例是大致平均的。（那些喜欢抬杠的同学，别跟俺扯啥双性人，俺没空搭理）所以，当你知道了某人的性别，就可以把范围缩小到原先的 1/2。用 IT 的术语来讲，就是：“性别”这个特征只包含“一个比特”的信息量。

以此类推：

当我们说：某特征包含3比特信息量，意思就是：该特征会有8种大致平均的可能性（8等于2的3次方）。一旦知道该特征，可以把目标定位到八分之一。

当我们说：某特征包含7比特信息量，意思就是：该特征会有128种大致平均的可能性（128=2^7）。一旦知道该特征，可以定位到 1/128

再来说“生日”。（不考虑闰年的情况下）生日有365种可能性（并且也是平均分布滴），所以生日包含的比特数大约是 8.51。为啥是 8.51 捏，因为“2 的 8.51 次方”约等于 365。因此，知道了某人的生日就可以把范围缩小到 1/365

通过上述举例，大伙儿对于指纹的信息量，应该有一些粗浅的认识了吧?

★多个指纹的综合定位

如果能同时获取【**互不相关**】的若干个指纹，就可以大大增加定位的精确性。

比如要在某个公司里面定位某人，如果你知道此人的“生日”和“生肖”，那么就可以达到 1/4380（1/4380 = 1/12 * 1/365） 的定位精度。因为综合定位之后，比例之间是【乘法】的关系，所以范围就被急剧缩小了。

为啥俺特别强调“互不相关”捏？假如你同时知道的信息是“生日”和“星座”，那么定位的精度依然是1/365——因为生日的信息已经包含了星座的信息。所以，只有那些相互独立的特征（所谓的相互独立，数学称为“正交”），在综合定位的时候才可以用【乘法】。

★啥是“浏览器的指纹”？

前面说的是预备知识，现在开始进入正题。

当你使用浏览器访问某个网站的时候，浏览器【**必定会暴露**】某些信息给这个网站。为啥俺强调“**必定**”捏？因为这些信息中，有些是跟 HTTP 协议相关的（本章节说的 HTTP 协议是广义的，也包括 HTTPS）。只要你基于 HTTP 协议访问网站，浏览器就【必定】会传输这些信息给网站的服务器。

再罗嗦一下：HTTP 协议是 Web 的基石。只要你通过浏览器访问 Web，必定是基于 HTTP 协议的。因此，Web 网站的服务器必定可以获取到跟你的浏览器相关的某些信息（具体是哪些信息，俺下面会聊）。

★“浏览器指纹”如何暴露隐私？

“浏览器指纹”的机制跟 cookie 有点相似。关于 cookie 的作用，建议那些健忘的同学先去“前面的博文”复习一下。

对于“浏览器指纹”导致的隐私问题，俺举2个例子来说明其危害。

◇对于无需登录的网站

如果你的浏览器允许记录 cookie，当你第一次访问某网站的时候，网站会在你的浏览器端记录一个 cookie，cookie 中包含某个“唯一性的标识信息”。下次你再去访问该网站，网站服务器先从你的浏览器中读取 cookie 信息，然后就可以根据 cookie 中的“唯一标识”判断出，你之前曾经访问过该网站，并且知道你上次访问该网站时，干了些啥。对付这种 cookie 很简单，你只需要在前后两次访问之间，清空浏览器的 cookie，网站就没法用 cookie 的招数来判断你的身份。

但是“清空 cookie”这招对“浏览器指纹”是无效滴！比如说你的浏览器具有非常独特的指纹，那么当你第一次访问某网站的时候，网站会在服务器端记录下你的浏览器指纹，并且会记录你在该网站的行为；下次你再去访问的时候，网站服务器再次读取浏览器指纹，然后跟之前存储的指纹进行比对，就知道你是否曾经来过，并且知道你上次访问期间干了些啥。

◇对于需要登录的网站

假如网站没有采用“指纹追踪”的技术，那么你可以在该网站上注册若干个帐号（马甲）。当你需要切换身份的时候，只需要先注销用户，清空浏览器的 cookie，然后用另一个帐号登录。网站是看不出来的。**一旦网站采用“指纹追踪”的技术**，即使你用上述方式伪造马甲，但因为你用的是同一个浏览器，浏览器指纹相同。网站的服务器软件可以猜测出，这两个帐号来自同一个人。

★“浏览器指纹”比“cookie”更隐蔽，更危险

刚才对比了“浏览器指纹”和“cookie”两种身份追踪技术。两者的原理类似——都是利用某些特殊的信息来定位你的身份。两者的本质差异在于：
\1. cookie 需要把信息保存在浏览器端，所以会被用户发现，也会被用户清除。
\2. 而“浏览器指纹”无需在客户端保存任何信息，不会被用户发觉，用户也无法清除（换句话说：你甚至无法判断你访问的网站到底有没有收集浏览器指纹）。

★“浏览器指纹”包含哪些信息?

浏览器暴露给网站的信息有很多种，常见的有如下几种：

◇User Agent

关于 User Agent 是啥，俺已经在本系列前一篇博文中扫盲过了（请看“这里”），健忘的同学先去复习一下，再继续往下看。

◇屏幕分辨率

这个比较通俗易懂。俺稍微补充一下：这一项不仅包括屏幕的尺寸，还包括颜色深度（比如你的屏幕是16位色、24位色、还是32位色）。

◇时区

这个也比较通俗。对于大部分天朝的网友，你的时区应该都是“东8区”

◇浏览器的插件信息

也就是你的浏览器装了哪些插件。
再罗嗦一次：浏览器的“插件”和“扩展”是两码事儿，别搞混了。本系列前面的博文扫盲了两者的差异，链接在“这里”。

◇字体信息

和浏览器相关的一些字体信息。
如果你的浏览器安装了 Flash 或 Java 插件，有可能会暴露某些字体信息。所以俺在本系列的第2篇就警告了浏览器插件的风险。

◇Canvas 绘图的指纹

Canvas 是 HTML5 新增的一个功能。该功能可以让 JavaScript 脚本在页面的 canvas 元素中绘图。由于不同的浏览器类型，不同的浏览器版本，不同的操作系统平台，都会导致“Canvas 绘图”在一些细节方面的差异。因此，该功能也会暴露浏览器信息（成为“浏览器指纹”的一部分）。
要想测试你的浏览器是否暴露“Canvas 绘图指纹”，可以查看“这个链接”。
如果你用的是 Firefox 浏览器，可以安装“这个扩展”，来禁用 canvas 绘图功能。该扩展可以针对不同的域名配置“黑名单和白名单”。

◇HTTP ACCEPT

这是 HTTP 协议头中的一个字段。考虑到列位看官大都不是搞 IT 技术的，俺就不深入解释这项。

◇其它

以上就是常见的浏览器指纹。当然啦，还有其它一些信息也可以成为“浏览器指纹”，考虑到篇幅，俺就不一一列举并解释了。有兴趣的同学，请自行阅读 Mozilla 官网的文档（在“这里”。提醒一下：是洋文）

★如何看自己浏览器的指纹?

关于浏览器指纹导致的隐私问题，可能是由“电子前哨基金会”（简称 EFF）率先在2010年曝光的。后来 EFF 提供了一个页面，帮助网友看自己浏览器的指纹（请猛击“这个链接”）。

打开此页面之后，当中有一个大大的，红色的“TEST ME”按钮。点一下此按钮，稍等几秒钟，会显示出一个表格，里面包含你当前的浏览器的指纹信息。

在这个表格中会列出每一项指纹的“信息量”以及该指纹的“占比”。关于“信息量”的含义，本文前面已经扫盲过，此处不再罗嗦。你只需记住，某项的信息量越大，就说明该项越独特。而越独特的指纹，对隐私的威胁也就越大。

考虑到篇幅有点长，今天先聊到这里。俺争取明后天发下一篇，聊聊如何防范“浏览器指纹”导致的隐私风险。

[如何保护隐私6]：如何防范“浏览器指纹”?

文章目录

前一篇介绍了“浏览器指纹的基本概念”今天这篇分享一些防范的技巧。

★防范“指纹”的一般性原则?

不管是哪一种特征，要想成为“指纹”至少要具备两个条件：“唯一性”和“稳定性”。比如人类手指的纹路就同时具有“唯一性”和“稳定性”——任意两个人的纹路都不同，而且每个人的纹路终生不变。所以，要对付“指纹识别”，咱们就必须反其道而行——破坏“唯一性”和“稳定性”。对浏览器而言，做到这两点并不难。且听俺细细道来。

★“浏览器指纹”的构成及信息量

前一篇博文已经给大伙儿介绍了 EFF 的浏览器指纹测试工具（链接在“这里”）。通过这个工具可以明显看出，User Agent 的信息量最大，至少占据一半以上的信息量。换句话说，其它所有特征的信息量加起来都没有 User Agent 大。而且除了 User Agent，其它特征的信息量都比较小。这说明啥捏?

请大伙儿换位思考一下：如果某个网站想要利用浏览器指纹进行用户身份定位，User Agent 是必不可少的一项。缺少这一项，定位的精度会大打折扣。所以，User Agent 是浏览器指纹的关键性信息。

俺已经在博客中多次唠叨了“二八原理”，浏览器指纹中的 User Agent 就是这关键性的“20%”。有鉴于此，本文的主要篇幅谈【User Agent】；聊完它之后，再聊另外几种指纹机制及防范。

★如何消除【User Agent】的指纹

◇利用浏览器内置的防范措施

先来聊最简单的招数——使用浏览器内置防御措施。

要想用这招，首先要判断你使用的浏览器，是否内置了“伪装 User Agent”的功能。

由于浏览器有很多种，以下介绍用 Firefox 来举例。

Firefox 从版本 59 开始，引入 RFP 功能（这是洋文“Resistance Fingerprinting”的缩写）。如果启用了该功能，Firefox 默认就会伪装 User Agent 的信息——把真实的 User Agent 伪装成某个常见的 User Agent。

本方案的优点

“易用性”很好——你只需开启这个功能，Firefox 自动就帮你伪装好了。

本方案的缺点

主要缺点是“不够彻底”——由于 Firefox 使用某个固定且常见的“User Agent 值”进行伪装。因此，目标网站依然【有可能】判断出你在使用 Firefox。仅仅知道你在使用 Firefox，这个信息量【很小】。因此，这个缺点不严重。

◇“多浏览器”方案

这个方案最简单，也最容易想到。一看这个小标题，估计大部分读者都猜到俺想喷啥口水。

如果你同时具有两个不同的浏览器（比如：一个 Firefox 一个 Chrome），那么这两个浏览器必然具有不同的 User Agent。如果某个网站收集了浏览器指纹，而你又想在这个网站注册两个不同的马甲，那么你就可以用“多浏览器方案”——分别用不同的浏览器注册不同的马甲。

本方案的优点

操作很简单，会装浏览器的同学都玩。

本方案的缺点

浏览器的种类毕竟有限（知名且靠谱的浏览器，一只手都能数过来）。万一你想注册十多个马甲，用这个方案就显得傻逼了。

◇“多实例”方案

为了解决“多浏览器方案”的局限性，自然会想到“多实例”这个方案。此招数俺曾经在《如何防止黑客入侵》系列博文中介绍过。

在主流的三大浏览器中，Firefox 和 Chrome 支持“多实例”，IE 不支持。所以那些喜欢 IE 的同学就没法用这招了。

关于 Firefox 和 Chrome 如何配置多实例，请看俺之前的博文（这里）。对于用 Chrome 的同学，俺

再次罗嗦一下：Chrome 同时提供“多实例”（洋文叫“Multiple Profiles”）和“多用户”（洋文叫“Multiple Accounts”）两种功能，这两者是完全不同滴。它的“多用户”依然在同一个“实例”中。
配置完“多实例”之后，一定要记得修改每一个实例的 User Agent，并确保【两两不同】。至于如何修改 User Agent 请参见“这篇博文”——里面提供了三大主流浏览器的修改方法。

本方案的优点
浏览器的实例可以配置任意多个（只要你有耐心，硬盘够大，配几百几千都可以）。

本方案的缺点
某些浏览器（比如 IE）不支持多实例。

◇“多虚拟机”方案

要对付 User Agent 的指纹，前面两招基本够用了。但某些同学可能有特殊需求，或者安全要求比较高，所以俺顺便介绍第三种方法。
第三种方法就是利用虚拟机软件安装不同的虚拟系统，然后在每个虚拟系统中安装浏览器。没用过虚拟化软件的同学，先看俺之前的扫盲教程（在“这里”）。再次罗嗦：如果你在不同的虚拟机中安装相同的浏览器，要记得修改【每个】虚拟机中浏览器的 User Agent。

本方案的优点
优点1：前面说了，某些浏览器不支持多实例。万一你偏偏喜欢这种浏览器，就可以考虑用“多虚拟机”的方案。
优点2：因为屏幕分辨率、系统时区也都是指纹特征。所以在虚拟系统中，你还可以调整屏幕分辨率和时区（使之不同于你真实系统的分辨率和时区）。

本方案的缺点
缺点1：你需要额外安装虚拟化软件，然后再安装虚拟系统。过程稍嫌繁琐。对技术菜鸟也有难度。
缺点2：对系统的硬件有一定的要求（如果你的电脑硬件太寒酸，就甭考虑这招啦）。

◇“动态 User Agent”方案

（现在来聊最高级，也最难的方案）
善于思考的同学会发现：前面三个招数，其本质是相通滴。说白了都是利用技术手段“隔离”出不同的浏览器环境，然后单独修改每个环境的指纹，以此来伪造出多个身份。但是对于每一个具体的环境，其指纹依然是固定的。换句话说，前面那三个招数都是针对指纹的【唯一性】。下面俺要介绍的招数可以用来破坏指纹的【稳定性】。
前面说了，浏览器指纹的信息量，至少有一半以上是来自于 User Agent。所以要破坏浏览器指纹的稳定性，只要让浏览器的 User Agent 动态变化即可。下面分别说明技术思路（以下的招数适合于有一定折腾能力的同学，需要用到一点点编写脚本的伎俩）。

如何获取常见的 User Agent
要构造随机的 User Agent，其实也不难。到“这个网站”可以看到各种各样浏览器的 User Agent。你可以收集一大堆预存着，然后每次从中随机挑选一条作为你的伪装。为了做到每次随机挑选并设置，你可以写一个脚本来干这事儿，然后顺便让这个脚本来帮你启动浏览器。
再唠叨一下：挑选 User Agent 是有讲究滴，要尽量选择那些比较常见的 User Agent——越常见的 User Agent 所包含的信息量越小。

对 Firefox 的定制
三大浏览器中，最有利于隐私保护的是 Firefox（具体的原因分析请看本系列前面的博文），所以先说它的技术实现。

通过修改 `user.js` 文件，可以手工指定 Firefox 使用的 User Agent。具体做法是：往 `user.js` 添加一个配置项，其“名称”是 `general.useragent.override`，其“值”就是“伪装的 User Agent”。

如何修改 `user.js`，请参见博文：《扫盲 Firefox 定制——从“user.js”到“omni.ja”》

上述做法只能达到【静态】效果——你伪装了 User Agent 之后，它就一直保持你设定的值。

要想做到【动态】的效果，就需要用到【脚本】来自动修改 `user.js` 文件里面关于 `general.useragent.override` 的那行代码，每次修改都使用某个【随机选定】的 User Agent。修改完 `user.js` 之后再把 Firefox 启动起来。

对于 Windows 下的 Firefox，可以用 VBScript 或 JScript 或 PowerShell 这三种系统内置的脚本；对于 Linux 或苹果系统，可以尝试各种 shell 脚本。

某些爱思考的同学可能会问，为啥不直接在 `user.js` 文件里面用 javascript 代码进行 User Agent 的随机生成。

俺也曾经企图这么干，可惜不行！因为 `user.js` 中对函数 `user_pref` 的调用，两个参数都必须是【常量】；而且， `user.js` 中除了调用该函数，不允许再出现其它的代码行。

对 Chrome 的定制

对于 Chrome，可以在命令行参数指定其 User Agent，具体请参见“这篇博文”。

所以，你可以自己写一个脚本，专门用来启动 Chrome。每次启动都传递一个随机的 User Agent 作为命令行参数。

对于 Windows 下的 Chrome，可以用 VBScript 或 JScript 或 PowerShell 这三种系统内置的脚本；对于 Linux 或苹果系统，可以尝试各种 shell 脚本。

对 IE 的定制

对于 IE 的 User Agent，需要修改注册表的键值（`HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\User Agent`）。所以捏，可以通过事先写好的脚本（VBScript 或 JScript 或 PowerShell）往相应的注册表键值中写入随机的User Agent，然后再由这个脚本启动 IE。

◇补充说明

（本文发出之后，看到某些读者留言，特地补充这个小节）

对于“动态 UserAgent 方案”，很多读者在问：为啥不直接给出代码？

俺提醒一下：**【授人以鱼不如授人以渔】**是本博客长期奉行的原则。所以在技术方面，俺更愿意分享一些思路，尽量避免直接给出现成的东西。自己动手实践，有助于能力的提升而且印象更深刻。

如果你是个程序猿/程序媛，写这样一个脚本应该是易如反掌滴；如果你不是搞技术的，顺便学一下简易的脚本编程（其实很容易滴）。不会编程的同学，俺建议从 Python 开始入手，功能强且门槛低，具体请参见《为啥俺推荐 Python》系列博文。

★如何消除【屏幕分辨率】的指纹

◇“屏幕分辨率”的信息量

对于【不】使用虚拟机的普通网友，其屏幕分辨率也就是【常见】的那几种（大概在20种以内）——这种情况下，“屏幕分辨率”暴露出的信息量【很低】，不要紧。

但如果你使用“操作系统虚拟机”，就需要留意“屏幕分辨率的问题”。

当你在 Guest OS 中上网，并且你【没有】使用全屏模式，那么 Guest OS 的分辨率可能会是一个很奇怪的分辨率（因为独特性很高，会包含【很多】的“指纹信息量”）。所以，俺建议那些用虚拟系统上网的同学，采用如下几个措施：

◇如何防范

措施1：每个 VM 都采用“全屏模式”

这个方案针对的是——【普通】的隐私需求。

VM 全都采用【全屏】，虽然每个 VM 的分辨率相同。但至少这个分辨率是某个【常见】的分辨率，因此【信息量很低】。

措施2：对每个 VM 都采用某个常见的分辨率（【非】全屏），且每个 VM 的分辨率【各不相同】

这个方案针对的是——【特别高】的隐私需求。

由于每个 VM 的分辨率各不相同，假设网站收集了分辨率作为指纹，不同 VM 的上网身份，会被识别为【不同身份】。

★如何消除【Canvas】的指纹

◇“Canvas”是啥玩意儿?

在本文发布约半年后，有读者在博客留言中询问了“基于 HTML5 的 Canvas 语法进行指纹追踪”。所以俺单独补充了这个章节。

所谓的“Canvas 指纹”，依赖的是 HTML5 新增的 Canvas 语法。利用这个 Canvas 语法可以实现一些绘图的功能。由于不同类型的浏览器使用了不同的绘图引擎；并且同一种浏览器在不同操作系统平台上，绘图引擎的特性也会有细微的差别。因此，“Canvas 的功能特性”会成为某种指纹信息。

◇如何防范

Canvas 的绘图功能要依赖于 JS 脚本。因此，只需“禁用 JS 脚本”就可以让“Canvas 指纹”失效。

俺的建议是：使用一些安全扩展（比如 NoScript），对你不信任的网站禁用 JS 脚本。另外，电子前哨基金会（EFF）提供的“Privacy Badger 扩展”，也可以屏蔽“Canvas 指纹”。

要测试自己的浏览器是否存在“Canvas 指纹”，请猛击“这个链接”。

对于 Firefox 浏览器，从版本 59 开始，已经引入 RFP（洋文“Resistance Fingerprinting”的缩写），默认就会限制 Canvas 指纹。

退一步讲，就算你无法屏蔽“Canvas 指纹”，也不用怕。在本文开头提到了“防范指纹的一般性原则”，其中之一是【破坏唯一性】。前面章节介绍了几个招数，用来破坏“User Agent 的唯一性”。这些招数也可以用来破坏“Canvas 的唯一性”。

提醒一下：单纯用“多实例”的招数无法破坏“Canvas 指纹”的唯一性。因为在“多实例”的情况下，每个实例共享【同一个】浏览器引擎。所以你必须采用“多浏览器”或者“多虚拟机”的方式。

★补充说明： EFF 的浏览器指纹测试

前一篇博文介绍了 EFF 的浏览器指纹测试工具，估计很多同学都去测试了。其实捏，不必太在意具体每一项的“比特数”。大伙儿只需要关注其“定性”而不必太在意其“定量”。因为 EFF 网站目前收集的样本还不够多（只有几百万），所以其分析出的信息量（相比全球的统计数据）会有所偏差。

另外，很多人测试下来的总信息量是 21.85 bits，这是因为 EFF 的总样本目前只有 370万 左右（370万 约等于“2的21.85次方”）。所以比特数到 21.85 就封顶了。

[如何保护隐私7]：其它桌面软件的隐私问题

文章目录

前几天又有热心读者来催促，让俺尽快把以前挖的“坑”填平。今天来继续补充《如何保护隐私》系列。在本系列之前的部分，俺已经花了5篇帖子，大谈特谈【浏览器】相关的隐私问题。接下来要聊聊其它桌面软件的隐私问题。

为了避免歧义，先声明一下：本文所说的“桌面软件”泛指各种“PC 端软件”。至于移动设备（手机、平板）的隐私问题，会在本系列的后续博文中另外介绍。

★安全类

在安全界混过的人或许明白一个道理——不靠谱的安全软件反而更危险。所以俺首先来介绍一下安全软件导致的隐私问题。

◇利用“扫描硬盘”收集隐私

安全软件如果要查杀病毒/木马，通常需要先扫描硬盘文件。如果某款安全软件不靠谱，那么它在扫描硬盘的过程中，就会悄悄滴收集你硬盘上有价值的个人资料。而且这类安全软件通常都需要定期升级“病毒特征库”或“木马特征库”。要升级特征库必然要联网（连接到厂商的服务器）。如此一来，这类不靠谱的安全软件就可以利用升级的时机，把收集到的用户信息发送到厂商服务器上。整个过程可谓是神不知鬼不觉啊。

举例1：

还记得前几年的“3Q 大战”吗？当时的导火索之一就是 QQ 存在隐私问题——据说 QQ 软件会偷偷地扫描硬盘。后来腾讯官方也承认了这点，但是又辩解说这是“QQ 电脑管家”在进行安全检查。不过懂行的网友分析了“QQ 电脑管家”的行为，发现它扫描硬盘的过程更像是在收集用户信息，而不像是查杀病毒/木马。具体参见“这篇”和“这篇”。

举例2：

在《360黑匣子之谜——奇虎360“癌”性基因大揭秘 @ 每日财经新闻》一文中，提及了奇虎是如何利用“360”这款所谓的“安全工具”来收集用户隐私。

◇利用“网络监控”收集隐私

除了查杀病毒/木马需要扫描硬盘，某些安全软件还需要进行流量监控，以此来防范网页挂马。如果某个安全软件不靠谱，那么它可以利用“网络流量监控”的机会，收集你的上网行为（比如你常看的网站）。收集好信息之后，同样可以利用升级特征库的机会，把用户隐私发送到厂商的服务器上。

◇其它收集隐私的手段

除了上述这两招，安全厂商还有其它一些手段来收集用户隐私。

比如奇虎大力推广的“360安全浏览器”。这款浏览器会把“用户访问的网址、网址对应的 cookie、用户在地址栏输入的内容”等信息上传到奇虎的服务器上。具体的报道请看《中科院报告：360产品存在三大隐私安全问题 @ 新浪科技》。说到这里，俺顺便感叹一下：奇虎/360 的老板周鸿祎，当年就是做流氓软件起家的（老网友应该还记得臭名昭著的“3721插件”）。如今已经过了10多年了，这厮真是“狗改不了吃屎”啊。

说到浏览器，再次罗嗦一下：

其一，千万别用【国产的】浏览器

其二，全球三大桌面浏览器，在隐私保护方面的排序是：Firefox 优于 Chrome/Chromium 优于 IE（Chrome/Chromium 的粉丝如果不服，可以看本系列“第2篇博文”的分析）

◇国产安全软件有靠谱的吗?

对这个问题，俺倾向于【否定】的回答。虽然俺无法拿出确凿证据来证明“每一款国产安全软件都有问题”，但是俺可以给出如下一些分析供参考。

刚才提到过，在安全软件中植入后门具有某种天生的优势——其一，杀毒软件或木马查杀软件天生就需要扫描硬盘；其二，杀毒软件或木马查杀软件天生就需要定期升级特征库（在线升级就需要联网）。于是乎，假如某款国产的杀毒软件或木马查杀软件达到【足够大】的装机量，通常会被朝廷的【有关部门】盯上。对于国产软件公司而言，如果“有关部门”找你谈话，要求你在软件中植入后门，你有胆量拒绝吗？显然没有。

某些读者可能会反问说——那美国的杀毒软件（比如：赛门铁克、迈克菲）也可能有 NSA（美国国安局）的后门啊。这类猜测是有道理滴。但是俺要提醒一下：本博客的大部分读者都是大陆网民，**美国国安局对你是没有“司法管辖权”滴**。所以就算美国的安全软件有 NSA 的后门，其危险性【远小于】咱们党国的后门。再退一步讲，假设你对隐私性的要求非常高，容不得一丁点后门，那俺的建议是：干脆放弃 Windows，改用 Linux（如此一来，也就无需杀毒软件了）。没用过 Linux 的同学可以看俺之前写的扫盲教程《新手如何搞定 Linux 操作系统？》。

★文字输入类

为啥俺把“输入法”排在第二项捏？首先是输入法非常普及（几乎每个天朝网民都会装）；其次是输入法可以暴露你本人的很多信息，从而导致比较严重的隐私问题。

◇敏感词监控

跟安全软件类似，那些装机量巨大的输入法，对朝廷是很有吸引力滴。如果能通过这些输入法监控网民的文字输入，就可以发现潜在的“不和谐分子”。这种风险可不是俺瞎编出来吓唬大家滴。不信请看如下的举例。

举例：

《国产软件的隐私问题 @ Solidot》，其中提到了：拼音加加2004v3.02，新华五笔，这两个软件已经发现有后门和**敏感词汇汇报功能**的存在，会在不知不觉中泄漏隐私。请注意俺标了粗体的“敏感词汇汇报功能”。这类功能很显然是为朝廷量身定做滴。

虽然这篇报道只提到了“拼音加加、新华五笔”这两款，但这【不】表示其它输入法就【没有】问题——可能只是尚未发现而已。

◇个性化词库同步

“词库同步功能”，大伙儿应该不陌生吧？目前主流的输入法都会根据输入频率存储用户的“个性化词库”。“个性化词库”是为了提高输入效率的——比如当你的输入出现二义性，你经常选中的那个候选词会出现在靠前的位置。

所谓的“词库同步功能”，就是把你本机的个性化词库同步到云端。这个功能对那些拥有多台电脑的网友而言，是有帮助滴。但是这也带来了隐私的风险。因为个性化词库存储在云端，那么该输入法厂商就可以对你的“个性化词库”进行分析，从而了解你本人的种种信息。假如说你经常用输入法进行网络聊天、撰写邮件、写文章、等等，那么你的个性化词库包含的信息量就非常非常大。

而且俺有充分的理由相信，朝廷六扇门的人肯定会去分析存储在云端的“用户个性化词库”。

◇“服务器泄露”导致的隐私问题

比如很有名的“搜狗输入法”就在去年（2013）曝出高危漏洞，具体参见《搜狗输入法收集用户隐私信息，未屏蔽爬虫 @ Solidot》一文。据说该漏洞曝光了很长时间（超过1个月）才修复。在这段时间内，别有用心的骇客可以利用这个漏洞，收集到搜狗输入法用户的大量聊天内容（尤其利用它输入的图片网址）。

★即时通信类

考虑到 QQ 是天朝即时通信（以下简称“IM”）的老大，下面以 QQ 为主介绍一下 IM 导致的隐私问题。关于 QQ 扫描硬盘的问题，前面已经提到了，所以本章节聊聊另外的几个问题。

◇敏感词监控

刚才介绍“输入法”的隐私问题，其中之一是“敏感词监控”。对于 IM 而言，同样也存在“敏感词监控”的问题，道理跟“输入法”一样。像 QQ 这种用户量如此巨大的软件，朝廷肯定不会放过滴。所以 QQ 系统中早就安插了朝廷方面的监控工具（据说是部署在腾讯的服务器上）。如果你在 QQ 聊天过程中频繁涉及到一些敏感的政治词汇，就会引起六扇门的注意。相关报道如下：

《QQ 是如何监视你的聊天记录的 @ 网易科技》

不光是 QQ，其它一些 IM 工具也存在敏感词监控的问题。比如臭名昭著的“TOM 版 Skype”。该版本完全是针对天朝的国情进行定制，内置了庞大的“敏感词词库”。一旦用户在聊天时输入了词库中的某个敏感词，相关的聊天内容和聊天帐号会被记录在 TOM 的服务器上。外媒的报道如下：

《Skype 中国合作伙伴“监控用户” @ BBC》

说到“TOM 版 Skype”，顺便提一下：TOM 跟 Skype 的合作已经在2013年11月到期。如今 Skype 换了一个新的合作伙伴，叫做“光明方正”（这是《光明日报》跟北大方正的合资公司）。以前在天朝访问 Skype 官网会自动跳转到 `skype.tom.com` 网站，如今则自动跳转到 `skype.gmw.cn` 网站。考虑到《光明日报》是朝廷的喉舌，估计现在这个“光明版 Skype”也不会是啥好鸟。

◇“服务器泄露”导致的隐私问题

除了“敏感词监控”，还有其它一些问题也会导致隐私泄露。比如腾讯的系统如果出现安全方面的漏洞并且被骇客利用，那么腾讯用户的很多隐私都会暴露无遗。

举例1：

如果你在 QQ 的聊天内容中发送过 URL 网址，这个网址会被“腾讯搜搜”抓取。如果这是一个公开的网址，倒也无所谓。但万一这是个【私密的网址】，那就麻烦啦。比如有些网民会在 QQ 聊天中发送“微信支付（财付通）”的订单网址，那么这些私密订单网址就会被“腾讯搜搜”收录下来。然后骇客就可以

去“腾讯搜搜”里面收集大量网友的订单信息。相关报道如下：
《腾讯 QQ 微信均现安全隐患 漏洞十分严重 @ IT商业网》

举例2：

去年（2013）11月，腾讯的 QQ 群数据被泄露，数据量高达90GB，涉及了7000多万个 QQ 群。根据这批数据，你可以用 QQ 号查询到姓名、年龄、社交关系网等大量个人隐私。如果你是 QQ 的长期用户，你的个人喜好（包括那些猥琐的喜好）就一览无遗啦 :)

相关报道如下：
《腾讯 QQ 群数据库泄露事件追踪 @ 新华网》
《腾讯群关系数据泄漏（可根据 QQ 号获得该人姓名经历等详细信息） @ 乌云》

★下载类

国内最知名的下载工具大概就是“迅雷”了。所以重点说说它的劣迹。

◇暴露你的私有文件

正常的 P2P 下载软件，只会上传“用户共享出来的目录里的文件”。但是迅雷可没有这么规矩。据说它会擅作主张，查找你电脑硬盘中的文件。如果某个文件正好是当前的热门下载，迅雷就会把该文件上传给其他正在下载的用户（以此来“提速”）。这种做法从某种程度上暴露了你的隐私。相关报道如下：
《小心，你的文件正在被迅雷盗窃！ @ cnBeta》
《迅雷被疑盗窃用户文件 @ 网易科技》

顺便说一下：这种搞法除了引发隐私问题，还会（在未经你许可的情况下）浪费你的网络带宽和硬盘性能。从某种意义上讲，这就是“耍流氓”。

◇收集你的下载网址——并有可能暴露你的【私有网址】

跟“360安全浏览器”类似，迅雷也会把每一个用户的每一个下载网址收集到自己的服务器上。迅雷公司这么干是为了尽可能多地收集“下载源”，但也导致了一个严重的隐私问题——迅雷公司掌握了你所有的下载历史。具体请看维基百科的词条。

如果你用迅雷下载的网址是一个【私有网址】，那么这个私有网址不但会被其他迅雷用户看到，还会被别人用来下载。其中的危害参见“这篇网文”。

◇其它的隐私问题

除了这两个，还有一些迅雷软件的功能也会导致隐私问题，具体请看下面这篇：
《不看不知道——迅雷正在疯狂的吞噬你的隐私》（这篇文章是2009年的，其中的描述可能跟当前的迅雷版本有一定出入）

★媒体播放类

最后再来聊聊媒体播放类的软件。媒体播放软件主要有“视频播放”和“音乐播放”两大类（当然，也有些是二合一的）。不论是“视频播放”和“音乐播放”，播放软件都【有可能】收集你播放的媒体信息，然后发送到厂商的服务器上。

举例：

请看2007年的一篇报道——《暴风排行榜正式发布 1.4亿用户观影习惯揭晓 @ 网易科技》。暴风影

音是如何统计出几亿用户的“观影习惯”？它收集用户的“观影习惯”，得到用户许可了吗？

经读者在评论中提醒，除了上述这种方式（播放器主动发送媒体信息），还有一种方式是利用“自动下载字幕”这一功能。比如“射手影音”提供的“智能字幕”功能，会自动去射手网匹配字幕。此时，字幕服务器会收集到你正在播放的视频信息。

★其它软件的隐私问题

前面俺列出了几类特别有代表性的桌面软件导致的隐私问题。千万【不要】以为，只有这几类软件会导致隐私问题。**其它类别的软件，同样也可能有隐私问题。**

举例：

比如前几天被曝光的“支付宝监控网络丑闻”，率先曝光的是“这篇博文”（此文已被广为流传）。

阿里巴巴旗下的支付宝安全控件，同样是一个装机量非常巨大（亿级）的桌面软件。考虑到阿里巴巴在电子商务和在线支付的市场份额，支付宝控件的装机量说不定比 360 还高。这么大的装机量，阿里巴巴动了邪念，那是很正常滴（马云本来就没啥人品可言）。退一步讲，就算阿里巴巴没有动邪念，朝廷也会动邪念滴——此中的道理，前面已经解释过了。

★结尾

本文列举了几类特别具有代表性的软件类型（分别是：安全类、输入法类、即时通信类、下载类、媒体播放类）。如果你认为其它某个分类也很有代表性，欢迎到本文留言。

在本系列的下一篇，俺会来介绍一下，如何反击桌面软件偷窥隐私的流氓行为。另外，有些读者可能会问：为啥本文聊了“即时通信”，但是却没有聊“电子邮件”和“社交网络”。因为俺觉得这两类更偏重于“服务器端”，所以在本系列的下下篇，再来单独聊“互联网服务”导致的隐私问题。俺初步考虑，会介绍“搜索引擎、电子邮件、社交网络、网盘、……”的隐私问题。

[如何保护隐私8]：流氓的桌面软件有哪些替代品？

文章目录

在本系列的前一篇博文，已经介绍了各种桌面软件（尤其是国产商业软件）存在的隐私问题。今天这篇介绍一下这些软件的替代品。**有些软件比较难以替代，会在本系列的下一篇介绍应对措施。**

本文中推荐的每一款工具，都附有维基百科的链接。打开维基百科的链接，就可以看到该工具的【官网链接】。

★“首选替代品”和“次选替代品”的差异

在本文中，俺会介绍常见软件类型的各种替代品。针对每一个大类，会同时给出“首选替代品”和“次选替代品”。这两者的差异如下：

首选替代品

主要是【国外】非营利机构和组织开发的【开源】软件。

这类软件的特点是——通常情况下，既【没有】咱们朝廷植入的后门，也【没有】欧美政府（比如NSA）的后门

次选替代品

主要是【国外】商业公司开发的【闭源】软件。

这类软件的特点是——通常情况下，【没有】咱们朝廷植入的后门，但【可能有】欧美政府的后门。另外，由于是商业公司开发的，在“隐私保护”方面不如“非营利组织开发的软件”（具体理由参见本系列第1篇）。

★安全类（防病毒、防木马）

◇【不】靠谱的软件有哪些?

各种国内商业公司的防病毒、防木马工具，至少包括“奇虎、瑞星、金山、江民、微点、腾讯、百度”等公司的产品。

◇首选替代品

（在首选替代品中列出的，全都是开源项目）

1. **改用 Linux，无需安装“防病毒/防木马”软件**
   如果你还在用 Windows，并且饱受病毒和木马的困扰，是时候考虑换成 Linux 啦。如今的 Linux 已经很成熟了，主流的发行版对硬件的支持足够好，而且 Linux 上有一款神器——Wine——它可以让你在 Linux 下运行 Windows 的软件，估计 90% 以上的常用软件（包括“翻墙代理、游戏、媒体播放”等）它都支持。就算碰到少数不支持的，咱们还可以玩【操作系统虚拟机】嘛。没玩过“操作系统虚拟机”的同学，可以看俺博客上的系列扫盲教程，链接在“这里”。
   由于 Linux 在桌面系统还是小众的，所以病毒和木马的作者都不会考虑 Linux 平台。另外，如果你遵守一些安全规范（比如：不用 root 进行日常操作），基本上就不用担心中病毒和木马。
   没用过 Linux 的同学可以看俺之前写的扫盲教程《新手如何搞定 Linux 操作系统？》。
2. **ClamAV / ClamWin**
   防病毒软件中，彻底开源的不多，今天给大伙儿天推荐的 ClamAV 就是一款**【彻底开源】**的防病毒软件。别看它是开源的，资历比某些商业杀毒软件还要老（诞生于1998年），而且查毒的功能毫不逊色。ClamAV 本身是命令行界面的，ClamWin 提供了 Windows 上的图形界面，内嵌 ClamAV 引擎。
   下面列举它的几个特色。
   特色1——ClamAV 和 ClamWin 性能开销很小，号称是最低功耗的“静音杀毒软件”。
   特色2——ClamAV 支持的操作系统非常多，除了三大主流的桌面系统，还支持 BSD、Solaris、AIX、HP-UX、OpenVMS、OS/2 等。
   特色3——ClamWin 支持“便携模式”（做成“绿色软件”），就可以放在“U盘或光盘”上即插即用。官网的说明在“这里”。
   补充说明：
   ClamAV 只有“查毒”功能，没有“杀毒”功能（对已发现的病毒文件，用户可以选择“删除”或“隔离”）。不过这无伤大雅。为啥捏？一旦某个软件感染了病毒，能否【彻底】杀掉，是一个问题？

杀掉之后，这个软件是否还能正常工作，又是一个问题。所以俺的经验是：一旦某个系统查出病毒/木马，最安全的处理措施就是——重装系统。综上所述，“杀毒功能”对防病毒软件而言，简直就是鸡肋。而 ClamAV 只提供“查毒”，不提供“杀毒”，反而让它不至于太臃肿。

本文发布后，有热心读者反馈说：ClamAV 的误报率偏高。如果你担心“开源的防病毒”不够好，可以参考“次选替代品”中列举的免费商业软件。

◇次选替代品

（在“次选替代品”中列出的，【不是】开源项目）

下面再介绍几款【国外】的防病毒/防木马产品。因为这些安全产品【不开源】，所以放到“次选替代品”。

考虑到咱们天朝的网民都不喜欢花钱买正版，所以特地介绍几款免费的防病毒和防木马产品（别以为只有 360 是免费滴，国外免费的同类产品也不少哦）。

1. **Avira / AntiVir（小红伞）**
   这款工具来自德国，诞生于1988年，“AntiVir”是之前的名字。它面向家庭用户和非营利组织的版本是免费的。
   免费版本支持：Windows、Android、Mac OS X
   据俺所知，小红伞在国内用户的口碑不错，所以排第一。
2. **Comodo（科摩多）**
   该公司1998年成立于英国，如今总部在美国。
   免费版本支持：Windows、Android、Linux、Mac OS X
   Comodo 提供的免费安全工具，种类挺多的（AV、FW、IPS、VPN），所以把它排在第二。
3. **AVG**
   这款工具来自捷克，诞生于1997。与小红伞类似，它面向家庭用户和非营利组织的版本是免费的。
   免费版本支持：Windows、Android
4. **avast（爱维士）**
   这款工具跟 AVG 是老乡（也是来自捷克），诞生于1988年。它的免费版本面向家庭用户。
   免费版本支持：Windows、Mac OS X

◇备注

如果你对上述列出的防病毒/防木马软件不中意，可以去看维基百科的“这个页面”，里面列出了几十种防病毒/防木马产品的详细对比。

★安全类（主机防火墙）

◇【不】靠谱的软件有哪些?

各种国内商业公司的主机防火墙，至少包括“奇虎、瑞星、金山、江民”等。

◇替代品

对 Windows 用户而言，你直接用 Windows 自带的防火墙，基本上就能满足需求了。早在 WinXP，就内置了带图形配置界面的防火墙。到了 Vista 之后，内置防火墙的功能又增加了不少。如今，你可以通过 Windows 内置的防火墙实现如下功能：
\1. 双向过滤（对内流量、对外流量）
\2. 针对某些端口的过滤
\3. 针对某些进程的过滤
\4. ……

对 Linux 用户而言，Linux 内核已经提供了 iptables / nftables，功能足够你用了。其中的 iptables 从 2.4 版本开始整合到 Linux kernel 中。而 nftables 是作为 iptables 的替代品，从今年（2014）1月份开始合并到内核主线。

◇为啥俺推荐系统内置的防火墙?

首先，操作系统内置的防火墙在性能、稳定性、兼容性等方面，至少【不会】比第三方的防火墙更差。
其次，操作系统内置的防火墙，它的开发者也就是操作系统的开发者。所以，【不会】引入额外的隐私风险。
最后，很多商业安全公司（包括国外的），它们提供的"个人主机防火墙"，其宣传的功能大都是唬人的噱头，华而不实，说不定还影响系统的网络性能。

★浏览器

关于"浏览器的选择"，已经在本系列的第2篇分析过了，此处不再罗嗦。

★输入法

◇【不】靠谱的软件有哪些?

某些国内商业公司的输入法，至少包括"搜狗拼音、华宇拼音（原"紫光拼音"）、拼音加加、QQ拼音、QQ五笔"等输入法。

◇首选替代品

（在首选替代品中列出的，全都是开源项目）

1. **Rime 输入法（中州韵、小狼毫、鼠须管）**
   这款输入法诞生没多久（2011年），原作者是天朝网友，网名"佛振"。
   操作系统支持：Windows、Linux、Mac OS X。该输入法有三个中文名，分别对应这三个操作系统——Linux 下称为"中州韵"，Windows 下称为"小狼毫"；Mac OS X 下称为"鼠须管"。
   输入方案：支持十多种输入方案（除了支持多种拼音方案，还包括两种"五笔"，另有一些俺从未听说过滴）。
2. **Fcitx 输入法（小企鹅）**
   这是一个 X Window 下的输入法框架，诞生于2004年，原作者是天朝网友，网名"Yuking"。

操作系统支持：类 Unix（Linux、BSD 等）
输入方案：多种拼音、码表（五笔、郑码、仓颉 等）、日文、韩文、手写输入

3. **iBus**
这也是一个输入法框架，诞生于2008年，原作者是黄鹏。
操作系统支持：类 Unix（Linux、BSD 等）
输入方案：多种拼音、码表（五笔、郑码、仓颉 等）、日文、韩文

◇次选替代品

（在“次选替代品”中列出的，【不是】开源项目）

1. **谷歌拼音输入法**
这款是2007年诞生的，由谷歌中国开发的。
操作系统支持：Windows、Android
该输入法具有“云端同步词库”的功能。如果你担心 Google 偷窥你的个性化词库，可以禁用该功能。
2. **微软必应输入法**
这款是2012年诞生的，由微软亚洲研究院开发。原先叫做“英库拼音输入法”。
据说整合了研究院的多项研究成果。至于效果如何，因为俺没用过，不好评价。虽然微软亚洲研究院在北京，不过俺估计：应该没被朝廷渗透 :)
操作系统支持：Windows、Android
该输入法具有“云端同步词库”的功能。如果你担心微软偷窥你的个性化词库，可以禁用该功能。

◇备注

有些网友已经用惯了某个国产输入法，不愿意切换到其它输入法。对这种情况，俺会在本系列的下一篇介绍几种应对措施。

★聊天类（IM）

◇【不】靠谱的软件有哪些?

某些国内商业公司的聊天工具，至少包括“腾讯QQ、网易泡泡、阿里旺旺”等。

◇首选替代品

（在首选替代品中列出的，全都是开源项目）

1. **Pidgin**
这款是很老牌的开源 IM 软件，诞生于1998年，原先叫做“Gaim”。
操作系统支持：Windows、Linux、Mac OS X
协议支持：它支持的 IM 协议很多，分“官方支持”和“第三方插件支持”。对于“XMPP、SIP、MSN、AIM/ICQ、YIM、IRC、SILC”是官方支持，对于“Skype、QQ、飞信、Napster”是第三方插件支持。估计很多人对“支持 QQ 协议”比较感兴趣。原先 Pidgin 直接支持 QQ 协议，后来腾讯封杀了 Pidgin 客户端。目前对 QQ 的支持需要依靠“pidgin-lwqq”这个插件（此插件也是开源滴，代码库在“这里”）。据说这个插件利用的是 webQQ 的协议，所以疼逊比较难封杀。

加密支持：有一个 OTR（Off-the-Record Messaging）插件，还有一个“Pidgin-Encryption”插件。

2. **Psi**

这款诞生于2001年，基于 C++ 开发，采用 Qt 的 GUI 库。

操作系统支持：Windows、Linux、Mac OS X

协议支持：它重点支持 XMPP 协议，可以通过 XMPP 网关跟其它的聊天服务（MSN、AIM/ICQ、Yahoo Messenger 等）互联

加密支持：内置 GnuPG 对消息进行加密；支持 OTR（Off-the-Record Messaging）插件。

3. **Gajim**

这款诞生于2004年，基于 Python 开发。

操作系统支持：由于是采用 Python 语言开发，凡是能运行 Python 的系统都支持。

协议支持：情况跟 Psi 类似。

加密支持：支持 TLS/SSL 和 OpenPGP，支持 OTR（Off-the-Record Messaging）插件。

4. **Linphone**

这款诞生于2001年。一开始只是 Windows 桌面软件，后来陆续开发了各种手机版本。到了2013年开始支持浏览器，称为“Linphone Web”。

操作系统支持：Windows、Linux、Mac OS X、FreeBSD、Android、iOS、BlackBerry OS

协议支持：SIP

加密支持：支持 TLS、ZRTP、SRTP

5. **Kopete**

这款诞生于2001年，基于 C++ 开发。看名称就能猜到，它采用的是 KDE 的 GUI（基于 Qt 开发）。

操作系统支持：类 Unix 的操作系统（Linux、Mac OS X），貌似不支持 Windows。

协议支持：XMPP、Skype、MSN、AIM/ICQ、YIM（Yahoo）

加密支持：支持 OTR（Off-the-Record Messaging）插件，还有个“kopete-cryptography”插件。

6. **Jitsi**

这款诞生于2003年，基于 Java 开发。

操作系统支持：由于是采用 Java 语言开发，凡是能运行 Java 的系统都支持。

协议支持：XMPP、SIP、MSN、AIM/ICQ、YIM（Yahoo）

加密支持：支持 OTR（Off-the-Record Messaging）插件。

7. **Tox**

这款是后起之秀，去年（2013）才诞生的（“棱镜门丑闻”催生了它）。虽然刚诞生不久，但是很火。它的目标是——提供一个无法监控的 Skype 替代品——彻底的加密，没有后门，无需中间服务器。

严格来讲它只是一个开源的协议框架，不同的开发人员可以根据该协议开发出不同的客户端。比如 uTox 是 Windows 的客户端；Venom 是采用 GTK+ 界面库的 Linux 客户端；qTox 是采用 Qt 界面库的 Mac OS X 客户端；而 Antox 是 Android 上的 Tox 客户端……这个页面列出了各种各样的 Tox 客户端（十多种）。另外，还有一个“Pidgin 插件”，你可以用它在 Pidgin 上进行 Tox 聊天。

操作系统支持：Windows、Linux、Mac OS X、Android、iOS

协议支持：跟前面几款不同，它采用的是自己独有的“Tox protocol”。

加密支持：采用 NaCl 库对所有通讯流量进行加密。

补充：Tox 出来的时间还很短（不到两年），俺估计成熟度还有待提高。先不要急着用它。

8. **Bitmessage（比特信）**

这款跟 Tox 很类似，也是去年才诞生的，也是完全依赖于 P2P（无需中心服务器），也是强调加密，也是为了对抗政府的监控。从0.3.5版本开始，它提供了 Chans 功能——相当于匿名化的邮件列表（同样无需中心服务器）。

操作系统支持：基于 Python 编写，跨平台

协议支持：跟前面几款不同，它采用的是自己独有的通讯协议。

加密支持：基于公钥加密体系，对所有通讯流量进行加密

补充：Bitmessage 出来的时间还很短（不到两年），俺估计成熟度还有待提高。先不要急着用它。

◇次选替代品

（在“次选替代品”中列出的，【不是】开源项目）

1. **Google Hangouts（环聊）**
   这款是 Google 在2013年发布的 IM 工具，它用来取代 Google Talk 滴。
   这玩意儿提供“纯网页版本”——这种情况下无需安装桌面客户端，于是就能避免桌面软件导致的隐私问题。但是服务端依然存在隐私问题——Google 会看到你的聊天内容。
2. **Skype 国际版**
   Skype 曾经是 eBay 旗下，2011年被转手卖给微软。
   它的优点在于：用户数量众多，而且语音聊天的质量也挺好。缺点在于：据未经证实的传闻，Skype 的客户端可能有 NSA 的后门。退一步讲，就算客户端没有后门，但是服务端依然存在隐私问题。
   另外再唠叨一下：如果想用 Skype，一定要用【国际版】的 Skype，千万【不要用】“TOM 版”和“光明版”。在2013年11月之前，Skype 的中方合作伙伴是 tom.com（李嘉诚旗下）。在过去几年，“TOM 版 Skype”已经被发现有严重的后门（会对聊天文本进行监控）。2013年11月之后，Skype 换了一个新的中国合作伙伴——光明方正（《光明日报》与北大方正的合资公司），于是有了一个新的“光明版 Skype”。考虑到《光明日报》是党国喉舌，这个“光明版 Skype”也不靠谱。

◇备注

除非是纯 P2P 的 IM 软件（比如 Tox），否则的话，除了要考虑客户端的隐私问题，还要考虑服务端的隐私问题。

打个比方：如果你用 Pidgin 跟别人进行 QQ 聊天。虽然 Pidgin 作为桌面软件本身是靠谱，但腾讯的【服务器】是不靠谱滴。这种情况下，腾讯依然有可能偷窥到你的隐私。（在本系列后续博文，俺会介绍互联网服务的隐私问题）。

★下载类

◇【不】靠谱的软件有哪些?

某些国内商业公司的下载软件，至少包括“迅雷、FlashGet（快车）、QQ旋风、VeryCD”等。

◇首选替代品

（在首选替代品中列出的，全都是开源项目）

1. **Free Download Manager（FDM）**
   这是一款多协议的下载工具，诞生于2004年。
   操作系统支持：Windows
   协议支持：HTTP、FTP、BT、RTSP/MMS
   特色：能从视频网站（比如 YouTube）下载 Flash 视频；它的“HTML Spider”功能可以抓取整个网站的页面（类似搜索引擎爬虫）；

补充说明：如果安装之后碰到中文界面乱码，解决方法是——安装过程中先选择“English”，安装完成后选主菜单“View”再选“Language”切换成中文。

2. **Shareaza**
这是一款老牌的 P2P 工具，诞生于2000年。它的原作者同时也是 Gnutella2（简称“G2”）协议的作者。
操作系统支持：Windows
协议支持：Gnutella、G2、eDonkey、BT、HTTP、FTP
特色：支持的协议算是比较全的，界面挺花哨
3. **MLDonkey**
这是一款支持多协议的下载工具，诞生于2001年。刚开始只是一个 Linux 工具，只支持 eD2k（电驴网络），后来扩展成跨平台并支持多种网络协议。
它本身是只有命令行界面。不习惯命令行的，可以去找第三方的图形界面前端（比如“Sancho”）。使用前需要先进行一些设置。总的来说，更适合懂技术的网友，而不适合新手。
操作系统支持：Windows、Linux、Mac OS X
协议支持：eDonkey、Kad、BT、HTTP、FTP
特色：下载 eD2k/Kad 资源时，可以同时连多个 emule 服务器
4. **eMule（电骡）**
这款 P2P 工具诞生于2002年，一度是很受欢迎的开源下载工具（截止2009年9月，eMule 在 SourceForge 的下载数超过5亿）。不过最近几年的开发不活跃了。
操作系统支持：Windows
协议支持：eDonkey、Kad
5. **Transmission**
这款是 BT 的客户端，诞生于2005年。
操作系统支持：Linux、Mac OS X、BSD、Windows（对 Windows 的支持依靠“Transmission-Qt for Windows”）
协议支持：BT
特色：被众多 Linux 发行版（包括 Ubuntu、Mandriva、Linux Mint、Fedora、Puppy Linux、openSUSE 选作默认 BT 下载工具）
6. **aMule**
这款 P2P 工具诞生于2003年，看名字就知道它的功能跟 eMule 很像。
操作系统支持：Windows、Mac OS X、类 Unix
协议支持：eDonkey、Kad
7. **qBittorrent**
这款是 BT 客户端，诞生于2006年，支持的操作系统比较多，难得还能支持 Android。
操作系统支持：Windows、Linux、Mac OS X、FreeBSD、Android
协议支持：BT

◇次选替代品

（在“次选替代品”中列出的，【不是】开源项目）

1. **µTorrent**
看名称就知道，这是 BT 的客户端。诞生于2005年。
操作系统支持：Windows、Linux、Mac OS X、Android
协议支持：BT
特色：非常轻量级（Windows 下的3.4.2版本竟然只有1兆多）
2. **Tixati**
这款是 BT 客户端，诞生于2009年，名气不大。
操作系统支持：Windows、Linux
协议支持：BT
特色：系统资源占用小，有丰富的图表展示界面

◇备注

很多天朝的网友使用迅雷是为了下载迅雷协议（thunder://）的网址。迅雷协议不是开放的，暂时没有太好的替代品，你不得不继续使用迅雷客户端。在这种情况下，你需要采用一些安全措施来防止迅雷客户端。本系列的下一篇博文会介绍各种隐私方面的防范措施。

如果你对上述列出的下载软件不中意，可以去参考维基百科的如下几个页面：

各种 BT 客户端比较
各种电驴客户端比较
各种文件分享工具比较
各种下载管理器比较

★媒体播放类

◇【不】靠谱的软件有哪些?

某些国内商业公司的媒体播放软件，至少包括"暴风影音、百度音乐（原"千千静听"）、QQ 音乐、射手影音"等。

◇首选替代品

（在首选替代品中列出的，全都是开源项目）

1. **VLC**
   这款诞生于2001年，也算是比较老牌的。在开源播放器中，它估计是最受欢迎的，截止俺写本文时，累计下载量超过13亿人次（参见"这个页面"）。另外，在所有的播放器中（包括开源和非开源），VLC 是支持操作系统最多的。所以俺把它排在第一位。
   操作系统支持：Windows、Linux、Mac OS X、Android、iOS、BSD、BeOS、Solaris、OS/2、DOS、Syllable、QNX
   格式支持：各种常见的视频、音频格式（你能想到的，应该都支持）
   特色：除了支持的操作系统最多，VLC 估计还是兼容性最好的。比如俺在一个干净的 WinThinPC 系统中测试，VLC 可以正常播放视频，而 SMPlayer 和 MPC-HC 都出现错误提示。
2. **MPlayer / SMPlayer**
   MPlayer 诞生于2000年，是一个视频播放的后端（命令行界面）。它可以跟几种不同的 GUI 前端搭配，比较有名的是前端是 SMPlayer。SMPlayer 诞生于2006年，是比较轻量级的，它的安装包（内置 MPlayer）比 VLC 小，Windows 下大概10多兆。
   操作系统支持：SMPlayer 支持 Windows、Linux、BSD
   格式支持：各种常见的视频、音频格式（你能想到的，应该都支持）
3. **Kodi（XBMC）**
   这款诞生于2002年。刚开始是打算做一个运行在 XBox 之上的播放器，所以原先名叫"XBMC"（XBox Media Center）。后来支持的平台越来越多了，XBox 反而不是重点了，于是改名叫 Kodi。
   操作系统支持：Windows、Linux、Mac OS X、Android、iOS、Apple TV OS、BSD
   格式支持：各种常见的视频、音频格式（你能想到的，应该都支持）
   特色：它采用基于 Python 的插件来扩展功能，甚至可以在它上面运行 Python 写的小游戏。
4. **MPC-HC**
   先来说一下 MPC——这是某个老外在2003年打造的轻量级播放器，界面完全模仿 Windows Media Player 6.4 版本。一开始是闭源的，后来完全开源了。可惜在2006年停止开发。然后另一

个老外根据 MPC 的源代码，衍生出一个新的开源项目 MPC-HC 并继续开发至今。

操作系统支持：Windows

格式支持：各种常见的视频、音频格式（你能想到的，应该都支持）

特色：界面风格是它的特色，继续保留原先 Windows Media Player 6.4版本的风格，适合喜欢复古的网友。

◇次选替代品

（在"次选替代品"中列出的，【不是】开源项目）

foobar2000

这款诞生于2002年，是免费的音乐播放软件，而且很小巧（目前的1.3.3版本只有3兆多）。

操作系统支持：Windows（XP 或更高）

格式支持：各种常见的音频格式，至少包括：MP1、MP2、MP3、MPC、AAC、WMA、Ogg Vorbis、FLAC/Ogg FLAC、ALAC、WavPack、WAV、AIFF、AU、SND、CD、Speex、Opus

特色：可以直接播放压缩包里面的音频文件（无需解压）。

◇备注

如果你对上述列出的媒体播放软件不中意，可以去看维基百科的"这个页面"，里面列出了几十种媒体播放软件的详细对比。

★邮件类（客户端）

◇【不】靠谱的软件有哪些?

某些国产的邮件客户端，比如 Foxmail。

◇首选替代品

（在首选替代品中列出的，全都是开源项目）

Mozilla Thunderbird（雷鸟）

这款是 Firefox 的同门兄弟——它俩都出自大名鼎鼎的 Mozilla 门下。Mozilla 是个非盈利组织，而不是商业公司。所以 Mozilla 旗下的软件是比较靠谱滴。

它估计是名气最大的开源邮件客户端，功能齐全——基本上你能想到的，它都有了。而且具有很强的可定制性。

操作系统支持：Windows、Linux、Mac OS X

加密支持：支持 S/MIME 标准，可以通过 Enigmail 插件进行公钥加密和签名（基于 OpenPGP）。

◇次选替代品

（在“次选替代品”中列出的，【不是】开源项目）

微软的 Outlook 客户端

（微软有两款 Outlook，此处泛指这两款）对于微软的 Outlook，大伙儿应该都比较熟悉，俺就不介绍了。

相比国产的邮件客户端，微软的 Outlook 可以避免被党国植入后门。但是微软的 Outlook 依然有可能被 NSA 植入后门。老实说，俺没觉得有啥理由可以让你“不选 Thunderbird 而去选 Outlook”。

★结尾

俺的见识很有限，所以本文介绍的替代品，肯定不全面。大伙儿如有其它补充，欢迎到本文留言。[如何保护隐私8]：流氓的桌面软件有哪些替代品？

文章目录

在本系列的前一篇博文，已经介绍了各种桌面软件（尤其是国产商业软件）存在的隐私问题。今天这篇介绍一下这些软件的替代品。**有些软件比较难以替代，会在本系列的下一篇介绍应对措施。**

本文中推荐的每一款工具，都附有维基百科的链接。打开维基百科的链接，就可以看到该工具的【官网链接】。

★“首选替代品”和“次选替代品”的差异

在本文中，俺会介绍常见软件类型的各种替代品。针对每一个大类，会同时给出“首选替代品”和“次选替代品”。这两者的差异如下：

首选替代品

主要是【国外】非营利机构和组织开发的【开源】软件。

这类软件的特点是——通常情况下，既【没有】咱们朝廷植入的后门，也【没有】欧美政府（比如NSA）的后门

次选替代品

主要是【国外】商业公司开发的【闭源】软件。

这类软件的特点是——通常情况下，【没有】咱们朝廷植入的后门，但【可能有】欧美政府的后门。另外，由于是商业公司开发的，在“隐私保护”方面不如“非营利组织开发的软件”（具体理由参见本系列第1篇）。

★安全类（防病毒、防木马）

◇【不】靠谱的软件有哪些?

各种国内商业公司的防病毒、防木马工具，至少包括“奇虎、瑞星、金山、江民、微点、腾讯、百度”等公司的产品。

◇首选替代品

（在首选替代品中列出的，全都是开源项目）

1. **改用 Linux，无需安装“防病毒/防木马”软件**
   如果你还在用 Windows，并且饱受病毒和木马的困扰，是时候考虑换成 Linux 啦。如今的 Linux 已经很成熟了，主流的发行版对硬件的支持足够好，而且 Linux 上有一款神器——Wine——它可以让你在 Linux 下运行 Windows 的软件，估计 90% 以上的常用软件（包括“翻墙代理、游戏、媒体播放”等）它都支持。就算碰到少数不支持的，咱们还可以玩【操作系统虚拟机】嘛。没玩过“操作系统虚拟机”的同学，可以看俺博客上的系列扫盲教程，链接在“这里”。
   由于 Linux 在桌面系统还是小众的，所以病毒和木马的作者都不会考虑 Linux 平台。另外，如果你遵守一些安全规范（比如：不用 root 进行日常操作），基本上就不用担心中病毒和木马。
   没用过 Linux 的同学可以看俺之前写的扫盲教程《新手如何搞定 Linux 操作系统？》。
2. **ClamAV / ClamWin**
   防病毒软件中，彻底开源的不多，今天给大伙儿天推荐的 ClamAV 就是一款**【彻底开源】**的防病毒软件。别看它是开源的，资历比某些商业杀毒软件还要老（诞生于1998年），而且查毒的功能毫不逊色。ClamAV 本身是命令行界面的，ClamWin 提供了 Windows 上的图形界面，内嵌 ClamAV 引擎。
   下面列举它的几个特色。
   特色1——ClamAV 和 ClamWin 性能开销很小，号称是最低功耗的“静音杀毒软件”。
   特色2——ClamAV 支持的操作系统非常多，除了三大主流的桌面系统，还支持 BSD、Solaris、AIX、HP-UX、OpenVMS、OS/2 等。
   特色3——ClamWin 支持“便携模式”（做成“绿色软件”），就可以放在“U盘或光盘”上即插即用。官网的说明在“这里”。
   补充说明：
   ClamAV 只有“查毒”功能，没有“杀毒”功能（对已发现的病毒文件，用户可以选择“删除”或“隔离”）。不过这无伤大雅。为啥捏？一旦某个软件感染了病毒，能否【彻底】杀掉，是一个问题？杀掉之后，这个软件是否还能正常工作，又是一个问题。所以俺的经验是：一旦某个系统查出病毒/木马，最安全的处理措施就是一一重装系统。综上所述，“杀毒功能”对防病毒软件而言，简直就是鸡肋。而 ClamAV 只提供“查毒”，不提供“杀毒”，反而让它不至于太臃肿。
   本文发布后，有热心读者反馈说：ClamAV 的误报率偏高。如果你担心“开源的防病毒”不够好，可以参考“次选替代品”中列举的免费商业软件。

◇次选替代品

（在“次选替代品”中列出的，【不是】开源项目）

下面再介绍几款【国外】的防病毒/防木马产品。因为这些安全产品【不开源】，所以放到“次选替代品”。

考虑到咱们天朝的网民都不喜欢花钱买正版，所以特地介绍几款免费的防病毒和防木马产品（别以为只有 360 是免费滴，国外免费的同类产品也不少哦）。

1. **Avira / AntiVir（小红伞）**
   这款工具来自德国，诞生于1988年，“AntiVir”是之前的名字。它面向家庭用户和非营利组织的版本是免费的。

免费版本支持：Windows、Android、Mac OS X
据俺所知，小红伞在国内用户的口碑不错，所以排第一。

2. **Comodo（科摩多）**
该公司1998年成立于英国，如今总部在美国。
免费版本支持：Windows、Android、Linux、Mac OS X
Comodo 提供的免费安全工具，种类挺多的（AV、FW、IPS、VPN），所以把它排在第二。
3. **AVG**
这款工具来自捷克，诞生于1997。与小红伞类似，它面向家庭用户和非营利组织的版本是免费的。
免费版本支持：Windows、Android
4. **avast（爱维士）**
这款工具跟 AVG 是老乡（也是来自捷克），诞生于1988年。它的免费版本面向家庭用户。
免费版本支持：Windows、Mac OS X

◇备注

如果你对上述列出的防病毒/防木马软件不中意，可以去看维基百科的“这个页面”，里面列出了几十种防病毒/防木马产品的详细对比。

★安全类（主机防火墙）

◇【不】靠谱的软件有哪些?

各种国内商业公司的主机防火墙，至少包括“奇虎、瑞星、金山、江民”等。

◇替代品

对 Windows 用户而言，你直接用 Windows 自带的防火墙，基本上就能满足需求了。早在 WinXP，就内置了带图形配置界面的防火墙。到了 Vista 之后，内置防火墙的功能又增加了不少。如今，你可以通过 Windows 内置的防火墙实现如下功能：
\1. 双向过滤（对内流量、对外流量）
\2. 针对某些端口的过滤
\3. 针对某些进程的过滤
\4. ……

对 Linux 用户而言，Linux 内核已经提供了 iptables / nftables，功能足够你用了。其中的 iptables 从 2.4 版本开始整合到 Linux kernel 中。而 nftables 是作为 iptables 的替代品，从今年（2014）1月份开始合并到内核主线。

◇为啥俺推荐系统内置的防火墙?

首先，操作系统内置的防火墙在性能、稳定性、兼容性等方面，至少【不会】比第三方的防火墙更差。
其次，操作系统内置的防火墙，它的开发者也就是操作系统的开发者。所以，【不会】引入额外的隐私风险。
最后，很多商业安全公司（包括国外的），它们提供的“个人主机防火墙”，其宣传的功能大都是唬人的噱头，华而不实，说不定还影响系统的网络性能。

★浏览器

关于“浏览器的选择”，已经在本系列的第2篇分析过了，此处不再罗嗦。

★输入法

◇【不】靠谱的软件有哪些?

某些国内商业公司的输入法，至少包括“搜狗拼音、华宇拼音（原“紫光拼音”）、拼音加加、QQ拼音、QQ五笔”等输入法。

◇首选替代品

（在首选替代品中列出的，全都是开源项目）

1. **Rime 输入法（中州韵、小狼毫、鼠须管）**
   这款输入法诞生没多久（2011年），原作者是天朝网友，网名“佛振”。
   操作系统支持：Windows、Linux、Mac OS X。该输入法有三个中文名，分别对应这三个操作系统——Linux 下称为“中州韵”，Windows 下称为“小狼毫”；Mac OS X 下称为“鼠须管”。
   输入方案：支持十多种输入方案（除了支持多种拼音方案，还包括两种“五笔”，另有一些俺从未听说过滴）。
2. **Fcitx 输入法（小企鹅）**
   这是一个 X Window 下的输入法框架，诞生于2004年，原作者是天朝网友，网名“Yuking”。
   操作系统支持：类 Unix（Linux、BSD 等）
   输入方案：多种拼音、码表（五笔、郑码、仓颉 等）、日文、韩文、手写输入
3. **iBus**
   这也是一个输入法框架，诞生于2008年，原作者是黄鹏。
   操作系统支持：类 Unix（Linux、BSD 等）
   输入方案：多种拼音、码表（五笔、郑码、仓颉 等）、日文、韩文

◇次选替代品

（在“次选替代品”中列出的，【不是】开源项目）

1. **谷歌拼音输入法**
   这款是2007年诞生的，由谷歌中国开发的。
   操作系统支持：Windows、Android
   该输入法具有“云端同步词库”的功能。如果你担心 Google 偷窥你的个性化词库，可以禁用该功能。
2. **微软必应输入法**
   这款是2012年诞生的，由微软亚洲研究院开发。原先叫做“英库拼音输入法”。
   据说整合了研究院的多项研究成果。至于效果如何，因为俺没用过，不好评价。虽然微软亚洲研究院在北京，不过俺估计：应该没被朝廷渗透 :)
   操作系统支持：Windows、Android
   该输入法具有“云端同步词库”的功能。如果你担心微软偷窥你的个性化词库，可以禁用该功能。

◇备注

有些网友已经用惯了某个国产输入法，不愿意切换到其它输入法。对这种情况，俺会在本系列的下一篇介绍几种应对措施。

★聊天类（IM）

◇【不】靠谱的软件有哪些?

某些国内商业公司的聊天工具，至少包括“腾讯QQ、网易泡泡、阿里旺旺”等。

◇首选替代品

（在首选替代品中列出的，全都是开源项目）

1. **Pidgin**
   这款是很老牌的开源 IM 软件，诞生于1998年，原先叫做“Gaim”。
   操作系统支持：Windows、Linux、Mac OS X
   协议支持：它支持的 IM 协议很多，分“官方支持”和“第三方插件支持”。对于“XMPP、SIP、MSN、AIM/ICQ、YIM、IRC、SILC”是官方支持，对于“Skype、QQ、飞信、Napster”是第三方插件支持。估计很多人对“支持 QQ 协议”比较感兴趣。原先 Pidgin 直接支持 QQ 协议，后来腾讯封杀了 Pidgin 客户端。目前对 QQ 的支持需要依靠“pidgin-lwqq”这个插件（此插件也是开源滴，代码库在“这里”）。据说这个插件利用的是 webQQ 的协议，所以疼逊比较难封杀。
   加密支持：有一个 OTR（Off-the-Record Messaging）插件，还有一个“Pidgin-Encryption”插件。
2. **Psi**
   这款诞生于2001年，基于 C++ 开发，采用 Qt 的 GUI 库。
   操作系统支持：Windows、Linux、Mac OS X
   协议支持：它重点支持 XMPP 协议，可以通过 XMPP 网关跟其它的聊天服务（MSN、AIM/ICQ、Yahoo Messenger 等）互联
   加密支持：内置 GnuPG 对消息进行加密；支持 OTR（Off-the-Record Messaging）插件。
3. **Gajim**
   这款诞生于2004年，基于 Python 开发。
   操作系统支持：由于是采用 Python 语言开发，凡是能运行 Python 的系统都支持。
   协议支持：情况跟 Psi 类似。
   加密支持：支持 TLS/SSL 和 OpenPGP，支持 OTR（Off-the-Record Messaging）插件。
4. **Linphone**
   这款诞生于2001年。一开始只是 Windows 桌面软件，后来陆续开发了各种手机版本。到了2013年开始支持浏览器，称为“Linphone Web”。
   操作系统支持：Windows、Linux、Mac OS X、FreeBSD、Android、iOS、BlackBerry OS
   协议支持：SIP
   加密支持：支持 TLS、ZRTP、SRTP
5. **Kopete**
   这款诞生于2001年，基于 C++ 开发。看名称就能猜到，它采用的是 KDE 的 GUI（基于 Qt 开发）。
   操作系统支持：类 Unix 的操作系统（Linux、Mac OS X），貌似不支持 Windows。
   协议支持：XMPP、Skype、MSN、AIM/ICQ、YIM（Yahoo）

加密支持：支持 OTR（Off-the-Record Messaging）插件，还有个“kopete-cryptography”插件。

6. **Jitsi**

这款诞生于2003年，基于 Java 开发。

操作系统支持：由于是采用 Java 语言开发，凡是能运行 Java 的系统都支持。

协议支持：XMPP、SIP、MSN、AIM/ICQ、YIM（Yahoo）

加密支持：支持 OTR（Off-the-Record Messaging）插件。

7. **Tox**

这款是后起之秀，去年（2013）才诞生的（“棱镜门丑闻”催生了它）。虽然刚诞生不久，但是很火。它的目标是——提供一个无法监控的 Skype 替代品——彻底的加密，没有后门，无需中间服务器。

严格来讲它只是一个开源的协议框架，不同的开发人员可以根据该协议开发出不同的客户端。比如 uTox 是 Windows 的客户端；Venom 是采用 GTK+ 界面库的 Linux 客户端；qTox 是采用 Qt 界面库的 Mac OS X 客户端；而 Antox 是 Android 上的 Tox 客户端……这个页面列出了各种各样的 Tox 客户端（十多种）。另外，还有一个“Pidgin 插件”，你可以用它在 Pidgin 上进行 Tox 聊天。

操作系统支持：Windows、Linux、Mac OS X、Android、iOS

协议支持：跟前面几款不同，它采用的是自己独有的“Tox protocol”。

加密支持：采用 NaCl 库对所有通讯流量进行加密。

补充：Tox 出来的时间还很短（不到两年），俺估计成熟度还有待提高。先不要急着用它。

8. **Bitmessage（比特信）**

这款跟 Tox 很类似，也是去年才诞生的，也是完全依赖于 P2P（无需中心服务器），也是强调加密，也是为了对抗政府的监控。从0.3.5版本开始，它提供了 Chans 功能——相当于匿名化的邮件列表（同样无需中心服务器）。

操作系统支持：基于 Python 编写，跨平台

协议支持：跟前面几款不同，它采用的是自己独有的通讯协议。

加密支持：基于公钥加密体系，对所有通讯流量进行加密

补充：Bitmessage 出来的时间还很短（不到两年），俺估计成熟度还有待提高。先不要急着用它。

◇次选替代品

（在“次选替代品”中列出的，【不是】开源项目）

1. **Google Hangouts（环聊）**

这款是 Google 在2013年发布的 IM 工具，它用来取代 Google Talk 滴。

这玩意儿提供“纯网页版本”——这种情况下无需安装桌面客户端，于是就能避免桌面软件导致的隐私问题。但是服务端依然存在隐私问题——Google 会看到你的聊天内容。

2. **Skype 国际版**

Skype 曾经是 eBay 旗下，2011年被转手卖给微软。

它的优点在于：用户数量众多，而且语音聊天的质量也挺好。缺点在于：据未经证实的传闻，Skype 的客户端可能有 NSA 的后门。退一步讲，就算客户端没有后门，但是服务端依然存在隐私问题。

另外再唠叨一下：如果想用 Skype，一定要用【国际版】的 Skype，千万【不要用】“TOM 版”和“光明版”。在2013年11月之前，Skype 的中方合作伙伴是 tom.com（李嘉诚旗下）。在过去几年，“TOM 版 Skype”已经被发现有严重的后门（会对聊天文本进行监控）。2013年11月之后，Skype 换了一个新的中国合作伙伴——光明方正（《光明日报》与北大方正的合资公司），于是有了一个新的“光明版 Skype”。考虑到《光明日报》是党国喉舌，这个“光明版 Skype”也不靠谱。

◇备注

除非是纯 P2P 的 IM 软件（比如 Tox），否则的话，除了要考虑客户端的隐私问题，还要考虑服务端的隐私问题。

打个比方：如果你用 Pidgin 跟别人进行 QQ 聊天。虽然 Pidgin 作为桌面软件本身是靠谱，但腾讯的【服务器】是不靠谱滴。这种情况下，腾讯依然有可能偷窥到你的隐私。（在本系列后续博文，俺会介绍互联网服务的隐私问题）。

★下载类

◇【不】靠谱的软件有哪些?

某些国内商业公司的下载软件，至少包括“迅雷、FlashGet（快车）、QQ旋风、VeryCD”等。

◇首选替代品

（在首选替代品中列出的，全都是开源项目）

1. **Free Download Manager（FDM）**
   这是一款多协议的下载工具，诞生于2004年。
   操作系统支持：Windows
   协议支持：HTTP、FTP、BT、RTSP/MMS
   特色：能从视频网站（比如 YouTube）下载 Flash 视频；它的“HTML Spider”功能可以抓取整个网站的页面（类似搜索引擎爬虫）；
   补充说明：如果安装之后碰到中文界面乱码，解决方法是——安装过程中先选择“English”，安装完成后选主菜单“View”再选“Language”切换成中文。
2. **Shareaza**
   这是一款老牌的 P2P 工具，诞生于2000年。它的原作者同时也是 Gnutella2（简称“G2”）协议的作者。
   操作系统支持：Windows
   协议支持：Gnutella、G2、eDonkey、BT、HTTP、FTP
   特色：支持的协议算是比较全的，界面挺花哨
3. **MLDonkey**
   这是一款支持多协议的下载工具，诞生于2001年。刚开始只是一个 Linux 工具，只支持 eD2k（电驴网络），后来扩展成跨平台并支持多种网络协议。
   它本身是只有命令行界面。不习惯命令行的，可以去找第三方的图形界面前端（比如“Sancho”）。使用前需要先进行一些设置。总的来说，更适合懂技术的网友，而不适合新手。
   操作系统支持：Windows、Linux、Mac OS X
   协议支持：eDonkey、Kad、BT、HTTP、FTP
   特色：下载 eD2k/Kad 资源时，可以同时连多个 emule 服务器
4. **eMule（电骡）**
   这款 P2P 工具诞生于2002年，一度是很受欢迎的开源下载工具（截止2009年9月，eMule 在 SourceForge 的下载数超过5亿）。不过最近几年的开发不活跃了。
   操作系统支持：Windows
   协议支持：eDonkey、Kad
5. **Transmission**
   这款是 BT 的客户端，诞生于2005年。
   操作系统支持：Linux、Mac OS X、BSD、Windows（对 Windows 的支持依靠“Transmission-

Qt for Windows”）
协议支持：BT
特色：被众多 Linux 发行版（包括 Ubuntu、Mandriva、Linux Mint、Fedora、Puppy Linux、openSUSE 选作默认 BT 下载工具）

6. **aMule**
这款 P2P 工具诞生于2003年，看名字就知道它的功能跟 eMule 很像。
操作系统支持：Windows、Mac OS X、类 Unix
协议支持：eDonkey、Kad
7. **qBittorrent**
这款是 BT 客户端，诞生于2006年，支持的操作系统比较多，难得还能支持 Android。
操作系统支持：Windows、Linux、Mac OS X、FreeBSD、Android
协议支持：BT

◇次选替代品

（在“次选替代品”中列出的，【不是】开源项目）

1. **µTorrent**
看名称就知道，这是 BT 的客户端。诞生于2005年。
操作系统支持：Windows、Linux、Mac OS X、Android
协议支持：BT
特色：非常轻量级（Windows 下的3.4.2版本竟然只有1兆多）
2. **Tixati**
这款是 BT 客户端，诞生于2009年，名气不大。
操作系统支持：Windows、Linux
协议支持：BT
特色：系统资源占用小，有丰富的图表展示界面

◇备注

很多天朝的网友使用迅雷是为了下载迅雷协议（thunder://）的网址。迅雷协议不是开放的，暂时没有太好的替代品，你不得不继续使用迅雷客户端。在这种情况下，你需要采用一些安全措施来防止迅雷客户端。本系列的下一篇博文会介绍各种隐私方面的防范措施。

如果你对上述列出的下载软件不中意，可以去参考维基百科的如下几个页面：
各种 BT 客户端比较
各种电驴客户端比较
各种文件分享工具比较
各种下载管理器比较

★媒体播放类

◇【不】靠谱的软件有哪些?

某些国内商业公司的媒体播放软件，至少包括“暴风影音、百度音乐（原“千千静听”）、QQ 音乐、射手影音”等。

◇首选替代品

（在首选替代品中列出的，全都是开源项目）

1. **VLC**
   这款诞生于2001年，也算是比较老牌的。在开源播放器中，它估计是最受欢迎的，截止俺写本文时，累计下载量超过13亿人次（参见"这个页面"）。另外，在所有的播放器中（包括开源和非开源），VLC 是支持操作系统最多的。所以俺把它排在第一位。
   操作系统支持：Windows、Linux、Mac OS X、Android、iOS、BSD、BeOS、Solaris、OS/2、DOS、Syllable、QNX
   格式支持：各种常见的视频、音频格式（你能想到的，应该都支持）
   特色：除了支持的操作系统最多，VLC 估计还是兼容性最好的。比如俺在一个干净的 WinThinPC 系统中测试，VLC 可以正常播放视频，而 SMPlayer 和 MPC-HC 都出现错误提示。
2. **MPlayer / SMPlayer**
   MPlayer 诞生于2000年，是一个视频播放的后端（命令行界面）。它可以跟几种不同的 GUI 前端搭配，比较有名的是前端是 SMPlayer。SMPlayer 诞生于2006年，是比较轻量级的，它的安装包（内置 MPlayer）比 VLC 小，Windows 下大概10多兆。
   操作系统支持：SMPlayer 支持 Windows、Linux、BSD
   格式支持：各种常见的视频、音频格式（你能想到的，应该都支持）
3. **Kodi（XBMC）**
   这款诞生于2002年。刚开始是打算做一个运行在 XBox 之上的播放器，所以原先名叫"XBMC"（XBox Media Center）。后来支持的平台越来越多了，XBox 反而不是重点了，于是改名叫 Kodi。
   操作系统支持：Windows、Linux、Mac OS X、Android、iOS、Apple TV OS、BSD
   格式支持：各种常见的视频、音频格式（你能想到的，应该都支持）
   特色：它采用基于 Python 的插件来扩展功能，甚至可以在它上面运行 Python 写的小游戏。
4. **MPC-HC**
   先来说一下 MPC——这是某个老外在2003年打造的轻量级播放器，界面完全模仿 Windows Media Player 6.4 版本。一开始是闭源的，后来完全开源了。可惜在2006年停止开发。然后另一个老外根据 MPC 的源代码，衍生出一个新的开源项目 MPC-HC 并继续开发至今。
   操作系统支持：Windows
   格式支持：各种常见的视频、音频格式（你能想到的，应该都支持）
   特色：界面风格是它的特色，继续保留原先 Windows Media Player 6.4版本的风格，适合喜欢复古的网友。

◇次选替代品

（在"次选替代品"中列出的，【不是】开源项目）

foobar2000
这款诞生于2002年，是免费的音乐播放软件，而且很小巧（目前的1.3.3版本只有3兆多）。
操作系统支持：Windows（XP 或更高）
格式支持：各种常见的音频格式，至少包括：MP1、MP2、MP3、MPC、AAC、WMA、Ogg Vorbis、FLAC/Ogg FLAC、ALAC、WavPack、WAV、AIFF、AU、SND、CD、Speex、Opus
特色：可以直接播放压缩包里面的音频文件（无需解压）。

◇备注

如果你对上述列出的媒体播放软件不中意，可以去看维基百科的“这个页面”，里面列出了几十种媒体播放软件的详细对比。

★邮件类（客户端）

◇【不】靠谱的软件有哪些?

某些国产的邮件客户端，比如 Foxmail。

◇首选替代品

（在首选替代品中列出的，全都是开源项目）

Mozilla Thunderbird（雷鸟）

这款是 Firefox 的同门兄弟——它俩都出自大名鼎鼎的 Mozilla 门下。Mozilla 是个非盈利组织，而不是商业公司。所以 Mozilla 旗下的软件是比较靠谱滴。

它估计是名气最大的开源邮件客户端，功能齐全——基本上你能想到的，它都有了。而且具有很强的可定制性。

操作系统支持：Windows、Linux、Mac OS X

加密支持：支持 S/MIME 标准，可以通过 Enigmail 插件进行公钥加密和签名（基于 OpenPGP）。

◇次选替代品

（在“次选替代品”中列出的，【不是】开源项目）

微软的 Outlook 客户端

（微软有两款 Outlook，此处泛指这两款）对于微软的 Outlook，大伙儿应该都比较熟悉，俺就不介绍了。

相比国产的邮件客户端，微软的 Outlook 可以避免被党国植入后门。但是微软的 Outlook 依然有可能被 NSA 植入后门。老实说，俺没觉得有啥理由可以让你“不选 Thunderbird 而去选 Outlook”。

★结尾

俺的见识很有限，所以本文介绍的替代品，肯定不全面。大伙儿如有其它补充，欢迎到本文留言。

[如何保护隐私9]：如何限制桌面软件的流氓行为?

文章目录

在本系列的前一篇，介绍了"流氓的桌面软件有哪些替代品"。今天主要介绍一下，如何对付那些"【缺乏替代品】的流氓软件"（比如：QQ、迅雷）。

本文会逐一介绍5种不同的方案，供大伙儿参考。在本文末尾会总结每种方案分别适用哪些情况。

★方案1：用流氓软件的【Web 版本】进行替代

◇概述

某些流氓软件的开发商，除了开发桌面客户端版本，有时候也会提供"Web 版本"。在这种情况下，咱们就可以拿"Web 版本"来取代桌面版本。

◇举例

比如：疼逊公司提供了一个 Web QQ。那么你就可以用 Web QQ 来替代桌面版的 QQ；

比如：很多网盘提供商同时提供网盘的"Web 操作界面"和"桌面客户端"。那么你就可以：只用 Web 界面操作网盘，不安装桌面客户端。

◇不同"Web 应用"的安全性等级

基于 Web 的应用软件，有好几种不同的类型，在隐私方面的安全性也各不相同。下面逐一介绍。在下面的介绍中，会提及"浏览器扩展"和"浏览器插件"。如果你搞不清楚这两者的差别，先看之前的"这篇博文"，其中有一段文字是"◇插件和扩展的区别"。

纯 Web

所谓的"纯 Web"，就是说只用到 HTML、JavaScript脚本、CSS 样式。除此之外，没有用到其它其它东西。

这种形式的 Web 应用是最安全的。对于这种形式，JS 脚本运行在浏览器内置的脚本沙箱中，它对文件系统的访问是受限的。换句话说，【无法】随意操作电脑本地的文件系统。

使用了"扩展/extension"技术

浏览器扩展能干的事情，比"纯 Web"更多。多出来的功能主要是：能够操作本地的文件系统。假设你安装了一个不怀好意的扩展，理论上它是可以读取你本地的某些敏感文件并通过 Web 方式（HTTP方式）发送出去。

提醒一下：并不是所有的浏览器都支持"扩展"机制。

使用了"插件/plugin"技术

"插件"比"扩展"牛逼的地方在于，插件是本地代码（也叫"原生代码"）。也就是说，普通客户端软件能干的事情，插件都能干。所以，插件的危险性更大，基本上等同于客户端软件。

比如前几天曝光的"支付宝偷偷监控网络流量"。那个监控你网络流量的进程，就是在安装支付宝控件之后，偷偷带进去的。而"支付宝控件"本质上就是"浏览器插件"。

综上所述，如果你企图用“Web 版本”来替代“桌面版本”，至少先要确保“Web 版本”没有使用“浏览器插件”技术。用到了“插件”的 Web 应用，其隐私威胁跟桌面客户端类似——也就没有必要替代了。

◇优点

主要优点是：操作简单，无需进行各种额外的配置，无需安装额外的桌面软件。

◇缺点

此招数的缺点有两个：

缺点1

使用了 Web 版本之后，虽然可以避免桌面客户端偷窥你的本地文件，但是你通过 Web 端产生的内容还是会被 Web 服务器收集到。

比如你用了 Web QQ 之后，虽然它无法偷窥你的本地文件，但是腾讯服务器依然可以偷窥你的聊天内容。

缺点2

通常，Web 版本的功能不如桌面版本的功能那么丰富。

比如“Web QQ”的功能比“桌面 QQ”的功能少很多。

★方案2：使用【网络防火墙】限制联网

◇概述

如果某个流氓软件的【基本功能】与网络无关，那么你可以采用网络防火墙限制该软件的网络行为。

◇举例

比如：你需要用国产的播放器来播放【本地的】视频文件和音乐文件。并且你既不需要“自动同步字幕”，也不需要“自动同步歌词”之类的功能。在这种情况下，该软件就完全没有理由访问网络。那么你就可以通过配置网络防火墙，让该软件无法联网。

◇网络防火墙的局限性

局限性1

某些流氓软件需要以管理员权限安装。在这种情况下，流氓软件也就获得了管理员权限。在“管理员权限”下，几乎可以进行任何软件层面操作。从理论上讲，可以利用管理员权限进行某些操作，来绕过网络防火墙的监控。

局限性2

某些比较狡猾的流氓软件，自己的主程序不会直接联网，而是通过创建一个临时的 exe 文件并启动该 exe（子进程），让子进程完成敏感的联网操作。以此来骗过网络防火墙。对这类软件，即使你禁止该软件的主程序联网，它还是能通过启动子进程的方式，把某些敏感的资料偷偷发送出去。

◇优点

用此招数，你可以直接使用流氓软件的桌面版本。相比“方案1”的优势在于——能保留软件的全部功能。

另外，此招数可以跟下面要介绍的“方案3”搭配组合。

◇缺点

此招数的缺点包括：

缺点1

如果该流氓软件的基本功能依赖于网络（比如：聊天工具、下载工具），就没法用这招。

缺点2

网络防火墙的限制有可能被绕过（参见上述提到的两个局限性）

★方案3：用流氓软件的【绿色版本】进行替代

◇概述

某些流氓软件可以制作成“绿色版本”（俗称“绿化”）。绿化之后的软件，无需安装，而且也无需管理员权限就可以运行。因此，你可以创建一个【低权限】的操作系统用户，以这个用户的身份来运行该流氓软件。关于 Windows 的“用户组、用户权限”等概念的更多介绍，请看之前的“这篇博文”。

另外，从 WinXP 开始，Windows 内置了“快速用户切换”（洋文叫：Fast User Switching）的功能。你可以利用此功能在不同的用户之前切换。不想使用“用户切换”的同学，也可以在同一个桌面上运行“隶属于不同用户的进程”（具体做法，上面那篇博文也有提到）。

◇需要结合“文件系统访问控制（ACL）”

和这个招数搭配的措施是：你要适当地配置文件系统的访问控制权限。比如你日常操作电脑的是“A用户”，用来运行流氓软件的低权限用户是“B用户”。那么你需要设置那些存放敏感文件的目录，让“B用户”【无法读写】这些目录。至于如何设置目录的 ACL，稍微 Google 一下就能查到教程。

提醒一下：

Windows 支持的文件系统主要有两类：NTFS 和 FAT 系列（FAT12、FAT16、FAT32）。“FAT 系列”【不支持】访问控制权限。

◇举例

比如“迅雷”是可以制作成绿色版本的。俺分别在它的5.8版本和7.2版本试验过。

◇优点

用此招数，你可以直接使用流氓软件的桌面版本。相比“方案1”的优势在于——能保留软件的全部功能。

另外，此招数可以跟“方案2”搭配组合。

◇缺点

此招数的缺点包括：

缺点1

如果某个流氓软件无法制作绿色版本，这招就失效了。

缺点2

操作相对繁琐（你需要先制作绿色版本，然后还需要创建专门的用户，最后还要设置文件系统的ACL）。

★方案4：使用【沙箱软件】进行隔离

◇概述

有一类安全工具叫做“沙箱”（有时也称“沙盒”）。沙箱软件可以创建出一个独立的运行环境，以实现【进程级别】的隔离。

所谓“进程级别的隔离”就是说：你可以在沙箱中运行某个进程，沙箱软件负责把该进程跟系统的其它部分隔离开来。即使该进程图谋不轨，想要进行某些危险操作，这些操作也被限制在沙箱之内。

所以，咱们可以利用“沙箱软件”来运行那些流氓软件。

◇举例

目前，【国外】比较有名的沙箱软件是 Sandboxie。如果你担心网页挂马，可以让浏览器运行在 Sandboxie 的沙箱之内。

◇沙箱的局限性

局限性1

沙箱软件依赖于操作系统本身的一些机制（比如系统的 hook 机制）。如果操作系统本身出现一些安全漏洞，可能会导致某些流氓软件突破沙箱的隔离。

另外，如果沙箱软件本身考虑不周到，也会导致隔离环境出现漏洞，可能被流氓软件用来突破沙箱。

局限性2

有些沙箱软件只能对磁盘操作进行隔离，但是【无法】对网络操作进行隔离（无法禁止流氓软件联网）。

如果你需要隔离网络操作，但是你用的沙箱又没有此功能，那么你可以考虑把此招数跟“方案2”搭配。

◇优点

用此招数，你可以直接使用流氓软件的桌面版本。相比“方案1”的优势在于——能保留软件的全部功能。

不管流氓软件是否能制作绿色版本，都可以使用此招数。所以此招数比“方案3”更灵活。

◇缺点

主要缺点有如下几个：

缺点1

某些软件（尤其是比较底层的软件）运行在沙箱中会出现兼容性问题。

缺点2

由于前面提到的“沙箱的局限性”，会导致隔离不彻底。

缺点3

多数沙箱软件通常无法跨系统（比如前面提到的 Sandboxie 是运行在 Windows 上，所以它只能用来隔离 Windows 软件）。

如果你对跨平台的需求不明显，这个缺点基本可以忽略。

★方案5：使用【操作系统虚拟机】进行隔离

◇概述

所谓的“操作系统虚拟机”（为了打字省力，以下简称“VM”），就是用专门的软件在你现有的操作系统内部再虚拟出若干个操作系统。在这种情况下，你原有的操作系统称为“物理系统”或“Host OS”；那些虚拟出来的系统称为“虚拟系统”或“Guest OS”。

有了 VM 之后，你就可以创建出一个或多个 Guest OS，在那上面运行流氓软件。由于 Guest OS 与 Host OS 是彻底隔离的，你完全不用担心 Guest OS 里面的进程偷窥 Host OS 的文件系统。

所谓的“彻底隔离”，俺补充说明一下：

因为 Guest OS 运行于 Host OS 之上，Host OS 是可以看到 Guest OS 里面的东西；反之则【不行】。如果有多个 Guest OS，互相之间看不到。

关于 VM 的更多介绍，请参见《扫盲操作系统虚拟机》系列教程。

◇举例

至少在10年前，俺就以虚拟系统（Guest OS）作为自己日常的使用系统。俺的 Host OS 除了装一个虚拟机软件，其它软件几乎都【不】装。需要用到的软件都装在 Guest OS 里面。而且俺的笔记本电脑里面装了 N 多的虚拟机，分别派不同的用处。比如：

一个专门用于“编程随想”这个身份
一个专门用于公司的办公环境（写文档、收发邮件）
一个专门用于公司的开发环境（留个开环境帮同事解决疑难编程问题）
一个专门用于业余时间的开发环境（业余时写代码）
一个专门用于业余时间的上网环境（业余时上网）
（还有很多，不一一列举）

搞这么多虚拟机的好处在于，把自己的敏感数据分散开——即便某个虚拟系统被入侵，其它虚拟系统【不受】影响。

而且 VM 软件都会提供【快照功能】——这个功能是 VM 的精华。比如每次刚装好系统，或者刚升级

好系统，俺都会做一个快照。用了几天之后，就回退到这个快照。就算这几天当中被恶意软件（病毒/木马/流氓软件）污染了系统，只要回退到快照，虚拟系统又纯洁如初 :)

前几天有读者问俺用的是哪个系统清理工具。俺回答说：有了 VM 软件，根本【无需】系统清理工具。为啥捏？因为每次回退快照，系统垃圾自动就没了（等同于一次彻底的清理）。

◇优点

VM 的优点很多，至少包括如下：

优点1

最好的兼容性。

前面提到的“方案3”和“方案4”，都可能会导致一些兼容性的问题。而 VM 完全没有。

优点2

最好的隔离性（关于这点，下面俺单独开了一个章节详述）

优点3

VM 的快照功能可以帮你完成很多额外的工作（参见俺刚才的举例）。

优点4

彻底实现“跨平台”。

比如你可以用 Windows 做 Host OS，然后在其中虚拟出一个 Linux 或 Mac 的 Guest OS；反之亦然。而沙箱软件做不到这点。

◇缺点

俺个人认为：对于那些安全要求高的网友，VM 的方案是最佳方案。此方案有如下两个不太明显的缺点：

缺点1

如果要使用，你的硬件配置不能太低。

如今电脑硬件的配置已经越来越好（动不动就是好几 GB 的内存，动不动就 4核/8核），这个缺点基本可以忽略。

缺点2

要花点时间学习如何使用 VM。

不过这个缺点也好办——俺在2年前（2012）就写了一个《扫盲操作系统虚拟机》的系列教程。不会用 VM 的同学可以先去看这个教程。

★为啥“OS虚拟机”的隔离性比“沙箱”更好?

俺在博客上写了不少操作系统虚拟机的博文。但是一直没有介绍过“沙箱软件”。曾经有几个读者在留言中问道：为啥从来没普及过沙箱？今天借着这个机会说一下。

前面俺提到了 VM 的各种优点。这里重点想说其中一点——**隔离性**。为啥捏？如果你对安全的要求比较高，“隔离性”是关键。而“VM”的隔离性远远好于“沙箱”。具体的分析如下：

◇关于隔离的“维度”

不论是 VM 还是沙箱，说白了都是为了创造出一个隔离的软件环境。这时候要考虑的维度有很多，至少包括：文件系统、内存、网络、外设（比如 USB 口）。

主流的 VM 软件可以支持上述【所有的】维度；而大部分沙箱软件只能做到文件系统和内存的隔离。有些沙箱虽然也支持网络隔离，但是其网络隔离的功能不如 VM 那么丰富（主流的 VM 软件通常支持至少3种以上的虚拟网卡模式）。

◇关于操作系统本身的漏洞

在使用“沙箱”的情况下，沙箱软件和流氓软件都运行在同一个操作系统中。如果这个操作系统出现某些安全漏洞，【可能会】导致流氓软件突破沙箱的隔离边界。

相对而言，使用 VM 的情况下，同时存在两个操作系统。VM 软件本身是运行在 Host OS 之上，而流氓软件运行在 Guest OS 之上。由于 VM 本身并不依赖 Guest OS。所以 Guest OS 出现安全漏洞，【不太可能】导致隔离边界被突破。

有些爱思考的同学会问了：如果 Host OS 出现安全漏洞，是否有可能导致隔离边界被突破。这种可能性是有的，但是操作的难度很大。为啥捏？因为流氓软件运行在 Guest OS 之内，它根本看不到 Host OS。换句话说，流氓软件根本就无法知道 Host OS 是啥类型的。连操作系统的类型都不知道，如何去利用操作系统的漏洞？

★总结

在本文的结尾，稍微做一下总结发言，说一下每种招数适用的情况。

◇方案1——用“Web 版本”替代

前提：对应的流氓软件提供 Web 版本

本方案无需额外的配置，也无需再安装其它桌面软件。所以适合那些非常菜鸟的网友，或者是那些非常懒的网友。另外，如果你只是想临时用一下某个流氓软件，也可以用这个招数。

◇方案2——用“网络防火墙”限制联网

前提：对应的流氓软件的【基本功能】与网络无关（比如：输入法、播放本地视频、播放本地音乐）

除了需要配置网络防火墙，无需额外安装其它软件。适合那些比较菜鸟的同学或者是比较懒的同学。

◇方案3——用“绿色版本”替代

前提：对应的流氓软件能够“绿化”

需要自己手工制作绿色版本，而且还需要配置额外的操作系统用户和文件系统的访问控制权限。所以此招数适合于具有一定折腾能力的网友。而且电脑硬件又很差，用不了虚拟机方案。

◇方案4——用“沙箱软件”隔离

前提：流氓软件需要能跟沙箱软件兼容。

如果你电脑硬件太差，用不了虚拟机。而且你又搞不定绿色版本（比如流氓软件不支持“绿化”，或者你不懂得如何“绿化”），那么你可以用此方案。

◇方案5——用“操作系统虚拟机”隔离

前提：无任何限制。

此方案堪称终极解决方案，尤其适合于对安全要求比较高的同学。此方案需要有一定的折腾能力（不需很高），而且电脑的硬件配置至少是中等水平。

◇所有这五种方案的局限性

不论你用的是哪个方案。如果你想使用某个流氓软件，并且**该流氓软件的基本功能是依赖于网络的**，那么，这款流氓软件就有可能收集到你的某些隐私并发送给厂商的服务器。即使你用了最安全的“虚拟机隔离”，顶多也只是降低了泄漏的信息量而已。

举例来说：你在虚拟机里面运行 QQ，可以阻止 QQ 偷偷扫盲你的 Host OS 的磁盘。但是，你的所有聊天内容，腾讯的服务器上都有记录。所以，**最安全的是：既不用流氓公司的软件，也不用流氓公司的网络服务。**

[如何保护隐私10]：移动设备的隐私问题

文章目录

★引子

这两天的劲爆新闻之一是《温州一公安分局采购木马病毒监控手机通话 @ 新浪》。俺只能感叹：朝廷及其走狗越来越丧心病狂了。再次应了那句话——俺向来是不惮以最坏的恶意，来揣测咱们的党国。

本来这个系列的下一篇要开始谈“网络服务的隐私问题”。但是有很多读者更关注手机的隐私问题，再加上这2天温州公安局的这个丑闻。于是俺改为——先谈移动设备的隐私，然后再谈网络服务的隐私。

和移动设备有关的隐私问题非常多，牵涉面很杂。俺打算分2篇来介绍。首先介绍“有啥危害”，然后介绍“如何防范”。

★哪些东西算“移动设备”?

按照俺一贯的风格，先界定清楚相关术语和范畴。

本文所说的“移动设备”，至少包括：手机（智能和非智能）、平板、可穿戴设备（比如：智能手表、智能腕带、智能眼镜）。

至于笔记本电脑或上网本电脑，不在此列。上网本虽然也很轻便，且容易移动，但是笔记本跟“手机/平板之类”，还是有本质的差异——进而导致了隐私防范方面的差异。

★【硬件】的隐私问题

咱们先从最底层的硬件说起。

◇麦克风

名词解释

这个俺就不用解释了吧？

隐私风险——基于麦克风窃听

麦克风可以收集到移动设备周围的声音，并且移动设备上的软件可以读取麦克风的数据。如果你的移动设备中了木马。木马当然可以用麦克风窃听设备周围的声音。

虽然笔记本电脑上也有麦克风，但是笔记本电脑不会随身携带，而手机通常是随身携带。所以，同样中了木马的情况下，手机的窃听风险远高于 PC。

隐私风险——跨系统盗取密码

有些手机为了提升安全性，采用了“双系统”的设计。其中一个系统专门用来进行敏感操作，另一个系统进行日常操作。从表面上看，这有点类似俺多次提到的“虚拟机隔离”。但是手机的特点决定了它无法做到像虚拟机那么彻底的隔离——比如这两个系统共享了摄像头、麦克风。如果A系统中了木马，木马可以利用摄像头、麦克风、陀螺仪获取信息，从而判断出你在B系统中输入的密码。

说得更具体一点：如果是基于【物理键盘】输入密码，可以利用麦克风截获按键音（根据不同物理按键微小的声音差异进行识别）。如果是基于【触摸屏】输入密码，可以根据摄像头或者陀螺仪，记录手机的微小位移，从而判断用户按了哪个数字/字母。（在信息安全领域，这种攻击手法称为“边信道攻击”）

引申阅读：

《科学家警告称：智能手机的摄像头和麦克风有暴露PIN码风险 @ 网易数码》

既然可以做到跨系统盗取密码，那么跨系统偷窥你的其它按键（比如输入了啥短信内容），当然也可以做到。

◇摄像头

名词解释

这个俺就不用解释了吧？

隐私风险——基于摄像头监控

既然手机中的软件可以操作摄像头进行“拍照”和“录像”。那么手机中的木马当然也可以干同样的事情。

很多人习惯把手机放在卧室里，万一放置的时候摄像头的角度比较好，万一你又中了木马。说不定你的裸照就流传出去了 :)

隐私风险——手机拍照，EXIF泄露的隐私

如今的智能手机和平板，基本上都内置摄像头了。很多手机用户经常会用手机拍照，然后分享到网上。

因为手机拍照后存储的照片文件，通常包含有很详细的 EXIF 信息，再次导致隐私泄露。所谓的

EXIF，通俗地说就是照片文件的元数据。有的手机拍的照片，照片文件中的 EXIF 非常详细，包含了手机的型号、拍摄时间、GPS 信息、等信息。

（顺便说一下：数码相机拍照后存储的照片文件，同样有 EXIF 信息）

引申阅读：

《美国国安局利用手机应用数据挖掘情报 @ 华尔街日报》

（WSJ 的这篇报道提到了 NSA 对手机照片中 EXIF 的数据挖掘）

隐私风险——跨系统盗取密码

此风险在前面介绍麦克风的时候，已经提及。此处不再罗嗦。

隐私风险——第一人称摄像会暴露摄像者的身份

类似 Google Glass 这类"头戴式设备"可以拍下我们肉眼所见的一切，并保存为视频文件（拍摄者不在视频之内）。

那么，别人有没有可能根据这段视频，判断出谁是拍摄者捏？

现在已经有一项技术，仅仅根据拍摄者头部的轻微晃动，来提取出某个独一无二的指纹。

引申阅读：

《第一人称摄像头可能暴露你的身份 @ 网易科技》

◇陀螺仪

名词解释

在移动设备中，"三轴陀螺仪"已经很普遍了。这玩意儿可以实时获取设备的运动状态（比如：运动方向、加速度）。并且这些运动状态的信息，是可以被设备中的软件读取的。

隐私风险——基于陀螺仪的窃听

可能很多人觉得陀螺仪不会有啥隐私问题。但是你错了！以目前的技术水平，**陀螺仪可以实现【窃听】**。

这可不是俺耸人听闻，请看大名鼎鼎的《Wired》的报道（链接在"这里"）。看不懂洋文的同学，请看这里的中文节译。此技术是半年前（2014）新鲜出炉的，目前还不成熟。但今后就不好说了。

隐私风险——泄漏你的生活规律

除了前面提到的窃听问题，还有其它一些隐私风险。

由于手机通常都是随身携带的。如果某个软件长时间收集你手机中的陀螺仪数据，就可以大致了解你的日常生活规律——

比如你是驾车上下班还是坐公交车/地铁（私家车/公交车/地铁，其【加速度】的规律是不同的）

比如你周末是宅在家里还是出门逛街。

比如你是否有体育运动的习惯。

……

隐私风险——跨系统盗取密码

此风险在前面介绍麦克风的时候，已经提及。此处不再罗嗦。

◇GPS

名词解释

GPS 是"全球定位系统"的缩写。这玩意儿如今也比较普及了，大伙儿应该都听说过。

隐私风险——泄露你的地理位置

GPS 的主要问题在于泄露了你的地理位置。在这个方面，GPS 暴露的信息远远高于陀螺仪。因为陀螺仪没法准确定位到经纬度，而 GPS 可以。

因此，如果某个流氓软件（或木马）收集了你的 GPS 信息，就可以非常清楚你的行踪。

隐私风险——泄漏你的生活规律

此风险在前面介绍陀螺仪的时候，已经提及。此处不再罗嗦。

◇其它硬件问题

还有其它一些硬件问题，放到后面的章节介绍。比如 wifi 网卡的问题，在“wifi无线网络”中介绍。

★【手机移动网络】（2G、3G、4G）的隐私问题

为了打字省力，在本章节中，凡是提及“手机”一词，均包含“带有移动电话功能的平板”。
不具有移动电话功能的“平板/可穿戴设备”，不在本章节的讨论之列。

◇手机的唯一标识（IMEI、IMSI、ICCID、MSISDN）

名词解释

大伙儿看到这几个英文缩写容易昏菜，大致解释一下：

IMEI（International Mobile Equipment Identity）
俗称“手机串号”，用来唯一标识某一部手机。共有15位数字。相关维基词条在“这里”。

IMSI（International Mobile Subscriber Identity）
在移动电话网中唯一标识某个用户。共有15位数字。咱们日常用的手机号，没法解决国际漫游问题。IMSI 可以用来解决国际漫游。相关维基词条在“这里”。

ICCID（ICC IDentity）
唯一标识某个 SIM 卡。相关维基词条在“这里”。

MSISDN（Mobile Subscriber ISDN）
这个就是咱们平常所说的手机号（对于大陆而言，就是“86”加上“11位数字”）。相关维基词条在“这里”

隐私风险——身份定位

理论上讲，智能手机软件可以通过各种 API 获取到上述这些标识符。举例：参见 Android 的这个 API。

在本系列的第5篇《扫盲“浏览器指纹”》中，俺已经解释了“信息量”的概念。如果你还记得的话，你会发现上述这几个标识符，任何一个的信息量都【相当大】，大到足以定位某个具体的人。

隐私举例

假如某个手机软件收集了你的 IMEI 和 手机号码。后来你换了一个手机号，但是手机没换，那么该软件再次收集这两个信息之后，就可以判断出：（有很大可能）这两个手机号码其实是同一个人。

◇基站

名词解释

凡是能工作的移动电话，不外乎都要跟“移动基站”打交道。所谓的基站，是移动运营商架设的通讯设施。你的手机需要先跟基站建立无线连接，才能进入到运营商的无线移动网络中。

和基站相关的隐私问题，主要包括如下2方面：

\1. 手机上的基站信息
手机跟基站建立通讯连接之后，手机上会存储当前基站的信息。手机软件可以读取这些信息。如果某个手机同时连接的基站达到三个，就可以用几何定位。
通过基站进行定位，精度通常不如 GPS 那么高。但也已经能够获得某些隐私信息。
\2. 基站上的手机信息

手机跟基站建立通讯连接之后，基站自然就获得了手机的信息。通过这个信息，运营商就可以知道你这部手机处于哪个位置。

隐私风险——泄露你的地理位置

此风险在前面介绍 GPS 的时候，已经提及。此处不再罗嗦。

隐私风险——泄漏你的生活规律

此风险在前面介绍 GPS 的时候，已经提及。此处不再罗嗦。

◇伪基站

名词解释

某些图谋不轨的人可以架设一个冒牌的基站，然后用来群发垃圾短信。这种就称为“伪基站”。维基词条参见“这里”。

隐私风险——泄漏手机标识

如果你的手机跟“伪基站”建立连接，那么“伪基站”就可以获取你手机的手机号（MSISDN）以及手机串号（IMEI）。

◇政府对运营商的监控

名词解释

这个不用多解释，大伙儿只需看看“棱镜门丑闻”。在这方面，咱们的朝廷其实比山姆大叔更糟糕——具体参见俺在《中美政府信息监控的差异——“棱镜门”丑闻随想》一文中的点评。

隐私风险——政府对移动运营商的监控

4年前，俺写了《如何隐藏你的踪迹，避免跨省追捕》系列的其中一篇《通讯工具的防范》。其中就已经提到政府对移动运营商的监控。在天朝，所有运营商（移动、电信、联通）的移动数据（语音通话、短信、彩信）都在朝廷的监控之下。

案例——“茉莉花事件”时期对移动网络的监控

2011年发生的“茉莉花事件”，大伙儿应该还记得吧？当时朝廷如临大敌。因此，也就对移动网络进行严密控制。那个时期，如果你在【群发】的短信中包含“茉莉花”3个字，估计当天就会有党国的走狗去敲你家门。

案例——美国的棱镜门丑闻

棱镜门丑闻，大伙儿应该都很熟悉了。从斯诺登曝光的材料看，美国的 NSA 可以监控其它国家（比如：巴哈马、菲律宾、墨西哥）的移动网络。这些国家的运营商并不在美国政府的管辖范围之下，NSA 是如何做到的捏？据说是 NSA 在这些运营商的网络中设置了后门。

◇移动通讯协议的破解

名词解释

通俗地说，“移动通讯协议”就是移动网络中用来传输数据（语音、文本）的协议。你的语音通话、短信、彩信、移动上网，底层都是基于“移动通讯协议”来传输的。

一旦攻击者能够破解某种移动通讯协议，就意味着攻击者可以拿到你的通讯内容（比如：语音通话、短信、移动上网数据）

隐私风险——对 2G 加密协议的破解

虽然移动通讯协议都是经过很多专家（当然也包括砖家）精心打造的，但是依然会有安全漏洞。尤其是 GSM 协议，诞生已经超过20年。当时设计的时候，并未考虑到如今这么强的运算能力（暴力破解）。

引申阅读：
《黑客宣布GSM手机加密算法已被破解并公开 @ 网易科技》
《安全研究人员谈破解与监听GSM手机 @ CNET》
《Open-Source Effort to Hack GSM @ IEEE》

隐私风险——对 3G 加密协议的破解

那么，比 GSM 晚很多年诞生的 3G 协议是否很严密了呢？好像也未必。

引申阅读：
《工程师2小时破解3G网络128位通讯加密方法 @ ZDNet》
《A Practical-Time Attack on the A5/3 Cryptosystem Used in Third Generation GSM Telephony @ IACR》

★【无线局域网】（Wi-Fi）的隐私问题

聊完了移动网络，再来聊聊 wifi 网络。

◇无线网卡的 MAC 地址

名词解释

所谓的“MAC 地址”，你可以通俗地理解为网卡的硬件唯一标识。更详细的介绍可以参见维基词条（在“这里”）

隐私风险——泄漏你的行踪

当你的手机了开启 wifi 功能，手机会时不时地扫描周围的 wifi 信号，看看是否有可用的 wifi 网络连接。在扫描信号的过程中，手机会发出一些 wifi 信号。这些信号中包含了你的手机 wifi 网卡的 MAC 地址。因此，有心人就可以利用某些专门的设备，收集这些手机的网卡的 MAC 地址。由于 MAC 地址具有非常好的唯一性，因此也就具有很高的信息量，足以定位到唯一的个人。

下面这篇引申阅读的文章，就提到了纽约的商家是如何利用这点，摸清顾客的行踪及偏好（很详细哦）。

引申阅读：
《小心，你的手机泄露了你的秘密 @ 华尔街日报》

◇公共场所的热点

名词解释

顾名思义，就不解释了。

隐私风险——你的网络流量可能被监控（被嗅探）或篡改

很多同学喜欢使用公共场所的 wifi 热点。这里面有潜在的隐私风险。

首先，你并不知道架设这个热点的人是不是一个攻击者。比如你在星巴克里面测试到的 wifi 热点，既可能是星巴克提供的，也可能是某个攻击者提供的（然后伪装成星巴克的热点）。

其次，至于那种来历不明的热点，就更可疑啦。

万一你使用的热点是某个攻击者架设的，那么这个攻击者就可以嗅探到你（基于该热点的）所有的上网流量。万一你访问的网站是走 HTTP 协议而不是 HTTPS 协议，那么你就形同裸奔了（HTTP 是明文的，HTTPS 才是加密的）。

隐私举例

假设你在公共场所用了某个来历不明的热点上网，并且访问了某个国内邮箱（大部分国内邮箱，都【不是】全程 HTTPS 加密）。而这个热点是一个攻击者架设的。那么攻击者就可以拿到你邮箱中的所有内容。

更有甚者，攻击者可以直接在邮箱的网页上挂马（因为 HTTP 是明文的，可以直接修改页面内容，植入恶意脚本）。

◇基于 Wi-Fi 热点的定位

名词解释

这种定位的原理，有点类似于"基站定位"。

如果移动设备开启了 wifi，设备上的软件可以收集周边 wifi 热点的信号强度，然后上传到该软件的服务端。如果服务端已经存储了大量的热点信息，就可以通过算法比对，定位出该设备的位置。

隐私风险——泄露你的地理位置

此风险在前面介绍 GPS 的时候，已经提及。此处不再罗嗦。

隐私风险——泄漏你的生活规律

此风险在前面介绍 GPS 的时候，已经提及。此处不再罗嗦。

◇对 Wi-Fi 协议的破解

名词解释

wifi 因为是无线的，意味它没有明确的物理边界，这就让攻击者有机可乘。比如你在公司里面架设的 wifi，如果 wifi 信号覆盖到公司之外（这种情况很常见），那么攻击者无需进入公司，就可以进行破解。

隐私风险——你的网络流量可能被监控（被嗅探）或篡改

此风险与"使用未知的公共热点"类似，俺就不再浪费口水了。直接转一篇帖子，给大伙儿瞧一瞧。

引申阅读：

《黑客讲故事：攻下隔壁女生路由器后，我都做了些什么 @ 知乎日报》

★【固件】（Firmware）的隐私问题

名词解释

所谓的"固件"，港台叫"韧体"，洋文叫做"Firmware"。维基词条在"这里"。你可以通俗地理解为"手机操作系统"。

隐私风险——上述的各种风险

既然是固件，自然可以访问到手机的各种硬件（如前所述的：摄像头、麦克风、陀螺仪、wifi网卡……）。所以，**如果固件不靠谱，前面提到的【所有风险】，都会存在。**

那么，啥情况会导致"固件不靠谱"捏？大致有如下几种情况：

\1. 手机厂商默认安装的固件就有问题——比如下面列举的"小米丑闻"和"酷派丑闻"。

\2. 你自己刷了第三方的固件，而这个固件本身就不靠谱——比如内置了后门。

案例——小米手机的丑闻

去年（2013）小米手机曝光了严重的隐私丑闻。经过海外的安全公司 F-Secure 进行专门的测试，小米手机不但会偷偷收集本机的"IMEI 和手机号"，更夸张的是——会收集每一个你拨打的号码或发送短信的号码。

引申阅读：

《小米手機偷傳資料到北京？ iThome找資安專家實測：有 @ iThome》

《小米窃取用户隐私被央视曝光，在台湾已道歉 @ 网易科技》

（如果你以为只有小米会这么干，那你就"图样图森破"了，请看其它几家公司的丑闻）

案例——其它国产手机的丑闻

《酷派多款手机存后门——可自动安装应用 @ 腾讯科技》
《酷派手机被曝存"后门" @ 人民网》
《中兴证实其一款在美销售手机存后门漏洞 @ 网易科技》

前面说的是国产手机。国外手机也未必是可信滴。别忘了"棱镜门"!

国产手机和国外手机的差别在于——国产手机的后门可能会被朝廷的六扇门控制，而国外手机（主要是美国手机）的后门可能会被 NSA（美国国安局）控制。由于 NSA 对咱们天朝的屁民【没有】司法管辖权，而天朝的六扇门分分钟就可以给你扣上"煽动颠覆"的罪名。所以，就算双方都有后门，国外手机还是好过国产手机。

★【应用软件】的隐私问题

移动设备上的应用软件，按照其运作机制可以分为两类：

\1. 该软件的特性和移动设备密切相关，因此只存在于移动设备上（比如说"计步器"）。
\2. 该软件的特性和移动设备关系不大，既存在于移动设备，也存在于 PC 上（比如说"邮件客户端"）。

对于第2类，在本系列前面3篇博文，俺已经花了很多口水介绍【桌面软件】的隐私问题以及防范手段——这就把第2类基本上覆盖到了。

至于第1类，其隐私风险与固件类似。也就是说，**只用某个应用软件不靠谱，前面提到的【所有风险】，都会存在。**

比如微信，装机量如此巨大，朝廷如果要求疼逊在微信中植入后门，疼逊敢拒绝吗？如果微信这个应用收集你的各种信息，然后加密发送到疼逊自己的服务器上，你能察觉到吗？

★与【物理安全】相关的隐私问题

◇失窃

手机/平板容易被偷，这应该是众所周知的了。一旦失窃，那上面的数据（通通都跟隐私有关）就拱手让人了。

相比而言，笔记本电脑的失窃率就远远小于"手机/平板"；至于台式机，失窃率就更低了（哪怕遭遇入室盗窃，小偷都懒得去扛台式机）。

从这个角度也可以看出，移动设备【额外的】隐私风险。

◇不恰当的报废处理

如今多数人至少都用过不止一个手机。淘汰下来的旧手机该咋办捏？

有些人就当废品卖了——愚蠢大大滴！旧手机能值几个钱？但是你牺牲了自己的隐私。因为旧手机上存储了很多跟你本人密切相关的信息。即使你把手机 reset（重置为出厂设置），这些数据也【不一定】能彻底清除。

引申阅读：

《清空的智能手机 数据仍能恢复 @ Solidot》

★其它的隐私风险

◇闭源的问题

（截止俺写此文时）占据手机市场份额的是 Android（约3/4）和 iOS（约1/4）。iOS 不用说，肯定是闭源的。Android 表面上看是开源的，其实捏，不完全是。

Android 系统包括两部分：AOSP（Android Open Source Project）和 GMS（Google Mobile Services）。GMS 是【不】开源的。而且自从 Android 占据市场主导地位之后，Google 逐渐把 AOSP 中的模块转移到 GMS 中（参见 36氪 的这篇报道）。

关于“开源/闭源”导致的不同隐私风险，本系列头一篇《如何选择软件和服务？》就已经分析过了。此处不再罗嗦。

好在 Mozilla 也开始介入手机市场，推出了 Firefox OS（应该是全开源的）。Mozilla 作为非盈利组织，在隐私保护方面会好于 Google（商业公司）。所以 Firefox OS 会比 Android 保险一些。

可惜 Firefox OS 目前的市场份额还太小，也不晓得这个系统将来能否做大 :(

◇缺乏“操作系统虚拟机”

（如果你从来没听说这玩意儿，先去看《扫盲操作系统虚拟机》）

PC 上有很多成熟的“操作系统虚拟机”（比如 VirtualBox、VMware）。但是手机上没有这么好的（虽然有些智能手机可以用 QEMU，但整体功能不如桌面的虚拟化软件）。

“操作系统虚拟机”可以帮你干很多“隐私防范”的工作。缺了它，手机上的隐私保护就更难搞了。

关于“操作系统虚拟机”在信息安全方面的用途，参见俺的另2个系列《如何隐藏你的踪迹，避免跨省追捕》和《如何防止黑客入侵》。

◇缺乏“优秀的磁盘加密”

手机上确实有一些磁盘加密和文件加密的工具，并且安卓手机和苹果手机也都逐步支持全盘加密。但总体而言，“手机的磁盘加密”在成熟度还是明显不如 PC 的磁盘加密。

比如说：PC 上的 TrueCrypt/VeraCrypt 同时支持“KeyFile 认证”和“Plausible Deniability”（隐藏卷），并且这2个特性也可以用在【加密系统分区】。作为对比，手机上的加密工具，貌似没有哪个软件能同时做到上述这些。

★结尾

本文开头俺说了——和移动设备有关的隐私问题非常多，牵涉面很杂——因此，本文恐怕有遗漏之处。欢迎列位看官补充 :)

[如何防止黑客入侵0]：概述

先声明一下：

其实本系列更恰当的名称应该是《如何防止骇客入侵》。因为"黑客"与"骇客"是截然不同的两个概念。只是考虑到本系列是面向技术菜鸟，而他们大都只听说过"黑客"。所以才用了这个标题。

上回俺在"如何隐藏踪迹，避免跨省追捕"系列中，谈到了操作系统的安全防范，其中有一个很重要的环节，就是防止自己的电脑被黑客入侵。当时由于篇幅有限，没有展开讨论。所以，今天就把这个话题补上。

防范黑客入侵，会涉及到不同方面的话题。因此，俺打算每个话题写一个帖子。有些话题，除了涉及技术领域，还会涉及非技术领域（也就是社会工程学）。对"**社会工程学**"不太熟悉的同学，可以先翻墙看"这里"，扫盲一下。

另外，本系列面向不太熟悉计算机安全的网友，某些技术细节会讲得比较啰嗦。懂行的网友请自行略过，以免浪费宝贵的时间。

为了方便阅读，把本系列帖子的目录整理如下（需翻墙）：

[如何防止黑客入侵1]：避免使用高权限用户

文章目录

为啥俺把这个话题列在头一条？——因为这是个非常普遍、且远远没有得到重视的问题。根据俺的经验，如果你能够养成好习惯，【不】使用高权限用户（尤其是管理员）进行日常操作，就可以大大降低被黑的概率。下面，俺就来具体介绍一下。

★基本概念扫盲

考虑到本文是面向外行人士，先进行一下名词解释。

◇用户权限

所谓“用户权限”，通俗地说，就是某个用户的权力有多大。权力越大，能干的事情越多。

◇用户组

用户组，顾名思义，就是一组用户的集合。

在主流的操作系统中，“用户权限”通常是和“用户组”挂钩滴。针对不同的用户组，分配了不同的权限。

为了让用户省事儿，Windows 系统内置了若干用户组（比如：Users、Power Users、Guests、等）。这些内置的用户组，事先已经预定义好若干用户权限。

◇高权限用户

本文提及的【高权限用户】，主要是指 Windows 系统中 Administrators 组的用户或 Linux/Unix 系统中 root 组的用户。

另外，顺便消除一个误解。很多菜鸟以为：Windows 系统中，只有用户名为“Administrator”的用户才具有管理员权限。其实捏，任何一个用户，即使用户名不叫“Administrator”，只要是属于“Administrators 组”，也同样具有管理员权限。

★反面教材

菜鸟的例子就不提了，光说说俺接触过的很多程序员吧。这帮家伙在使用 Linux/Unix 系统进行开发时，都晓得应该用普通用户的帐号进行操作；当需要做某些高级权限的操作，再切换到管理员帐号（root 帐号）。但即便是这些开发人员，在自己的 Windows 系统中，却喜欢用管理员（Administrator）进行日常操作，实在是很讽刺。

如果连 IT 专业的开发人员都这样，那不太懂技术的菜鸟，就可想而知了。

★危害性

如果你平时总是用管理员权限登录到系统并进行日常工作，那就意味着你所运行的每一个程序，同时也具有了管理员权限。要知道，管理员权限的权力是非常大滴——几乎可以干任何事情。

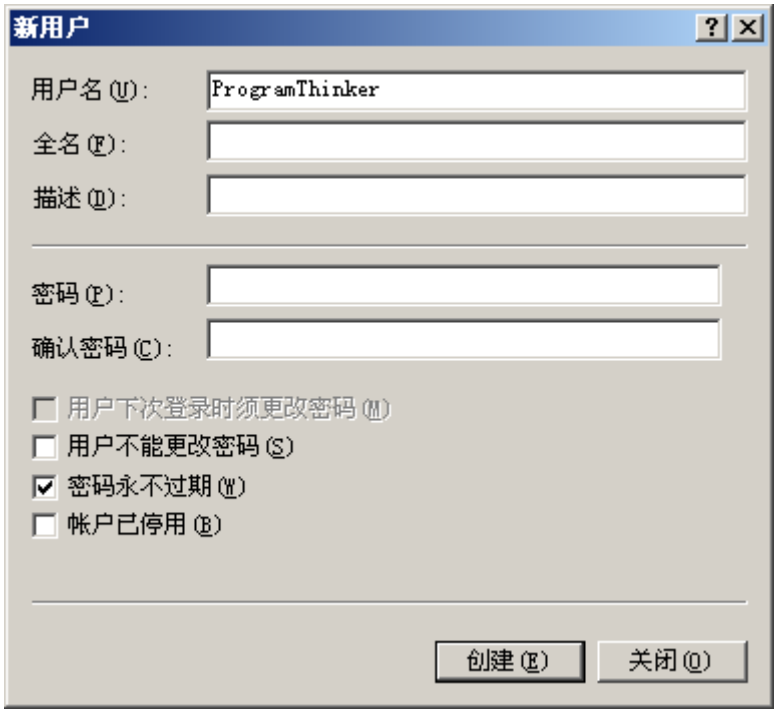
假设你有上述【坏】习惯。某天，你从网上下载了一个软件，且软件已经感染了病毒。那么，当你运行这个软件时，这个病毒就会被激活。更要命的是，它也同样具有【管理员权限】。也就是说，病毒获得了与杀毒软件平起平坐的地位。假如这个病毒的作者水平再高一些，甚至可以骗过杀毒软件或者直接把杀毒软件干掉。

除了病毒，木马也是一样。假设你上网的时候，一不留神访问了某个挂马的网站。一旦木马被激活，也同样是以【管理员权限】运行，危害同样也巨大。

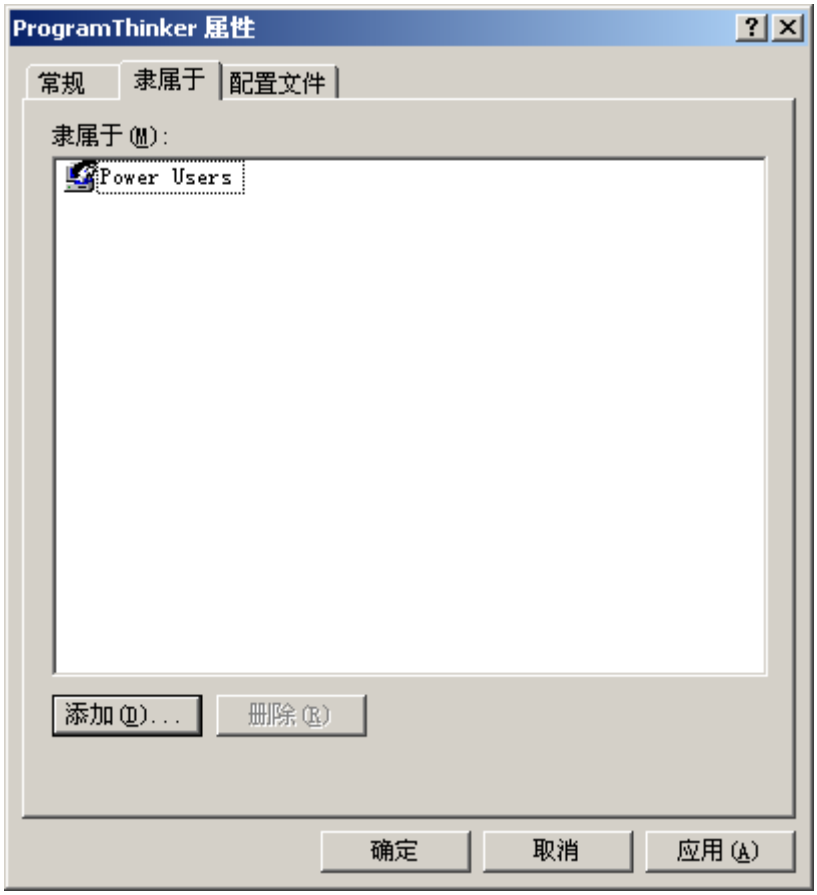
★你该如何做?

考虑到 Windows 系统的用户占绝大多数，俺就光拿 Windows 系统来说事儿。希望 Linux 及 Mac OS 的 fans 不要见怪哦。

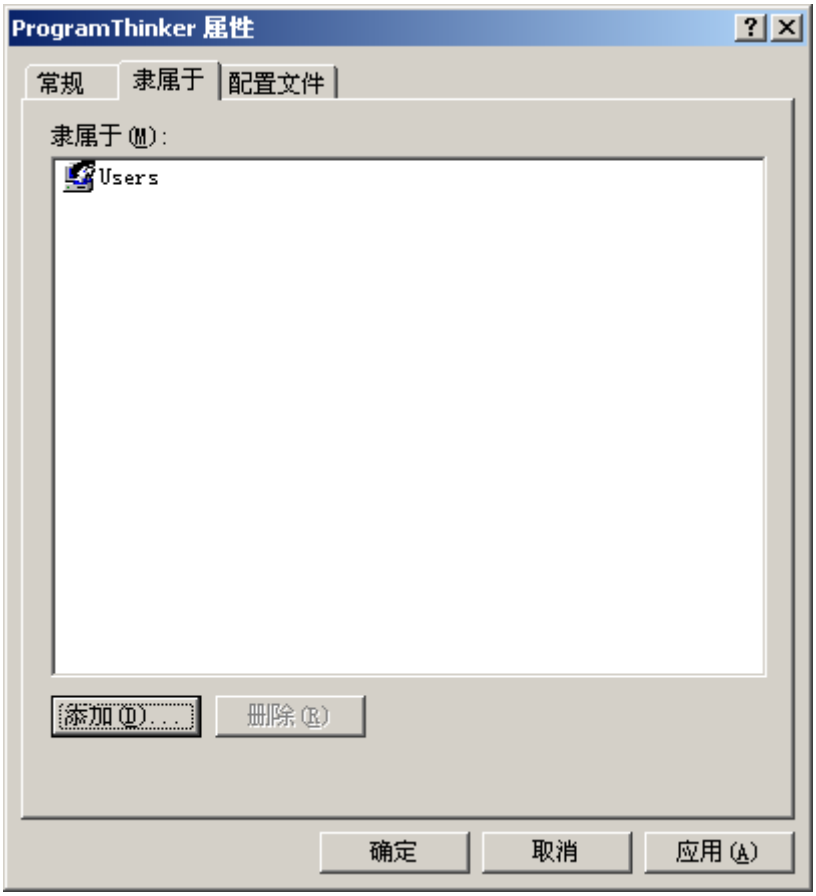
为了尽量少用高权限用户。你最好刚装好系统之后，单独创建一个非管理员用户。



你可以让该用户仅仅属于“Power Users 组”，如下图：



如果想更安全的话，可以只加入“Users 组”，今后，就主要通过这个用户进行日常的操作。如下图：



考虑到有些同学不了解这两个用户组，在权限方面与管理员有啥区别。俺简单列举一下。

◇“Power Users组”与“Administrators 组”的差别

相对于“Administrators 组”，“Power Users 组”缺少了如下几项权限（俺只列主要的）：
\1. 不能添加、删除、禁用系统中的其它用户。
\2. 不能修改其它用户的属性（包括口令、所属的用户组、等）
\3. 不能安装/卸载硬件驱动程序。
\4. 不能安装/卸载某些应用软件。
\5. 不能查看系统的安全日志。

◇“Users 组”与“Administrators 组”的差别

“Users 组”的权限比“Power Users 组”更小。除了“Power Users 组”做不到的事情，“Users 组”还【缺少】如下权限（俺只列主要的）：
\1. 不能修改系统时间。
\2. 不能修改某些系统目录（包括：系统盘的 `\WINDOWS` 目录、系统盘的 `\WINDOWS\SYSTEM32` 目录、系统盘的 `\Program Files` 目录）。
\3. 不能启动/停止某些系统服务。
\4. 不能修改注册表 `HKEY_LOCAL_MECHINE` 下的所有键值。

从上述对照，明显可知，“Users 组”的权限更小，使用起来更安全。比如说，即使你运行了一个带毒的程序，由于病毒和你一样，也仅有“Users 组”的权限。所以病毒也就无法修改/破坏重要的系统目录，掀不起太大风浪。

★可能的麻烦

通常来说，越安全的措施，往往也意味着越麻烦。但是这些麻烦，都有相应的解决之道。

◇切换用户的麻烦

当你以普通用户身份登录后，可能由于某些原因，需要用管理员用户干点事情。但是你（可能是开了很多程序）又不想把当前用户注销。

俺的建议是：使用【快速用户切换】（洋文叫：Fast User Switching）功能来切换用户。此功能从 Windows XP 开始提供。简单地说，就是可以让几个不同的用户同时登录同一个系统，平滑地切换。有了此功能，这个麻烦就不明显啦。

如果你非常不幸，还在使用比较古老的 Windows 2000 系统；或者你使用的是 Windows 的服务版本（比如 Windows Server 2003）。在这些版本的 Windows 系统中，默认是没有“快速用户切换”功能滴。这可咋办捏？

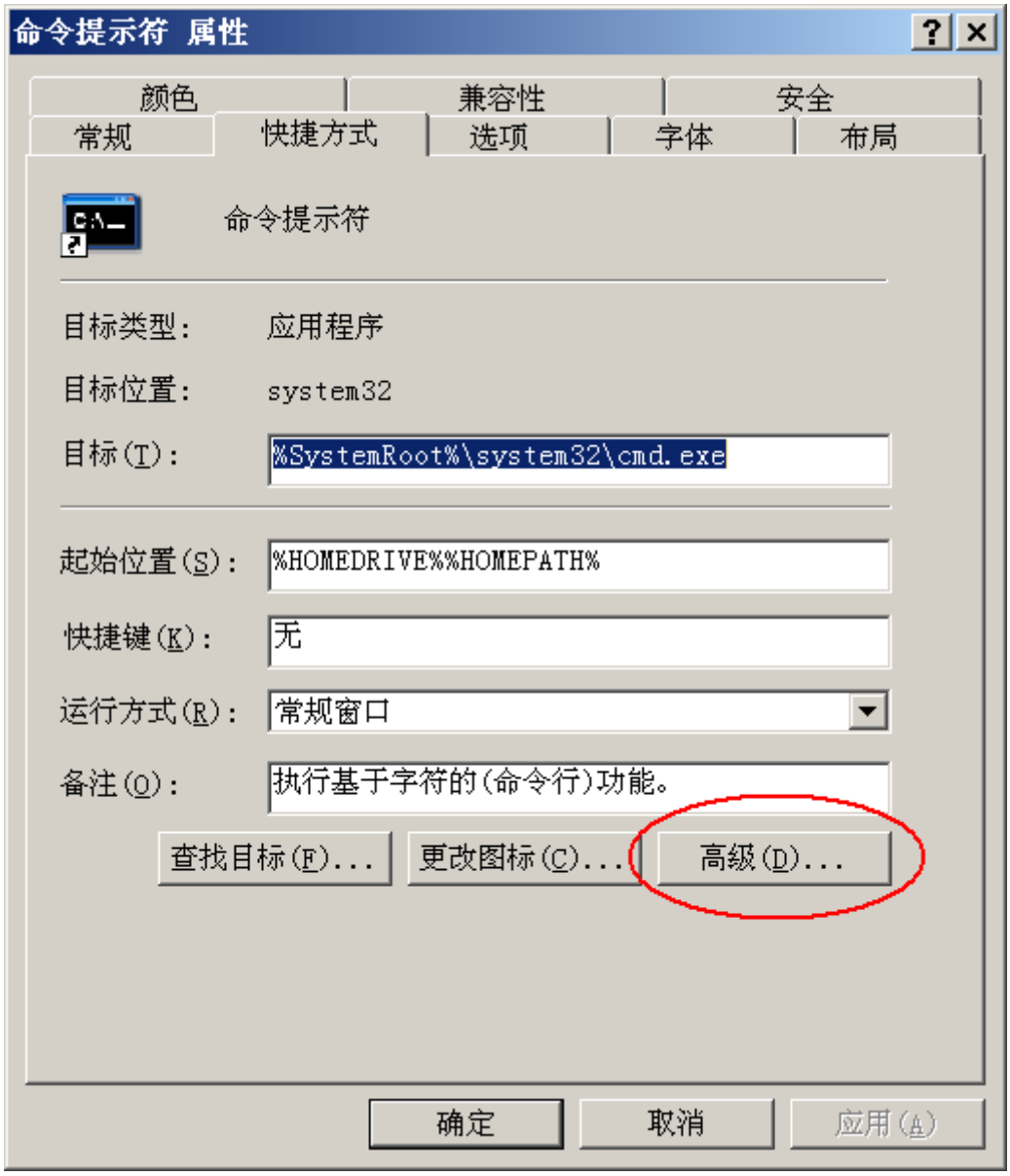
俺的建议是：

可以在【不】注销当前用户的情况下，以另一个用户（比如管理员）启动某个需要更高权限的程序。为了说清楚，举例如下：

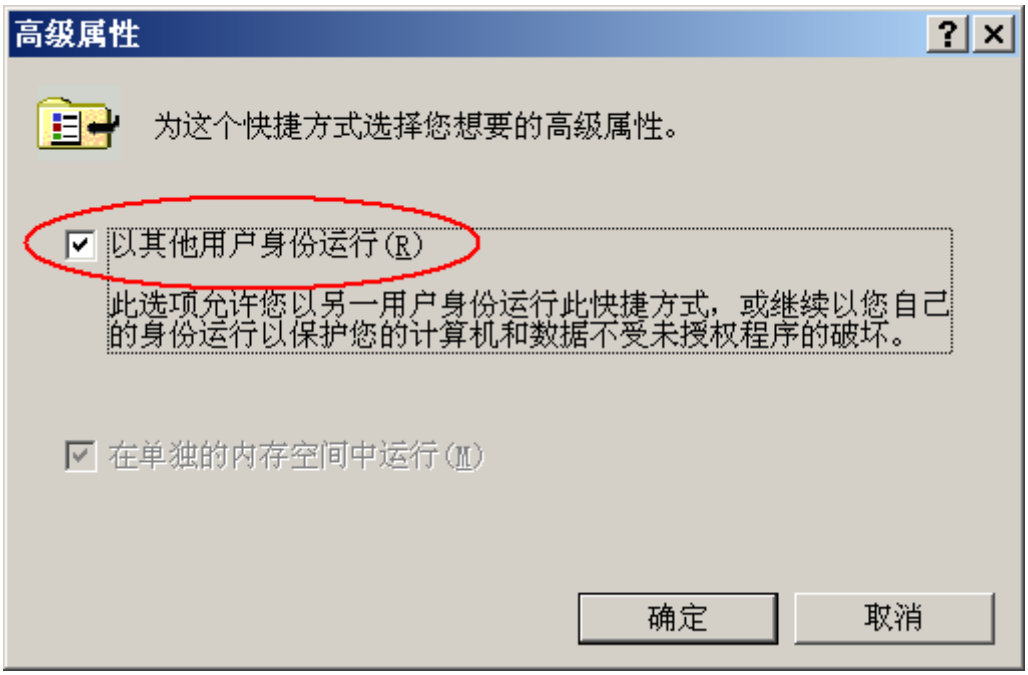
假设你当前处于一个普通用户的环境，但是想另外启动一个具有管理员权限的程序，比如说命令行程序（cmd.exe）。

\1. 你先创建一个指向 `cmd.exe` 的快捷方式。（该咋样创建快捷方式，就不用俺来教了吧？）

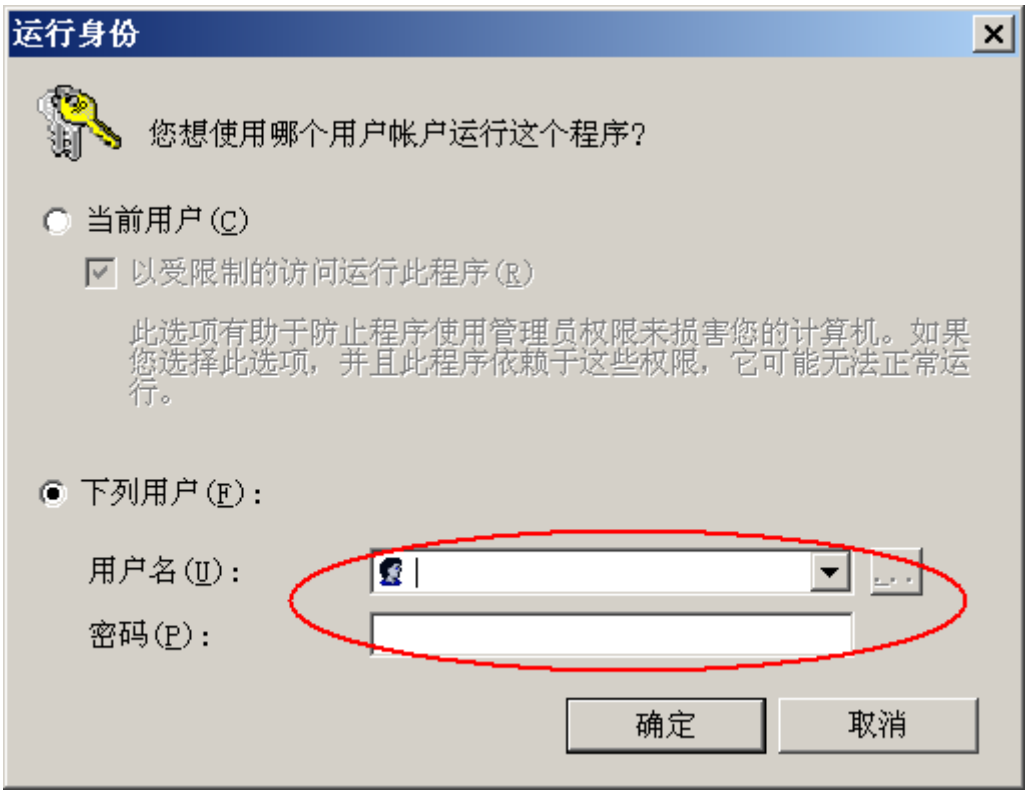
\2. 用鼠标选中该快捷方式，在快捷菜单（右键菜单）中，选择“属性”菜单项。出现如下对话框。



\3. 在该对话框中，点“高级”按钮。出现如下对话框。把“以其他用户身份运行”选项打勾，即可。至此，快捷方式创建完毕。



\4. 以后，如果你想在普通用户环境中，以管理员身份执行命令行，只需点击该快捷方式，就会弹出如下对话框。然后在该对话框中输入管理员的用户名和口令，就能以管理员的身份，把该命令行启动起来。



◇安装软件/驱动的麻烦

安装驱动总是需要【管理员权限】才行；另外，很多软件（比如 MS Office）在安装时，也要求用【管理员权限】进行。

俺的建议是：

在刚装好系统之后，先用管理员用户把上述这些软件/驱动程序都搞好。然后，就无需再用管理员用

户了。毕竟你经常使用的软件相对固定，不可能三天两头安装软件或驱动（除非你是软/硬件发烧友）。即便偶尔需要重新装个软件或驱动，也可以用上述介绍的方式，【临时】切换到管理员权限。

◇修改系统时间的麻烦

如果你平时用的是“Users 组”而不是“Power Users 组”，那你连修改系统时间的权限也没有。
俺的建议是：
启用 Windows 系统自带的时间同步服务，让它帮你自动同步系统时间。

除了上述这几点，如果还有谁碰到其它的麻烦，也欢迎来信跟俺交流。俺会补充到本文中。本系列的下一个帖子，将会介绍攻击者是如何搞定你的口令/密码。

[如何防止黑客入侵2]：攻击者如何搞定你的口令/密码？

文章目录

在上一个帖子，俺强调了高权限用户的潜在风险。接下来，咱要介绍一下，和口令相关的安全话题。毕竟在大伙儿的日常生活中，口令的使用是必不可少滴。
考虑到和口令相关的内容较多，俺分两部分来说：今天首先揭露攻击者的种种伎俩；下一帖再详述应对的措施。

★使用密码的场合（密码的类型）

为了便于后面的叙述，俺有必要先总结一下，使用口令的几种场合。
针对这几种不同的场合，攻击者会采取不同的攻击手法；因此，大伙儿也要采取针对性的防范手法。

◇操作系统用户的口令

这种场合应该好理解。目前主流的操作系统都具有口令验证的用户登录机制。

◇各种网络应用的口令

随着网络（尤其是 Web）的普及，这种场合越来越多。比如：电子邮件（Email）、即时通讯（IM）、论坛（BBS）、上网炒股、等，都需要有用户口令认证。

◇各种本地应用程序的口令

此种场合可能不如网络应用的口令那么常见。比如：用口令加密的压缩文件、用口令加密的 Office 文档、PGP 密钥的口令、Outlook 设置的启动口令……

◇其它

除了上述3种类型，其它那些比较少见、杂七杂八的，统统归为其它。比如：BIOS 的开机口令。

★攻击者如何通过技术手段搞定的你的密码？

前面列举了密码的不同使用场合。接着咱要介绍一下：攻击者会利用哪些技术手段，攻破你的密码。

◇木马盗取

如果你的电脑已经被攻击者安装了木马，那你的一举一动有可能都会被监视。在这种情况下，你在这台电脑上输入的任何密码，都将会被攻击者获取。所以，这种情况是很危险滴——不管是哪种类型的密码，都可能被盗。

至于如何防止自己的计算机被植入木马，【不是】本帖的重点。俺会在本系列后续的帖子中专门介绍木马的防范。

◇弱密码猜解

所谓的“弱密码猜解”，就是说：如果你的密码比较弱，攻击者就容易猜出来。这种攻击手法，对于操作系统用户口令、网络应用口令、本地应用口令，统统适用。

而且攻击者在盗取口令的时候，通常会先尝试进行弱口令猜测。为啥捏？因为大部分用户都不太具有安全意识，口令都会比较简单（比较弱）。并且，根据【二八原理】，绝大多数的傻瓜用户会使用极少数的弱口令。所以，攻击者先把最流行的那些个弱口令挨个试验一遍，没准就已经成功了。

为了让大伙明白弱口令的严重程度，来看看2009年底的“一个案例”。

话说国外一个小有名气的交友网站（RockYou）被黑客攻破。里面大约3260万用户数据被盗。更加杯具的是，RockYou 采用明文方式存储用户的口令。因此，这3260万用户的口令也统统暴露鸟。后来有好事者把被盗的用户口令拿来分析一番。结果发现，有相当多的用户在使用一些极其弱智的口令。

用的最多的 TOP 10 分别是：

> \1. 123456
> \2. 12345
> \3. 123456789
> \4. password
> \5. iloveyou
> \6. princess
> \7. rockyou
> \8. 1234567
> \9. 12345678
> \10. abc123

据说名列第一的口令（123456）有30万人使用，真是不看不知道，一看吓一跳啊！

◇暴力破解

除了对弱密码进行猜解，攻击者还可以通过穷举的的方式，破解中等强度的密码。所谓的穷举法，就是把所有可能的字母/数字的组合都试验一遍，直到找到正确的密码。

现在 CPU 的计算能力日新月异，尤其是多核 CPU/GPU 普及之后，暴力破解的效果会越来越好。你的密码必须【很强】，才能彻底消除暴力的风险。

由于这种攻击手法，需要进行成千上万次的试错，所以比较适合针对本地应用的口令（比如破解加密的压缩文件），而不太适合对网络应用进行在线口令破解。

在下一个帖子，俺会介绍《如何构造安全的口令/密码》。

◇网络传输截获（嗅探）

在这种方式下，攻击者会通过【嗅探】的方式，分析你的上网数据。如果你在上网过程中，存在明文传输的口令，就会被截获。

非安全专业的网友，可能不太明白什么是“嗅探”，俺来稍微解释一下。攻击者会利用某些嗅探软件，收集网络上传输的所有数据。这个过程好比电话窃听。嗅探软件类似于窃听器；你的上网数据类似于电话的通话内容。

这几年，随着现在无线网络（WiFi）的普及，网络嗅探的风险大大增加，列位看官切不可掉以轻心哦。

◇客户端截获

所谓“客户端截获”，通常是针对网络应用的口令而言。举几个例子。

举例1：

很多网友上网时，为了免去输入口令的麻烦，会让浏览器帮忙记住口令。通常浏览器会把这些口令保存在某个文件中（可能以明文方式，也可能以密文方式）。如果某天你的电脑中了木马，那么木马程序有可能会盗走这个保存口令的文件。然后攻击者就可以通过分析该文件，破解出你保存过的所有网络应用的口令。

举例2：

如果你是软件公司的开发人员，多半你会使用某种源代码版本管理工具（比如 SVN、CVS、等）。为了免去每次操作时输入口令的麻烦。通常开发人员会让这些客户端软件记住用户名和口令。如果哪天你中了木马，或者笔记本电脑被盗，那么攻击者同样可以破解你保存下来的口令，进而用你的身份盗取源代码。

◇服务端截获

和“客户端截获”方式相对的，还有“服务端截获”。具体是啥意思捏？俺来解释一下。

凡是利用口令进行验证的软件系统，都需要存储和口令相关的信息。否则的话，软件系统就无法验证用户输入的口令。如果攻击者能够拿到这些口令的关联信息，那他/她就有可能分析出口令是啥。

（如果你不是搞 IT 专业滴，下面这段可能看不太明白。不过没关系，不影响你理解后续章节）

简而言之，通常有三种方式来存储口令的关联信息：1、存储口令的明文；2、存储口令经过加密后的密文；3、存储口令的散列值。

第一种方式是最土鳖的，稍微先进一些的系统，都不这么干了。后面两种方式，虽然看不到明文，但是攻击者还是有可能通过某种技术手段，反推出口令的明文。具体细节，本文就不再多说了。

那攻击者如何获得存储在软件系统的口令关联信息捏？其实前面提到的 RockYou 网站的杯具，就是一个很好的例子。俺再举另一个例子。

比如说：某个 Linux/Unix 服务器存在安全漏洞，攻击者利用此漏洞搞到了 `/etc/shadow` 文件。那么攻击者就可以采用上述提到的暴力破解的招数，攻破该服务器上所有强度较弱的口令。

★攻击者如何通过【非技术】手段搞定的你的密码？

说完了技术手段，自然就得再说说【非】技术手段。所谓的非技术手法，也就是社会工程学手法（关于社会工程学的扫盲，请看“这里”）。用于盗取密码的社会工程学手法，大概有如下几种。

◇偷窥

偷窥是最简单的一种社会工程学攻击手法。虽然简单，但是有效。比如很多盗取银行卡的家伙，就是偷窥的手法，得到被害人的银行卡密码。

◇网络钓鱼（phishing）

另外一个骗取口令的方式，就是通过网络钓鱼。比如某些攻击者，会伪造一个银行的网站。其界面和真实的网站一模一样。然后通过某种方式（比如：虚假链接、欺诈邮件、DNS 欺骗 ...），引诱你到这个网站上。由于假网站和真网站的界面很像，你可能信以为真，然后在假网站中输入你的用户名和密码。

有些高明的钓鱼网站，会采用类似Web代理的技巧：把你的所有输入操作，转而提交给真网站；然后把真网站输出的界面，再转回给受害者看。这样的话，受害者就跟在真实网站进行插作，没啥区别，不易看出破绽。

更多关于网络钓鱼的介绍，可以参见维基百科的“这个页面”。

◇分析

如果攻击者对你比较了解，那么他有可能通过深入的分析，攻破你的口令防护。

有木有觉得很神奇？很匪夷所思？其实这种招数很常见，且不算太难。俺来举个例子。

相信很多网友都用过电子邮箱的找回口令功能。当你口令遗忘之后，可以通过回答事先预设的问题，来找回口令。很多不太专业的用户，预设的问题都很简单（比如：你的手机号是多少？比如：你的生日是哪天？）。对于这类过于简单的问题，攻击者可以很容易地找到答案，从而窃取到你的邮箱口令。

◇欺骗

最近几年，通过电话诈骗，骗取银行卡密码的案例越来越多。这种作案手法，就属于社会工程学中，“欺骗”的范畴。其实在IT领域，某些黑客也会利用这种手法来获取口令。具体的一些欺骗的伎俩，可以参见俺之前的“社会工程学系列”帖子。

★结尾

介绍到这里，列位看官对黑客盗取口令的手法，应该有一个初步的认识了。本系列的下一个帖子，具体介绍如何构造安全的口令/密码。

[如何防止黑客入侵3]：如何构造安全的口令/密码

文章目录

在上一个帖子，俺介绍了攻击者，是如何攻破口令这道关口的。为了避免口令被轻易地破解，有必要了解构造安全密码的技巧。所以，今天就来介绍此话题。

★【不要】共用口令/密码

俺发现有相当多的同学喜欢靠一个口令包打天下。这是相当相当危险的事情。同一个口令，用的场合越多，则泄密的危险越大。而一旦泄露，你的安全防线就会全面崩溃。

所以，今天要讲的头一个要点，就是绝对不要在所有（大多数）场合，使用同一个口令。

★密码的【分级机制】

由于共用口令存在很大的风险，比较稳妥的办法就是——每一个场合仅使用一个密码。但是很多人会抱怨说：这样会很繁琐，增加了很多的麻烦。那如何才能做到既安全，又不太麻烦捏？这就要引入密码的分级机制。

根据安全圈内一个人所共知的常识：越安全的措施，通常也就越麻烦，成本也高；反之亦然。另外，根据二八原理，非常重要口令毕竟只占少数。所以，就像电影要有分级机制一样，你的密码/口令也要引入分级的概念。通过分级机制，对大多数不太重要的口令，可以采取简化的安全措施；而对少数重要的口令，采取高度安全的措施。

下面，就来介绍一下，如何对不同的口令，进行分类。

◇第1级：不重要的口令

所谓不重要的口令，就是说万一被盗了或者忘记了，对你没啥损失。

比如，俺经常碰到一些土鳖的论坛，只允许注册会员从上面下载附件。因此俺就经常临时注册一个账号，然后登录上去下载东西。这类账号，基本上都属于一次性的（用完即扔），所以重要程度很低。

对于那些不重要的口令，基本上不用考虑太多安全性的因素。随便设置一个即可。

◇第2级：重要但少用的口令

对于重要的口令，还要根据其使用的频繁程度，再区别对待。有些口令虽然重要，但是使用的频度很低。由于这类口令很少使用，所以设置得麻烦一些，问题也不大。

比如俺管理的一些研发的服务器（比如源代码服务器），其重要程度非常高，但是平常基本无需登录。

◇第3级：重要且频繁使用的口令

最后这类口令，既重要，又经常用。所以，设置这类口令就比较讲究。要同时兼顾安全性和易用性。比如自己日常使用的操作系统用户密码，就属于此类。

★一些反面教材——【脆弱】密码的举例

说完了分级机制。接下来俺先列举一些反面教材，让大伙儿看看，啥样的口令算是脆弱的？（顺便说一下：2011年底，国内各大网站纷纷被脱库，大量用户口令侧漏。俺专门写了一篇博文，分析国内用户的口令习惯）

◇口令和用户名一样

无需多说，这种情形的口令，非常脆弱。

◇口令是一串简单数字

在上一个帖子，俺举了RockYou 网站用户数据被盗的案例。在该网站3200万用户中，最受欢迎的十大弱智口令分别是：

```
1.  123456
2.  12345
3.  123456789
4.  password
5.  iloveyou
6.  princess
7.  rockyou
8.  1234567
9.  12345678
10. abc123
```

从这个 TOP 10 可以看出，有一半是采用连续数字。所以，用连续的数字串（包括顺序和逆序）作密码，也是很愚昧滴。

◇口令太短

如果你的口令小于6个字符，是很容易被暴力破解滴。毕竟，小于6个字符的所有组合，也没多少个。对专门穷举密码程序来说，那简直是小菜一碟。

◇用英文单词作口令

用【单个】英文单词作口令，也很容易被破解。毕竟，常用的英文单词，也就千把个；算上冷僻的，也就几万个。

在许多年以前，就有黑客专门收集整理了英文单词的列表（称之为“口令字典”）。而且这个字典是根据单词的使用频度进行排序。有了这种密码破解字典，密码破解程序就可以轻易猜解出那些使用单个英文单词的密码。

◇用日期作口令

有些同学希望用某个具有特殊意义的日期（比如：生日、结婚纪念日...）作为口令。要知道这种伎俩也是不灵滴。因为常见的日期，大都分布在最近100年的范围内。所以充其量，可能的个数也就大约是365*100个。即便把不同的日期表示格式考虑进去，也多不了几倍。在这个数量级上，对于暴力破解工具而言，还是小菜一碟。

◇其它的烂口令

上述列举的这几种情况，大伙儿一定要避免。另外，你还可以去围观一下某老外整理的一个滥口令大全（这老外真有耐心）。提醒一下：这个列表是根据欧美用户统计的，未必适合中国的国情。

★如何构造【复杂】密码?

前面已经说了：口令太简单，容易被破解。但是太复杂的话，万一自己也忘了，那可就完蛋了。所以，很多网友都纠结于口令到底该复杂到什么程度。俺的经验是：**口令要做到对自己简单，对别人复杂。**

下面就来介绍俺在这方面的经验。

◇用多个单词构成词组

前面提到，如果用【单个】英文单词作密码，容易遭受字典攻击。为了避免字典攻击，可以考虑由2~5个英文单词构成密码。如果你英语不灵光或你比较习惯中文，也可以考虑用2~5个汉字的拼音来构成密码。

优点

由于能显著增加密码长度，可以抗击暴力破解。

缺点

有可能需要改变你记忆密码的习惯。

口令中仅包含字母，复杂度不够高。

◇插入特殊字符

刚才提到了用多个单词或汉字拼音构造密码。为了让密码的强度再好一些，还可以在单词或汉字拼音之间，插入一些特殊字符。

最常见的是插入空格。当然，你也可以考虑插入其它字符（比如：下划线、减号、斜杠、井号、星号、等）。

通常进行暴力破解时，为了加快破解进度，都只针对字母和数字进行暴力破解。如果你的口令中含有特殊字符，会大大提高攻击者的难度。

优点

由于口令包含较多特殊符号，复杂度大大提高。

缺点

很多特殊字符的输入，要依赖于 SHIFT 键辅助。对于键盘指法不流畅的同学，可能会影响你输入密码时的击键速度，给偷窥者留下可乘之机。

◇字符变换

所谓的字符变换，就是用形状类似的字母和数字进行相互替换，通过这种变换，可以规避前面提到的基于口令字典的攻击。

常见的变换有如下几种：

```
字母o 和 数字0
字母l 和 数字1
字母z 和 数字2
字母s 和 符号$
字母g 和 数字9
字母q 和 数字9
字母a 和 符号@
字母b 和 数字6
字母x 和 符号*
```

假设俺想用单词 `program` 作为口令，那么经过上述的变换之后，就成为 `pr09r@m`

很明显，变换之后的口令同时具有字母、数字、符号。强度相当好 :)

以上变换仅仅是举例。你可以对俺给出的这几个变换，进行扩展，以满足自己的习惯与偏好。

优点

【不用】改变你原先的记忆习惯。

由于口令包含较多特殊符号，复杂度大大提高。

缺点

如果你想好的口令中，恰巧所有字母都没有对应的变换，那就比较不爽啦。

◇键位平移

这个招数也比较简单，就是在进行键盘输入时，把手【向右】平移一个键位。通常咱们在盲打时，两只手的食指分别对着字母 `F` 和字母 `J`。平移之后，则食指对着 `G` 和 `K`。

假设俺想用单词 `program` 作为口令，那么经过上述的变换之后，就成为 `[tphts`

经过这种输入法，口令已经面目全非。但是对你自己来说，并不难记。

优点

【不用】改变你原先的记忆习惯。

口令看起来完全没规律。

缺点

依赖于 QWERT 的键盘布局。万一哪天你想在非 QWERT 键盘（比如某些手机键盘）上输入口令，那你就歇菜了。

◇藏头诗

在某些古代小说的情节中，经常可以看见藏头诗的桥段。藏头诗的点子，也可以借用来构造安全口令。

为了用此招数，你先要想好一句令你印象深刻的话。这话可以是中文，也可以是英文、法文、火星文……反正只要是你熟悉的语言既可。最好这句话的字数（单词数）在8~20之间。然后你把这句话每一个单词的头一个字母取出来，组合成一个口令。如果是中文的话，就把每一个字的拼音的声母取出，组合成口令。

假设俺想好的话是：“只有偏执狂才能生存”。那么用拼音的声母表示就成为 `zypzkcnsc`

优点

【不用】改变你原先的记忆习惯。

口令看起来完全没规律。

缺点

口令中仅包含字母，复杂度不够高。

如果句子中的字数（单词数）不够多，效果就不够好。

对于港台的同学，由于没学过汉语拼音，只好用英文的藏头诗了。好在港台的英语教育通常比大陆好，应该关系不大 :-)

◇巧用 SHIFT 键

在构造口令的时候，适当地组合一下 SHIFT 键，有时也可以达到不错的效果。假如你的口令中，有部分字符是数字，那当你输入口令时，按住SHIFT键会让这些数字字符变为特殊符号。

优点

【不用】改变你原先的记忆习惯。

由于口令包含较多特殊符号，复杂度大大提高。

缺点

万一你原先的口令仅有字母，没有数字，则密码的强度会稍微打折扣。

由于要依赖于 SHIFT 键进行切换，会影响你输入密码的击键速度。这会给偷窥者留下可乘之机。

◇运用数学等式

还有一种又好记，看起来又复杂的密码构造方式——运用数学等式。

比如，可以把密码设计成：`7+8=15`

虽然只有6个字符，但是由于包含了符号，已经有一定的强度。如果你觉得6字符太少，可以很容易增加字符数及复杂度，比如改为：`37+(9*2)=55`

如果你觉得还不够复杂，还可以搞得再变态一点——把某个数用英文表示。比如：`two+7=nine`

优点

密码同时包含了字母、数字、符号。标准的高复杂度！

缺点

需要改变你记忆密码的习惯。

一旦你的口令被别人看到，别人很容易就可以发现你构造口令的规律。

◇利用散列值（哈希函数）

最后，来说一种俺的看家本领——利用散列值构造口令。

不熟悉 IT 技术的同学，可能不了解"散列值"是啥玩意儿。俺不想多浪费口水解释，好奇的同学请看俺的另一篇博文《扫盲文件完整性校验——关于散列值和数字签名》。

要构造基于散列值的密码，有好几种散列算法可供选择。对于不太懂技术的网友，俺建议用 CRC32 散列算法。为啥用它捏？因为这玩意儿操作起来比较方便。比如，假设俺想得到某个文件的 CRC32 散列值，只要用 7-Zip、WinRAR 之类的压缩工具，把它压缩成 zip 格式的文件，然后就可以看到该文件的 CRC32 值了（因为 zip 格式用 CRC32 散列算法作为文件的校验码）。不信你随便拿手头一个 zip 格式的文件打开来看看就明白鸟。

因为 CRC32 生成的散列值比较【短】，对于懂技术并且安全要求较高的网友，可以用散列值【更长】的散列算法（比如：MD5、SHA1、SHA256 ……）。

现在，详细说一下基于散列值的密码如何构造（以 CRC32 为例，其它散列算法依样画葫芦）

首先，你先想好一个字符串，作为计算散列的种子。这个字符串不需要很复杂，也不需要很长。比方

说你叫张三，那你可以拿张三的拼音声母 `zs` 作为【种子串】（注：此处纯属举例，实际情况中，你应该用【更长】的字符串作为种子串）。

接下来，假设你有一个 hotmal 的邮箱，需要设置口令。你可以先用记事本（notepad）生成一个 txt 文件。里面先写上种子串 `zs` 再写上 `hotmail`，存盘。然后把这个 txt 文件用工具压缩成 zip 格式，看一下它的 CRC32 校验码 `9C9041C0`，然后就拿它作为密码。

如果你再有一个 gmail 邮箱需要设置口令，只要同样地，新建一个 txt 文件并写入 `zsgmail`，同样计算 CRC32，就可以得到另外一个值 `03B2F77D`。大伙注意到没有？这两个值看起来没有任何关联性，而且从这两个口令，也看不出和种子串 `zs` 有啥关系。

优点

密码同时包含了字母、数字，但是没有特殊符号。复杂度属于中高！

由于散列值具有随机性。也就是说，你看到的绝大多数散列值都没啥规律。

由于散列值具有不可逆性。也就是说，即便有一个密码暴露了，攻击者也看不出规律。

即使有一个密码暴露，别人完全看不出规律。

缺点

这种密码是完全随机的，常人是【不可能记住】滴。所以，在密码分级机制中，它仅适合第二级的密码。第三类密码没法这么玩。

此招数的进阶：

1. 你可以把 CRC 算法换成其它散列算法（比如： MD5、SHA1、SHA256 ……），就可以轻易构造出【超长的】密码或口令（几十个字符，甚至上百个字符）。
2. 如果你自己会写点小程序或小脚本，你可以进行 N 次散列（N 可以是几千或几万）。这样一来，别人拿到你的某个密码后，更加难逆向分析出你的"种子串"。因此也就无法分析出由种子串构造出来的其它密码。

★结尾

今天又花了不少篇幅，总算把俺平生积累的，关于如何构造复杂密码的经验，都讲完了。如果哪个网友还有其它独到的经验，希望来信和俺分享。如果俺觉得实用，也会补充到本文中。

本系列的下一个帖子，会说说安全漏洞的基本防范。

2010年8月2日 评论数：10楼 / 16条 标签： IT, IT.信息安全

[如何防止黑客入侵4]：安全漏洞的基本防范

文章目录

前面用2个帖子来介绍口令方面的安全（在"这里"和"这里"）。今天扫盲一下跟安全漏洞相关的知识，为下一篇（Web 相关的防范）做好铺垫。考虑到俺博客的读者群，本文主要拿Windows桌面系统来举例，并且尽量说得浅显一点。

★扫盲基本概念

◇什么是漏洞?

所谓的“漏洞”，简单来说，就是会引起各种问题的软硬件缺陷（软件业的行话叫 Bug）。要知道，任何东西都不可能是完美的，软硬件系统也不例外（毕竟开发软硬件系统的程序员/工程师，也是凡人，也会有出错的时候）。

◇什么是安全漏洞?

在上述提到的缺陷中，那些会被攻击者加以利用的，并因此导致安全问题的缺陷，就是所谓的“安全漏洞”。

◇什么是补丁?

补丁是一个很形象的说法。如果你衣服破了个洞，只要打个补丁，不需要把整件衣服换掉。同样的，如果你的某个软件有漏洞，也只要打一个补丁，不需要重新安装新版本的软件。大多数情况下，补丁和漏洞是配套的。

◇什么是攻击代码 / 攻击程序?

为了利用某个安全漏洞，黑客需要运行某个程序，这个程序就叫做攻击程序（也叫“攻击代码”）。通俗地说，攻击程序和补丁之间的关系，就好比矛和盾之间的关系。

★安全漏洞的分类

除了上述的基本概念，你还需要大致知道安全漏洞的分类方式。

◇按照所在软件的类型分类

根据出现漏洞的软件的类型，可以把漏洞分为“操作系统漏洞”、“应用软件漏洞”、“Web 漏洞”等。

所谓的“操作系统漏洞”，就是操作系统本身有的安全缺陷。

所谓的“应用软件漏洞”，就是你安装的软件所内含的安全缺陷。

近几年来，基于 Web 的攻击日益增多，所以把 Web 漏洞单独分一类。所谓的“Web 漏洞”，也就是跟Web相关的漏洞——包括了浏览器本身的漏洞和网站的漏洞（本系列的下一篇会具体介绍这方面的知识）。

◇按照危险级别分类

还可以根据漏洞的危险程度，进行分类。通常按照“高中低”分三级（也有把危险级别定为5级的）。级别越高，就越危险。

对于高危的漏洞，有可能导致攻击者在你的电脑上植入木马。

◇按照攻击代码的位置分类

针对攻击代码所处的位置，可以把漏洞分为“远程漏洞”、“本地漏洞”两类。

所谓的“远程漏洞”，就是说，攻击者只需要在另外一台机器运行攻击代码，就可以让你的电脑中招。

所谓的“本地漏洞”，就是攻击者的攻击代码必须在你的机器上执行。

◇按照补丁的情况分类

刚才已经解释过了“补丁”和“漏洞”之间的关系。

对于大部分漏洞而言，都有对应的补丁；但是少数漏洞没有补丁。没有补丁的漏洞是很危险的。为啥会出现这种情况捏？俺稍微解释一下。

1. 未公开的漏洞

有些黑客发现某个漏洞后，没有在圈内公开，也没有告知对应的软件厂商。那么这种漏洞就变成“未公开的漏洞”。因为没有公开，软件厂商不知情，自然也就没有发布补丁。

这种漏洞【最危险】，有可能长期被攻击者用来入侵。据俺所知，有些高危漏洞已经在黑客圈内流传多年，而相应的软件厂商依然不知情。

2. 零日漏洞（Zero-Day）

如今互联网很发达，某些资深且勤奋的攻击者可以对漏洞的发布作出快速反应。一旦某个漏洞的细节被公开，他们可以在24小时之内制作出相应的攻击代码。而这个时候，软件厂商多半还没来得及发布补丁。那么这些攻击者就可以利用这个时间差，进行入侵活动。所以，安全圈内也把这种漏洞称为“零日漏洞”（洋文叫：Zero-Day 或 0-Day）。相对于“未公开漏洞”，“零日漏洞”可利用的时间段会短一些。比较负责任的软件厂商通常会在一周或一月之内发布补丁。不过捏，也有些不靠谱的公司（包括大公司），要拖上好几个月才发布漏洞补丁（比如 Oracle）。

★漏洞的防范措施

针对漏洞的这几种不同分类，俺分别介绍一下几种基本的、常见的防范方式。

◇个人防火墙

个人防火墙主要用于防范“远程漏洞”，对于“本地漏洞”，防火墙基本帮不上忙。

因为大多数远程漏洞，都存在于你机器对外开启的监听端口中。个人防火墙可以阻止这些端口对外开放，从而避免潜在的漏洞被攻击者利用。

自从 Windows 2000 开始，微软就在操作系统中内置了防火墙功能。对于 Windows XP 以及之后的版本，可以直接到控制面板中开启它。

另外，俺强烈建议：**【不要】用国产的防火墙产品。**别要怪俺崇洋媚外，具体原因可以参见俺之前的帖子（在“这里”）。

如果你的电脑只是用来上上网、聊聊天、看看电影、用用办公软件，那你完全可以把防火墙设置成——不开放任何对外的端口。这样一来，即便你的电脑中存在远程漏洞，也不易被攻击者利用。

◇定期升级系统补丁

Windows 系统的漏洞一直比较多——毕竟用户群太大，容易被黑客盯上。所以从很多年以前，微软就开始定期提供 Windows 补丁。具体的做法是每月的第二个星期二，发布新发现的漏洞的补丁。另外，如何发现高危的漏洞，也会临时发布紧急补丁。

从 Windows 2000 开始，系统就支持自动的补丁升级机制。你只要在“控制面板”里打开“自动更新”这

个功能，然后选择“自动”方式。你一定要确保系统的"自动更新"机制处于启用状态。这就可以堵住很多操作系统漏洞，从而降低攻击的风险。

提醒一下：自动更新不是 Windows 独有滴。其它一些用户群比较大的桌面系统（比如：Mac OS，Ubuntu）也提供了自动更新安全补丁的功能。

◇启用软件的自动更新

有一些做得比较好的软件，会内置自动更新功能（比如：Firefox、Chrome ...），一旦其官方网站有新的版本或补丁，就会自动下载并更新。如果你担心这类软件有安全问题，可以启用它们的自动更新功能。

◇使用小众且活跃的软件

俗话说树大招风。越知名的软件，就越容易引来黑客的注意，被发现安全漏洞的概率也会增大。

比如说：IE 的用户群最大，针对 IE 漏洞的挂马攻击是各种浏览器中最普遍的；Firefox 相对就少很多；而 Chrome 和 Safari 就更少了（不过捏，随着 Firefox 和 Chrome 的流行，针对这两款浏览器的攻击也多起来了）。

再比如说：同样是 PDF 阅读器，Adobe Reader 被曝光的漏洞就比较多，相对而言，Foxit Reader 和 PDF-XChange Viewer 就没这么多问题。

为啥俺还要强调【活跃】捏？小众软件根据活跃程度可以分为：活跃 or 不活跃。不活跃的软件，通常说明：开发者维护该软件不够积极，也就意味着该软件在【漏洞修复】方面不够迅速/及时——这就会导致安全风险；反之，活跃的软件，就没这个问题。

关于安全漏洞的基本防范，今天就聊到这里。考虑到近年来，针对 Web 攻击的情况剧增，本系列后续的帖子会说一下 Web 相关的话题。

[如何防止黑客入侵5]：Web相关的防范（上）

文章目录

由于俺比较懒，导致本系列已经中断了2年之久。上星期有读者留言，希望俺尽快把本系列补上。再加上昨天看到新闻，说 Java 7 爆出全系列的高危漏洞。凡此种种，促使俺补上了本系列的第5篇，关于 Web 的防范。这部分的内容比较长，为了避免大伙儿阅读疲劳，俺把《Web相关的防范》分为上中下3个部分。

★Web安全的重要性

在聊正题之前，先给大伙儿强调一下“Web 安全”的重要性。

如今互联网非常普及，大部分的家用电脑和商业电脑，都具备联网功能。而且大部分电脑只要一开机，就处于联网状态。作为电脑的使用者，有相当一部分时间是花在 Web 浏览（俗称网上冲浪）。在这样的环境中，Web 就成了恶意软件（病毒、木马、蠕虫、勒索软件.....）最理想的一种传播媒介。据说如今大部分电脑中招，都与 Web 有关。

正因为如此，才把 Web 相关的内容，单独汇总一篇。接下来，俺先介绍一下攻击者常见的招数，然后再介绍一下各种应对措施。

★Web 相关的攻击手法

◇嗅探（sniffer）

所谓的“嗅探”，就是攻击者利用某些技术手段，截获你的网络数据流并进行分析，从而获取某些有价值的信息。通常来说，“嗅探”只是入侵的初始阶段（准备阶段）。攻击者通过“嗅探”获取到的信息，通常用来进行辅助后续的入侵行动。

举例：
很多人喜欢通过公共场所的 WiFi 热点上网。假如你使用的 WiFi 热点没有设置为强加密。那么，某个攻击者就有可能利用 WiFi 嗅探工具，截获你的上网流量。如果你正好在收发 Web 邮件，而且没有通过 HTTPS 加密（好多国内的 Web 邮箱【不】支持全程 HTTPS 加密）。那么，攻击者就可以看到你的收发的邮件内容。

不过捏，关于嗅探的防范，不是本文的重点。因为俺之前写一个系列博文《如何隐藏你的踪迹，避免跨省追捕》，里面介绍的各种招数（比如加密代理的使用），已经足以帮你对抗“嗅探”了 :)

◇钓鱼（phishing）

“钓鱼攻击”包括很多种，基于 Web 的网络钓鱼是其中之一。
由于“钓鱼攻击”属于社会工程学的范畴，也不是本文的重点。今后有空的话，单独写一篇“关于钓鱼攻击的防范”。

◇利用浏览器自身的安全漏洞

在本系列前一个帖子里，俺已经扫盲了"漏洞"、"补丁"等概念以及相关的一些常识。健忘的同学，可以再去温习一下。
在软件行业中，几乎每一款软件都会有漏洞——浏览器自然也不例外。浏览器的漏洞有很多种，其中一类叫做"安全漏洞"。顾名思义，就是会导致安全问题的漏洞。
如果某款浏览器的安全漏洞被攻击者发现，而浏览器厂商自己还不知晓。那么攻击者就可以利用该漏洞，发起广泛的攻击。

举例：
假设某个黑客研究 IE 的内核，首先发现 IE 存在一个“渲染图片导致缓冲区溢出”的漏洞。由于该漏洞是独家发现，只要该黑客不公开漏洞的信息，连微软（也就是 IE 浏览器的厂商）也会蒙在鼓里。因此，也就【没有】针对该漏洞的补丁。那么，这个黑客会如何利用该漏洞捏？
\1. 首先挑选一张图片（为了吸引人，通常会选一张美女图之类的照片），然后精心地嵌入一段攻击代码在图片内部。
\2. 把这张图片放到网上（比如张贴到某个大型论坛，再配上一个吸引人的标题）。
\3. 过不了多久，就会吸引到很多网友来围观。如果围观的网友用的浏览器不是 IE，那么他仅仅是看到一张美女图而已，不会有啥异样。如果围观的网友用的正好是有漏洞的 IE 版本，当 IE 打开那张图片的瞬间，攻击代码就会被激活（被运行）。然后，攻击代码会悄悄地在这台电脑中安装一个木马（技术行话叫“植入木马”）。之后，这台电脑就成为攻击者的肉鸡了（攻击者可以远程控制肉鸡，为所欲为）。
\4. 攻击者控制了肉鸡之后，既可以拿去卖钱（有专门的地下肉鸡交易市场），也可以偷窥机主的隐私（看看有没有网银、裸照、QQ 靓号）。如果控制的肉鸡数量巨大，还可以搞 DDOS 攻击.....

◇利用浏览器【插件或扩展】的安全漏洞

如今大部分浏览器上，都安装了插件。最常见的插件就是 Flash 插件。另外还有“PDF 插件、Java 插件”等等。

浏览器的插件也属于软件，也会存在安全漏洞，因此也可以被攻击者利用。一般来说，攻击者对插件漏洞的利用，类似于对浏览器漏洞的利用。

举例1：

2011年，美国大名鼎鼎的安全公司 RSA 遭到入侵并且被深度渗透，连看家产品 SecureID 的密钥也被偷了。

攻击者之所以能得手，就是利用了 Flash 插件的一个零日漏洞。洋文好的同学，可以看“这里”的详细报道。

举例2：

同样是在去年，有不少 Gmail 用户遭到入侵。但实际上，Gmail 本身并没有出问题。攻击者是利用 Flash 的漏洞，伪造跨站请求，然后在 Gmail 的转发列表中加入一个攻击者的邮箱。之后，被害人收到的所有邮件，都会自动转发给攻击者。

从最近几年的趋势来看，插件漏洞导致的安全问题，要多于浏览器漏洞导致的安全问题。

◇跨站脚本攻击（XSS）

最后再来说说"跨站脚本"的问题。

大部分 XSS 攻击，都是利用网站本身的漏洞。所谓的网站，其实就是若干 Web 服务器，上面运行若干软件。前面说了，只要是软件，就可能存在漏洞（包括安全漏洞）。所以，Web 服务器上的软件自然也不例外。

基于 XSS 的攻击有很多种类型，具体的技术原理也有所差异。考虑到大部分读者不是搞技术的，俺就不深入展开了。仅举一例，让大伙儿有个感性的认识。

举例：

比如某个 BBS 论坛存在漏洞——【没有】对用户发布的帖子内容（此处的“内容”，不是指文字的内容，而是指特殊字符）进行严格的检查。如果某个攻击者发现了此漏洞，就可以精心构造一个帖子，在帖子的正文中包含一段攻击脚本（通常是 JavaScript）。接下来，攻击者把这个帖子发布到该论坛上。

然后捏，如果有人浏览了这篇帖子，这段攻击脚本就会被激活，然后干坏事.....

★如何选择浏览器?

对于用户来说，浏览器是 Web 的根基。所以，谈 Web 的安全防范，首先得聊一聊如何选择浏览器。

挑选浏览器有如下几个指标供参考：

1. 浏览器的质量好不好

评判安全方面的质量，最关键的一条是：看浏览器有没有经常出安全漏洞。

2. 浏览器的更新快不快

爆出漏洞后，浏览器的开发团队是否及时出补丁或新版本。在《安全漏洞的基本防范》一文，俺介绍了【零日漏洞】的概念。浏览器修补漏洞越及时，网友暴露在“零日漏洞攻击”的时间就越短。

3. 浏览器的功能强不强

除了要看浏览器本身的功能，还要看其支持的扩展是否丰富。

根据上述指标，俺把市面上常见的浏览器，根据靠谱的程度，划分为如下三类：

◇第一类：Firefox 和 Chrome（含 Chromium）

俺个人强烈建议用 Firefox 或 Chrome 进行网上冲浪，因为这两款浏览器很符合上述指标。

质量好

本世纪初，浏览器市场被 IE 一统天下。但随着时间推移，IE 在全球的市场份额逐步被 Chrome & Firefox 占去一半以上。这充分说明 Firefox 和 Chrome 的质量很好。另外，在安全漏洞方面，Firefox 和 Chrome 也优于 IE。

更新快

说到快速更新（快速迭代），这是 Chrome 的首创。从去年开始，Firefox 也学 Chrome 采用快速版本更新。

功能强

说到功能，Firefox 刚出道时，利用丰富的扩展吸引了足够多的用户。如今，无论是扩展的种类还是扩展的下载量，Firefox 是最多的；至于 Chrome，由于出道时间晚，这扩展方面不如 Firefox，但显然比 IE 强多了。

除了上述这几条，这两款浏览器还具有如下优点：

支持的平台很多

支持三大主流的桌面系统（Windows、Mac OS、Linux），支持两大主流的移动系统（Android、iOS）。

开源项目

由于开源而且参与的程序员也多，所以软件中的漏洞容易被及早发现。

（Chromium 是开源滴；Chrome 虽然基于 Chromium，但包含【闭源】模块）

以上就是俺推荐 Firefox 和 Chrome 的理由。在本文的后续章节，俺会以这两款浏览器为主，进行介绍。

◇第二类：IE、Safari、Opera

先说 Safari 和 Opera。这两款的出道时间早于 Firefox 和 Chrome。忙活了这么多年，如今的市场份额依然很低，这已经说明某些问题。另外，俺个人觉得 Safari 和 Opera 的扩展和插件不够丰富，更新速度也不够快，所以俺不推荐。

至于 IE，曾经是市场份额最大的浏览器，而是集成（捆绑）在 Windows 系统中。为啥俺把它放到第二类捏？有如下几个原因：

\1. IE 跟 Windows 集成得太紧密。IE 如果爆漏洞，通常要等微软发布 Windows 补丁来修复。而 Windows 补丁是按月发布的——太不及时啦。

\2. 相比 Firefox 和 Chrome，IE 是闭源项目。由于源代码不公开，而且参与的人不够多，导致潜在的漏洞难以被发现。

\3. IE 用户大都是菜鸟用户（很多菜鸟只知道用系统内置的浏览器）。由于菜鸟不太懂安全防范，有些【低级】骇客就喜欢盯着这个用户群。

说到这儿，可能有同学会问：天朝的好多网银都只能用 IE（很多网银客户端依赖于 IE 的 ActiveX 控件），咋办捏？别担心，在本系列的下一篇《Web相关的防范 (下)》会谈到此问题的解决方法。

◇第三类：五花八门的【国产】浏览器

说到【国产】的浏览器，有必要谈一下浏览器的内核（也就是浏览器的引擎）。绝大部分国产的浏览器，都不是自己开发内核，而是基于老外现成的内核。常见的浏览器内核有三款，分别是：

Gecko 内核（来自于 Mozilla 开源组织，主要供 Firefox 使用）

Trident 内核（来自于微软，主要供 IE 使用）

WebKit 内核（独立的开源项目，Chrome 和 Safari 使用此内核）

几款常见的国产浏览器（360浏览器、傲游浏览器、QQ 浏览器），使用的是 Webkit + Trident 的双内核模式。

某些国产浏览器把双内核作为吹嘘的亮点。但在安全层面，双内核反而会带来安全问题。假如你手头的国产浏览器采用了 Webkit + Trident 双内核。只要这两款内核中，有一个爆出安全漏洞，你就有可能中招。也就是说：双内核会增加你中招的概率。

俺极力反对【国产】浏览器，还有另一个原因——政治层面的安全问题。朝廷为了监控屁民在互联网上的一言一行，会跟国产浏览器厂商合作，通过浏览器记录网民的行踪。

举例：

前几年腾讯搞的"TT浏览器"，会把用户上网行踪记录在某个文件中。

至于 360，名声更是臭不可闻。360 浏览器本身就存在收集用户隐私的问题，居然还好意思自称是"安全浏览器"。而且大伙儿别忘了，奇虎公司跟 GFW 一直保持着暧昧的关系哦。

综上所述，俺个人【非常反对】使用国产浏览器。

★如何选择插件和扩展?

说完浏览器的选择，再来聊聊如何选择插件和扩展。

◇插件和扩展的【区别】

先来扫盲一下插件和扩展的区别（连很多 IT 技术人员都把这两者混为一谈）。所谓的插件，洋文叫"plugin"；所谓的扩展，洋文叫"extension"。两者的区别如下：

插件

在功能上，插件通常是用来渲染 HTML 页面中的 `<object>` 或 `<embed>` 标签。

插件通常实现比较【底层】的功能，通常以平台相关的代码（本地代码）编写，可以调用操作系统的 API。形式上，插件以动态库（Windows 上就是 DLL 文件）的方式，加载到浏览器的进程内。由于使用本地代码编写，插件通常依赖于特定的操作系统（不同系统的插件不能混用）。

举例：
Flash 插件
媒体播放器插件
PDF 插件
Java 插件

扩展

扩展，顾名思义，是用来扩展浏览器自身的功能。所以，扩展可以调用浏览器自身的 API，但是大部分扩展【不能】调用操作系统的 API。

一般来说，扩展是跟操作系统无关的。比如 Firefox 的大部分扩展，既可以用于 Windows 平台的 Firefox，也可以用于 Linux 和 Mac 的 Firefox。

举例：
俺曾经推荐的 GreaseMonkey，就属于扩展。

◇插件和扩展在安全方面的差异

由于插件比较底层，一旦出现高危漏洞（比如能够执行本地代码的漏洞），攻击者就可以在操作系统中植入木马。可以这么说，插件出现漏洞，其危险性类似于浏览器出现漏洞。

相对而言，扩展出现漏洞，其危险性往往不如插件严重，通常也不会导致攻击者植入木马。

◇尽量使用口碑好的扩展

虽然扩展出漏洞导致的危险性不如插件那么高，但也不能掉以轻心。

俺的经验是：尽量使用知名度高且评价好的扩展。这样的扩展通常成熟度也比较高——即使出了漏洞，更新也比较及时；这类扩展也会有更多安全研究人员对其进行研——即使有漏洞，也更容易被发现。

反之，对于某些很少人用的扩展，最好敬而远之。顺便提一下。某些层次低的入侵者，甚至会把木马伪装成浏览器扩展，再忽悠一个很花哨的功能，然后放到网上给大伙儿用。

◇尽量避免使用【插件】

从上述对比可知，插件如果出现漏洞，危险性很高。所以，俺的建议是：**尽量避免使用【插件】**。

不过捏，避免使用插件，说起来简单，但是做起来有点难度。其它插件，说不用就不用了。但是 Flash 插件，实在是用得太广泛了（视频网站用到它，网页休闲小游戏也用到它），估计大伙儿难以割舍啊。

不幸的是，Flash 插件又最危险。一方面是因为 Adobe 的程序猿，安全意识太差；另一方面是因为 Flash 是用得最多的插件，成为攻击者的重点研究对象。根据2011年的统计数字，去年一年，光是**【高危漏洞】**，Flash 就爆了4次——当之无愧地坐上漏洞排行榜的头把交椅。

面对 Flash 插件，该咋办捏？列位看官，请听下回分解。（下一篇会尽快发布）

[如何防止黑客入侵6]：Web相关的防范（中）

文章目录

在本系列的前一篇聊了些基础性的东西，包括：常见的攻击手法、如何选择浏览器和插件。今天，俺继续介绍几个相对高级一点的话题。

★如何防范浏览器和插件的漏洞?

在前一篇已经告诉大伙儿“如何选择浏览器”。但是光知道这个是不够滴！因为浏览器也是软件，只要是软件就可能会出现漏洞（包括安全漏洞）。

即使你按照俺的建议，选择 Firefox 或 Chrome 作为日常的浏览器，也【无法完全避免】浏览器自身出漏洞的问题。而且浏览器的漏洞中，有一些是没有补丁的高危漏洞（包括“未公开漏洞”&“零日漏洞”，俺在前面的帖子里介绍过）。因为没有补丁，所以这类高危漏洞就特别危险。这就引出了第一个问题：**如何防范浏览器的漏洞?**

另外，在浏览器插件中，Flash 插件既是最危险的插件，也是使用最广的插件。这就引出了第二个问题：**如何安全地使用危险的插件?**

要解决上述2个问题，可以使用同一个原则，那就是：**【对浏览器进行隔离】**。具体的隔离方式有很多种，今天俺由浅入深，分别介绍一下。

★“多浏览器”的方案

◇操作步骤

这招是最简单的——就是在一台电脑上安装多款【不同内核】的浏览器。具体步骤俺就不多说了，节约点口水。

◇优点

1. 解决网银的问题

前面提到了国内网银依赖于 IE 的问题。但是 IE 的安全性又不如 Firefox 和 Chrome，咋办捏？最简单的办法就是同时安装两款浏览器（比如 IE + Firefox）。平时你上网的时候，用 Firefox，需要用网银，则改用 IE。

由于你仅仅在使用网银的时候，才开启 IE，大大降低了 IE 被入侵的机会。

2. 解决跨站脚本（XSS）的问题

使用多种浏览器，还可以帮你解决跨站脚本攻击的问题。

单纯的XSS攻击，主要是依赖 JavaScript。而 JavaScript 只能在浏览器进程内起作用，无法跨浏览器进程。所以，如果你同时使用 A B 两款浏览器。如果 A 浏览器发送 XSS 攻击，通常不会影响到 B 浏览器。除非这个 XSS 攻击结合了浏览器漏洞或插件漏洞。那么，多款浏览器是否能防范浏览器漏洞和插件漏洞捏？请看往下看。

3. 部分解决高危插件（Flash）的使用问题

俺在上一篇提到了利用 Flash 入侵 Gmail 的案例。

比方说，你同时用 Firefox 和 Chrome。其中 Firefox 安装 Flash，用来上普通的网站；而 Chrome 不装或禁用 Flash，专门用来上 Google 的相关服务器（比如 Gmail）。

某天，你收到一封恶意的邮件，该邮件会利用 Flash 的漏洞来入侵用户的电脑。幸运的是，你用来访问 Gmail 的 Chrome 上没有 Flash 插件（被禁用），于是攻击者的阴谋没有得逞 :-)

◇缺点

1. 无法彻底解决浏览器漏洞和插件漏洞的问题

细心的读者会注意到，刚才那个小节的标题是部分解决高危插件（Flash）的问题。为啥俺要特地强调【部分解决】捏？因为浏览器的漏洞和插件的漏洞有很多种。有些漏洞只是让浏览器崩溃，无伤大雅；而有些漏洞可以导致在本地执行代码，并且能进一步植入木马。一旦你碰到后者，即使你采用“多种浏览器”的方式，也抵挡不住。

某些天真的网友可能会问：浏览器厂商和插件厂商不是会升级补丁吗？但是问题在于，有少数一些漏洞是没有补丁的。为啥会这样捏？请看本系列上一篇《安全漏洞的基本防范》，里面介绍了"未公开漏洞"和"零日漏洞"。

不过别担心，俺后面还会介绍更高级的招数，来防范这种情况。

2. 灵活性不够

对于“多浏览器”的方案，还有一个缺点是不够灵活。

因为你要为每一种用途安装一款浏览器，如果你的用途比较多（比如：上网银、普通浏览、Web 邮箱、看视频），就要装好几款。

另外，有些用户就喜欢某一款浏览器，你让这些用户装好几款浏览器，他们会觉得不爽。

对于“灵活性不够”的缺点，大伙儿可以参考接下来要介绍的第二招——“多实例”的方案。

★【多实例】的方案

◇什么是浏览器的多实例?

所谓的浏览器多实例，有时候也称之为“Multiple Profiles”。

不论是 Firefox 还是 Chrome，默认安装的时候，只有一个实例（Profile）。和浏览器相关的各种信息，包括：插件、扩展、外观（皮肤）、页面缓存、cookie、等等，都存储在这个实例中。

反之，如果使用多实例，每个实例都具有独立的插件、独立的扩展、独立的外观（皮肤）、独立的页面缓存、独立的 cookie、等等。不同实例之间是相对隔离的，【不会】互相影响。

对于 Chrome，再特别提醒一下：

Chrome 同时支持“Multiple Profiles”与“Multiple Accounts”。但别把这两者搞混了。即使你配置了多个 Accounts，依然在【同一个】实例里。

◇操作步骤

多实例的配置，很多人不晓得。所以俺详细说一下。考虑到 Windows 用户居多，俺以 Windows 举例。Linux 和 Mac OS 的操作步骤大同小异。

多实例的 Firefox

对于 Windows 上的 Firefox，Profile 都放置于 `%APPDATA%\Mozilla\Firefox\Profiles` 目录下。

假设你想创建一个【全新的】Firefox 实例，可以通过如下命令行来启动 Firefox。

```
"{PATH}\firefox.exe" -P "XXXX" -no-remote
```

上述命令中，`{PATH}` 表示你的 Firefox 的【安装目录】。`XXXX` 表示你要创建的【实例名】（实例名可以是任意字母组合，你也可以把 `XXXX` 改成其它名称）。

运行完上述命令，Firefox 会启动出一个名叫 `XXXX` 的实例。与此同时，在 `%APPDATA%\Mozilla\Firefox\Profiles` 下会多出一个名叫 `XXXX` 的目录。

按照上述方式依样画葫芦，你就可以启动出任意多个 Firefox 实例。为了省事，你可以把上述命令行做成一个快捷方式，放到桌面上，免得每次都敲键盘。

另外，Firefox 还提供了一个多实例的管理界面，用如下命令可以调出该界面。

```
"{PATH}\firefox.exe" -p
```

多实例的 Chrome

Chrome 类似于 Firefox，也可以通过命令行启动新的实例。

对于 Windows 上的 Chrome，由于 Chrome 安装的时候没让选安装目录，所以其**【exe 的安装目录固定如下】**：

Windows 7 或 Vista 系统
`C:\Users\**{USER}**\AppData\Local\Google\Chrome\Application\chrome.exe`
Windows XP 系统
`C:\Documents and Settings\**{USER}**\Local Settings\Application Data\Google\Chrome\Application\chrome.exe`

上述的 `{USER}` 表示你的 Windows 用户名

找到 `chrome.exe` 之后，接下来，你需要创建一个目录，用来存放新创建的实例。比方说，你用的是这个目录： `X:\XXXX\`

然后，通过如下命令启动 Chrome，就可以创建出新实例

Vista 或 Windows 7 系统

```
C:\Users\{USER}\AppData\Local\Google\Chrome\Application\chrome.exe --user-data-dir="X:\XXXX\"
```

Windows XP 或 Win 2003 系统

```
C:\Documents and Settings\{USER}\Local Settings\Application Data\Google\Chrome\chrome.exe --user-data-dir="X:\XXXX\"
```

按照上述方式依样画葫芦，你就可以创建出任意多个 Chrome 实例。为了省事，你可以把上述命令行做成一个快捷方式，放到桌面上，免得每次都敲键盘。

◇优点

1. 解决跨站脚本（XSS）的问题

这个优点跟“多浏览器”方案是类似的。俺就不再啰嗦了。

2. 部分解决高危插件（Flash）的使用问题

这个优点跟“多浏览器”方案是类似的。俺就不再啰嗦了。

3. 灵活性高

与“多浏览器”的方案相比，“多实例”的方案明显灵活多了。因为 Firefox 和 Chrome 可以创建出任意多个实例（只要你的硬盘够大，想建几个实例都行）。而且，这个方案可以满足某些 Firefox 粉丝或 Chrome 粉丝的个人偏好。

◇缺点

1. 无法彻底解决浏览器漏洞的问题

在这方面，“多实例方案”与“多浏览器方案”具有共同的缺陷——无法抵御具有【本地代码执行】的高危漏洞。包括浏览器漏洞和插件漏洞都有可能出现这类高危漏洞。

那么，该咋办捏，请看下一节——“多用户”的方案。

★【多用户】的方案

前面提到的两种方案，都无法防范某些浏览器或插件的高危漏洞。因为这些高危漏洞会导致在本地执行攻击代码，并有可能植入木马。现在，俺来介绍第三种方案——多用户方案。此方案可以防范**大部分**在本地执行的攻击代码。

先说明一下，此处的“用户”指的是【操作系统用户】。

◇某些高危漏洞为啥很危险?

俺前面反复提到“导致本地执行攻击代码的漏洞”。这样的漏洞是非常非常危险的。为啥捏？俺简单扫盲一下。

如果你的浏览器或者浏览器插件具有这类漏洞，当你访问某个恶意网页时，如果该恶意网页能够利用此漏洞，就可以在你的本地的操作系统中执行攻击代码。由于这个攻击代码是在浏览器的进程内触发的，所以该攻击代码就具有当前用户所具有的权限。

比方说，如今大部分网友都用 Windows 上网。并且，这些网友都是以“系统管理员”的身份使用浏览器。一旦碰到这类漏洞时，本地的攻击代码也同时具有了“系统管理员权限”。有了这么高的权限，攻击代码可以为所欲为。

某些网友可能会问：那不用管理员身份上网，是不是就没事了？

俺只能说，用低权限的用户身份（比如 Windows 里面的“Users 组”）上网，相对于用管理员身份上网，会好一些。但是捏，还是有问题。

举个例子：

即使你用低权限用户上网，一旦遭遇这类漏洞，攻击代码还是有可能植入木马。然后捏，这个木马有可能会查找你电脑上的各种私密文件（比如你的裸照）。然后木马会把这些私密文件发送给木马的主人。

从上述例子可见，用低权限的用户上网，【不能】彻底解决问题。所以，俺隆重推出第三种方案——多用户方案。

◇啥是“多用户方案”捏?

如今的桌面操作系统，无论是 Windows 还是 Linux 或 Mac OS，都可以创建多个操作系统用户，并且这多个用户可以同时运行程序。如果你用过 WinXP 的快速用户切换，应该对此深有体会。

多用户方案的关键在于：

你需要创建一个或多个【低权限】的“上网用户”（所谓低权限，必须低于你当前使用的用户权限）。这些“上网用户”专门用来访问各种网站。

假使你不幸访问了恶意网页，遭遇本地执行的攻击代码，问题也不大。因为这些上网用户的权限很低，所以它们触发的攻击代码，权限也很低。因此攻击代码就比较难钻空子。一般来说，对“上网用户”的权限限制得越严格，攻击代码就越难有作为。

◇操作步骤

1. 创建上网用户

如何在桌面系统中创建新的用户，大家应该都会吧？不会的同学，请看《避免使用高权限用户》一文的相关章节。上网用户可以只创建一个，也可以创建多个。具体建几个，看你的需求。

举例：

假如你非常看重网银的安全，可以创建两个上网用户，一个专门用来访问网银，一个专门用来上其它网站。

提醒一下：在这些上网用户的环境中，除了浏览器，其它啥软件都【别】装。

2. 设置上网用户的权限

再啰嗦一次，上网用户的权限，一定要低于你目前使用的用户权限。

以 Windows 为例：

在 Windows 中，常见的用户组的权限大小分别是：**“Administrators 组” > “Power Users 组” > “Users 组” > “Guest 组”。**

如果你平时用“Administrators 组”的用户，那么可以把上网用户设置为：“Power User”或“User”或“Guest”

如果你平时用“Power Users 组”的用户，那么上网用户就只能设置为 User 或 Guest。

以此类推……

（如果你想了解这些用户组之间的权限差异，可以参考本系列的第一个帖子《避免使用高权限用户》）

提醒一下：

Guest 用户组的权限是最低，低得难以想象。所以，从安全角度而言，把上网用户设置为“Guest 组”是最安全滴。但是，也是最麻烦滴。你需要进行好多设置——否则的话，你可能连浏览器都跑不起来。

所以，除非你对 Windows 系统比较熟悉，否则的话，俺【不】建议使用“Guest 组”的权限。比较理想的权限是“Users 组”。这个组的权限也比较低，但是用起来不麻烦。

3. 设置文件目录的访问控制权限（ACL）

每个人的电脑中，都会有某几个目录是用来存放你的敏感个人资料的。

你必须设置这些目录的访问控制权限（简称 ACL），设置为：“上网用户”所在的组**【不允许读写】**。这样一来，你可以把自己那些敏感的，私密的文件（比如自己的裸照），都通过上述 ACL 保护起来。即使“上网用户”遭遇高危漏洞的攻击，本地执行的攻击代码也【难以拿到】你的隐私（俺说"难以拿到"，意思就是说，还是有可能滴，但是可能性很小，后面会解释）。

顺便提醒一下：对于 Windows 系统，只有使用 NTFS 格式的分区，才支持 ACL。FAT32 格式的分区是不支持滴！

如果你是个菜鸟，不懂得如何设置文件及目录的 ACL，没关系！Google 一下，你就知道。

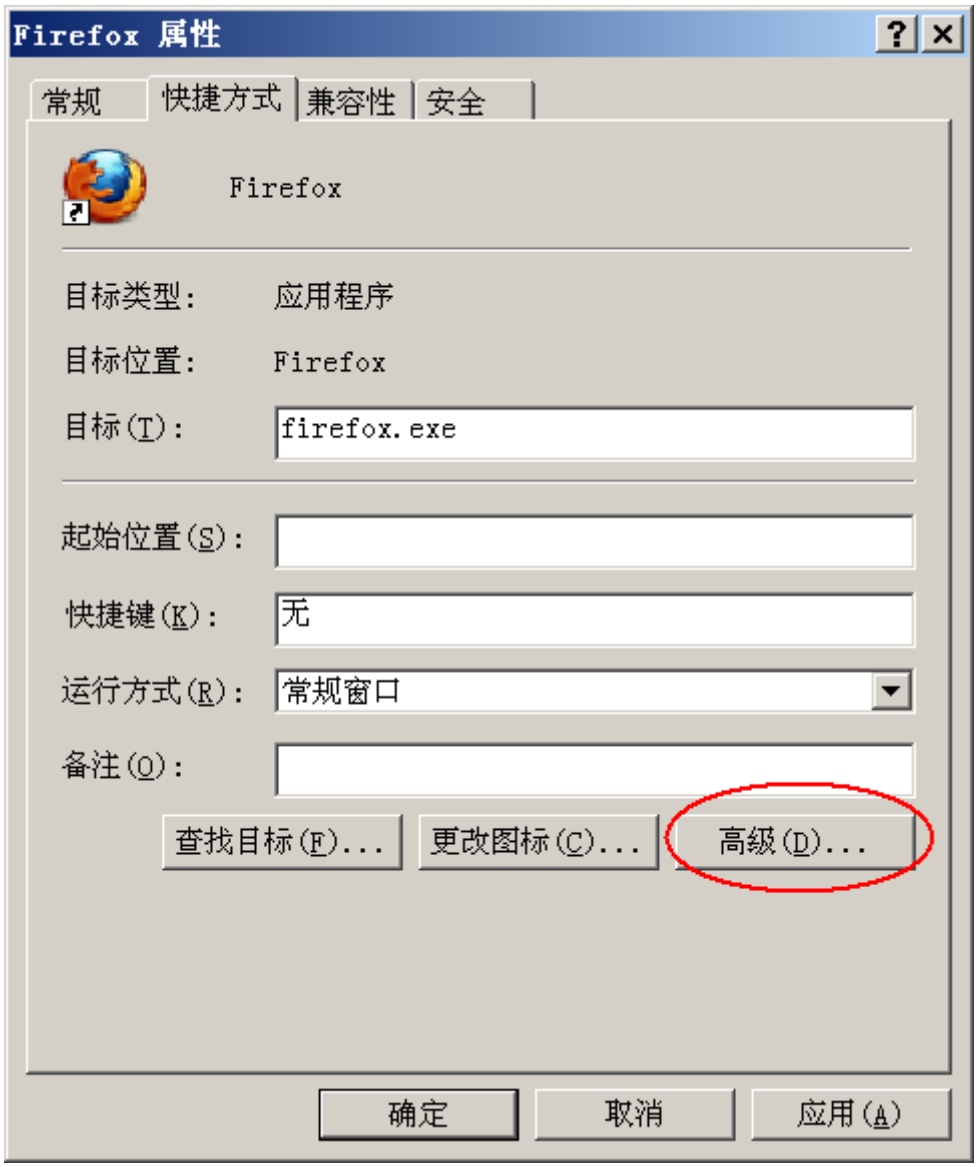
4. 多用户浏览器共享同一个桌面的技巧

你可以直接用 Windows 提供的“快速用户切换”。对于 Windows 系统，从 WinXP 开始就提供了"快速用户切换"的功能。用它来切换用户还是很方便滴。但是这招有一个缺点：你一次只能看到某一个用户运行的软件，其它用户运行的软件看不到。

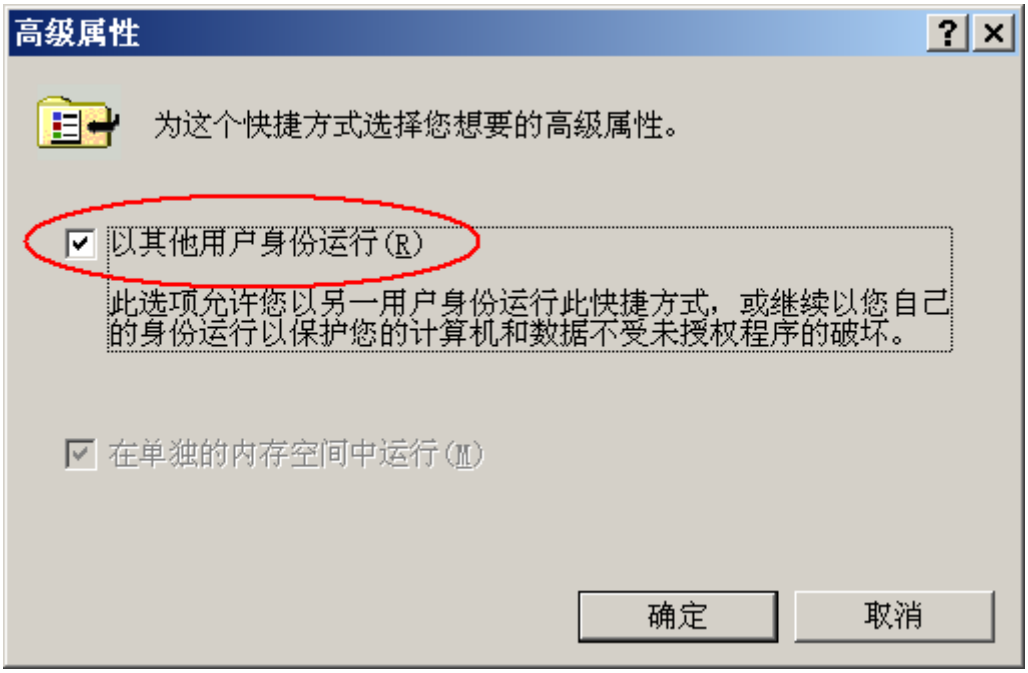
不过没关系，还有一个小技巧，可以让你在同一个桌面中，运行不同用户的软件（包括浏览器）。

假设你创建了 A & B 两个用户。其中 A 是主用户，用来完成你日常的工作；B 是上网用户。那么你可以先登录“A用户”，然后在“A用户”的桌面上创建一个浏览器的快捷方式。（如何创建快捷方式，就不用俺手把手教了吧）

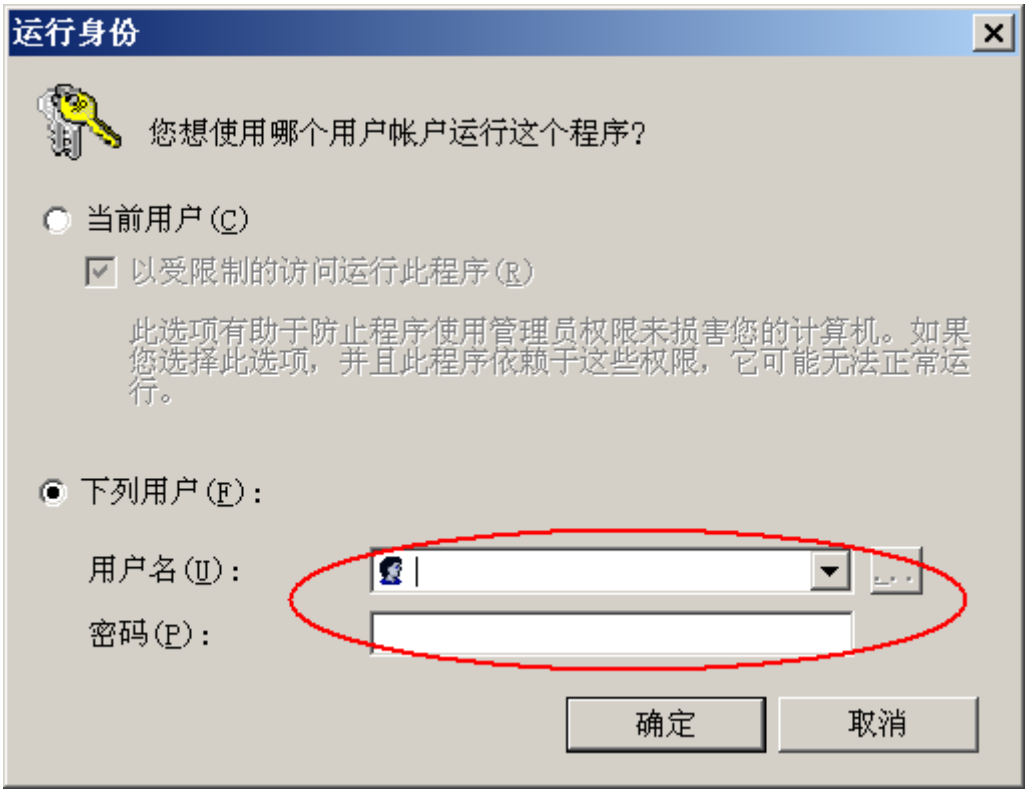
用鼠标选中该快捷方式，在快捷菜单（右键菜单）中，选择“属性”菜单项。出现如下对话框。



在该对话框中，点“高级”按钮。出现如下对话框。把“以其他用户身份运行”选项打勾，就可以啦。



之后，如果你想在“A用户”的桌面上运行“B用户”的浏览器，只需点击该快捷方式，就会弹出如下对话框。然后输入“B用户”的用户名/口令，就能以“B用户”的身份运行浏览器。



刚才介绍的是图形界面的配置。对于习惯于命令行的 IT 专业人员，还可以用命令行的方式启动指定用户的某个进程。Linux 系统和 Mac OS 系统有 `su` 或 `sudo` 命令；Windows 系统有 `runas` 命令。

◇优点

1. 解决跨站脚本（XSS）的问题

这个优点跟前两种方案类似，俺就不再啰嗦了。

2. 解决高危插件（Flash）的使用问题

这个优点跟前两种方案类似，俺就不再啰嗦了。

3. 防范各种浏览器漏洞

正如刚才提到的——前两种方案（多浏览器、多实例）无法防御浏览器及插件的某些高危漏洞（具备本地执行攻击代码的漏洞）。而“多用户方案”可以大大降低这类漏洞的危害性。使得攻击代码只能威胁到“上网用户”本身，而不会威胁到其它操作系统用户。

◇缺点

1. 初始配置稍嫌麻烦

相比前面两种方案，这个方案的【初始配置】比较麻烦。而且你要分清楚哪个用户是用来干啥的。不过捏，一旦用久了，习惯了，也就不觉得麻烦了。

2. 无法防范极个别高明的攻击者

“多用户方案”之所以可以隔离攻击代码，是因为如今所有主流的桌面操作系统，都能够在操作系统层面，对不同的系统用户进行隔离。因为有操作系统层面的隔离，所以才限制了攻击代码的危害性。

但是，操作系统层面的隔离，也不是百分之百可靠滴。不要忘了，操作系统本身也是软件，也可能出现安全漏洞。在操作系统的安全漏洞中，有一类漏洞叫做“权限提升漏洞”（简称“提权漏洞”）。所谓的“提权漏洞”，顾名思义，就是能够提升【当前执行代码】的权限。比方说，本来攻击代码没有管理员权限，通过"提权漏洞"，就可以拿到管理员权限。

如果你的操作系统本身存在“提权漏洞”，同时你的浏览器或者插件存在“能够在本地执行代码的漏洞”，那么，高明的攻击者就有可能把这两者组合起来，对你的系统进行组合攻击。

不过大伙儿别担心——要实现这类组合攻击，需要同时掌握【未公开的】浏览器或插件漏洞，并且还要有【未公开的】操作系统提权漏洞（俺在前面的帖子里介绍过，“未公开漏洞”总是比“零日漏洞”更危险）。另外，攻击者还需要做很多准备工作，才能诱使你访问到恶意网页。一般的入侵者根本没有这个本事，也没有耐心去搞这些。反之，有这个本事又有这个耐心的入侵者，通常不会拿这种招数去入侵普通网友（这么做简直是大材小用）。所以，如果你只是一个普通的网友，在前面三个方案中挑选一个，即可。

估计有些好奇的同学会打听了。什么样的入侵者具有这种实力？什么样的人会成为他们的目标？

由于本文的篇幅已经很长了，俺稍微调一下列位看官的胃口，在《Web相关的防范（下）》再来八卦这些，同时也介绍一下比“多用户方案”更高级的方案。虽然这种方案很多人用不到，但俺还是会写出来，就当满足一下大伙儿的好奇心 :-)

[如何防止黑客入侵7]：Web相关的防范（下）

文章目录

在本系列前一篇，俺介绍了三种隔离浏览器的方式（多种浏览器、同种浏览器多实例、多操作系统用户）。今天继续介绍第四种隔离方式——虚拟机，然后再推荐一些浏览器的安全扩展。

★【多虚拟机】的方案

◇什么是“虚拟机”?

本文提到的“虚拟机”，全称是“操作系统虚拟机”。

最近10年来，硬件水平显著提升，操作系统虚拟化的技术开始普及，出现了若干针对操作系统的虚拟化软件。这种软件可以让你在一台电脑上，同时运行**【多个操作系统】**（是不是很有趣？）。通过虚拟化软件来运行的操作系统，称之为“虚拟操作系统”；与之对应，你原先的操作系统称之为：真实操作系统或宿主操作系统。

由于“虚拟操作系统”是虚拟出来滴，你可以在里面为所欲为，而【不会】对真实操作系统产生实质性的影响。比方说，你可以在虚拟系统中把硬盘格式化，但不会影响到你的真实系统。同样的，如果某个虚拟系统被病毒感染了，也不会影响真实系统和其它虚拟系统。

◇什么是“多虚拟机”的方案?

所谓“多虚拟机”的方案，就是在你的电脑上创建多个虚拟机，分别用来实现【不同安全级别】的上网行为。

举个例子：

你可以创建虚拟机A，只用来访问网银（不访问其它网站）；然后创建虚拟机B，用来进行其它上网行为。那么，即使你在虚拟机B受到攻击，对虚拟机A也完全没有影响。这样一来，就可以彻底保证网银的安全。

◇为啥要用“多虚拟机”的方案?

前一篇博文提到的三种隔离方案，“多用户”比前两种方案安全。这种方案是基于操作系统提供的用户壁垒——包括：不同用户的进程隔离性、文件系统的访问控制（ACL）、等等。

但是“多用户”方案还是有缺陷的。如果攻击者同时利用了未公开的浏览器漏洞和未公开的操作系统【提权】漏洞，就有可能攻破操作系统的用户壁垒。不过大伙儿别担心，实现这种攻击的难度比较大，只有足够牛B的入侵者能够做到这点。

为了满足列位看官的好奇心，稍微介绍一下所谓的牛逼黑客，大都是哪些人。

御用骇客/御用高手

所谓的“御用高手/御用骇客”，也就是官方资助的入侵者（类似于武侠小说中的大内高手）。这种类型的入侵者，往往不是一个人单干独斗，而是一个团队群殴。

“御用高手”的目标大致有如下：

1. 军事目标
   军事目标主要有：外国重要的政府机构（比如：五角大楼）、外国重要的军工企业（比如：洛克希德-马丁公司）
   俺在上一篇提到了 RSA 被入侵的案例：攻击者首先利用零日漏洞入侵某个 RSA 公司的雇员，经过深度渗透之后，入侵者搞到了 RSA 动态令牌产品（SecureID）的种子（种子是用来生成动态口令的）。由于 SecureID 产品被许多大公司采用，所以入侵者可以利用偷来的种子算出动态口令，进而实现对美国多家大型军工企业的入侵。
   计划得如此严密的系列入侵，通常只有御用黑客团队能够干得出来。
2. 经济目标
   所谓的经济目标，主要都是国外知名的大公司。
   通过入侵这些公司，可以窃取商业机密，从而获得巨大的经济效益。
3. 政治目标
   政治目标就比较杂，比如：知名的反共网站、某些知名的政治异议人士的电脑/手机/邮箱/IM、等等。
   举个例子：
   2009年底，Google 被来自中国大陆的攻击者深度渗透（这就是传说中的“极光行动”）。此事直

接导致 Google 愤而退出中国市场。根据事后分析，入侵者的注意力集中在某些 Gmail 邮箱的内容。而这些 Gmail 邮箱恰恰属于中国的持不同政见者。由此可见，入侵 Google 的人很可能是朝廷的走狗。

民间高手

除了御用黑客，也不排除民间有高手。甚至不排除某个御用黑客在业余时间干点脏活。

和御用高手不同，民间高手的目标相对比较单一，大部分人是为了获取经济利益。

综上所述，能被这2类人盯上的，往往是高价值目标。如果你只是一个普通网民，不用担心被高手盯上（也就是说，“多用户”的方案基本上可以满足你的安全需求）；反之，如果你自认为是一个高价值的目标，或者你对安全的要求非常非常高，不妨尝试一下“多虚拟机”的方案。

◇如何操作?

刚才已经说了“多虚拟机”的原理，还举了例子。

如果你已熟悉“虚拟化软件”（比如：VMware 系列、VirtualBox、等）的使用，那么本方案对你来说其实很简单——无非就是安装若干个虚拟操作系统，然后在虚拟系统中安装软件，仅此而已。

由于本文的重点是防范黑客入侵，所以俺就不再介绍虚拟化软件本身的安装和配置。**关于虚拟化软件的扫盲（包括原理、安装、配置、使用），俺已经另写了一个系列（在“这里”）。**

◇优点

在4种浏览器隔离方案中，多虚拟机的安全性最高。即使你的某个虚拟机被病毒感染或者被植入木马，也几乎不会影响到你的其它虚拟机和真实系统。

当然，绝对的安全是不存在滴。虚拟化软件也是软件，只要是软件，就有可能出现漏洞，只要出现漏洞，就有可能被利用。但是，想通过虚拟化软件的漏洞来突破虚拟机的壁垒，难度更大（远远大于突破操作系统的用户壁垒）。这方面的技术细节，说来话长，俺就不展开了。

◇缺点

对硬件的要求高（主要是物理内存和 CPU）。

具体要多高的硬件配置，取决于你**同时**开几个虚拟机。**同时运行**的虚拟机越多，就需要越大的物理内存和越多的CPU核心。

★使用浏览器的【安全扩展】

终于把浏览器的话题说完了。接下来再介绍几款安全方面的浏览器扩展。这些扩展可以帮你提高 Web 的安全性。

◇NoScript——同时支持 Firefox & Chrome

简介

NoScript 是一个名气很大、功能很强的 Firefox 扩展，主页在“这里”。

通过它，你可以定制网站白名单。只有当你浏览白名单内的网站时，才启用浏览器的 JavaScript 脚本功能和插件功能（比如 Flash 插件、Java 插件、PDF 插件、等）。

白名单只是它的功能之一，更多的功能介绍，请看它的主页。

局限性

这个扩展可以有效地避免【陌生网站】上的挂马。但是，如果你【经常访问】的网站出现跨站脚本的漏洞，NoScript 有可能帮不了你。

在前面的博文中，俺曾经举了一个跨站脚本攻击的例子。比如说，你经常上某个 BBS，并且该 BBS 的界面功能依赖于 JavaScript。那么，你就必须把这个 BBS 站点加入到 NoScript 的白名单中。假如这个 BBS 本身出现了基于 JS 的跨站脚本漏洞，那你还是有可能中招 :(

◇NotScripts 与 ScriptNo——专用于 Chrome

简介

某些用 Chrome 的同学，如果不喜欢刚才提到的 NoScript，可以考虑 Chrome 上的另外两款扩展——跟 NoScript 很像（不但功能很像，连名称也很像）。

一款叫做 NotScripts，主页在“这里”；另一款叫 ScriptNo，主页在“这里”。

NotScripts 的用户数比 ScriptNo 略多。至于要选哪个，请大伙儿自行判断。

局限性

这两款扩展的局限性类似于前一个小节提到的 NoScript，俺就不再啰嗦了。

◇HTTPS Everywhere

这是著名的电子前线组织（EFF）发布的扩展，主页在“这里”，同时支持 Firefox 和 Chrome。说到 EFF，顺便提一下：Tor & TorBrowser 也是该组织发布的产品。

如今，有很多网站都同时提供明文的 HTTP 协议和加密的 HTTPS 协议（比如维基百科）。装了 HTTPS Everywhere 扩展之后，如果你浏览的网站支持 HTTPS 协议，该扩展就会强制浏览器通过 HTTPS 协议访问该网站。从技术上讲，就是把所有针对该网站的 HTTP 请求都转换为 HTTPS 请求。

为啥要强制用 HTTPS 协议捏？因为 HTTPS 是加密协议，可以保护你免受入侵者的嗅探（关于“嗅探”的案例，前面的博文提到过）。

除了上述功能，HTTPS Everywhere 扩展还可以帮你侦测有问题的 CA 证书，降低“中间人攻击”（MITM）的风险。

引申阅读：

关于 CA 证书的扫盲，参见《数字证书及 CA 的扫盲介绍》。

局限性

如果某个网站只有 HTTP 连接，不提供 HTTPS 连接，那 HTTPS Everywhere 也帮不了你。

◇LastPass

简介

LastPass 名气最大的在线口令管理工具。官网在“这里”，维基百科的介绍在“这里”。

该工具提供了针对所有主流浏览器的扩展（包括IE、Firefox、Chrome、Opera、Safari、等），帮你自动填写网站的登录口令，免除你记忆诸多口令的麻烦。你本人只需要记住一个【主密码】，LastPass 会利用主密码来加密本地的密码数据库——你的其它口令都存在在该数据库中。

为了确保安全性，LastPass 进行在线同步时，传输的是加密后的数据库。因此，即使 LastPass 网站被黑，入侵者拿到的也只是加密后的用户口令数据库。同样的，如果有人偷了你的电脑，但不知道你的主密码，也无法打开你的密码数据库。

局限性

LastPass 的做法相当于把鸡蛋都放在一个篮子里，有好处也有坏处。

最大的风险在于主密码被盗。一旦主密码被盗，密码数据库中的所有密码就都暴露了。什么情况下会发生主密码被盗捏？比如你的电脑被植入木马，并且此木马具有键盘记录的功能；比如你输入主密码的

时候，有人在旁边偷窥；……

每个人都有很多密码。不过根据二八原理，真正重要的密码不到20%，大部分密码都不太重要。所以俺个人的建议是：少数特别重要的密码，还是靠自己脑子来记；大多数不太重要的密码，可以交给类似 LastPass 之类的口令管理软件。

◇BetterPrivacy

简介

BetterPrivacy 是一个侧重于隐私保护的 Firefox 扩展，主页在“这里”。

当你浏览某些网站的时候，网站可能会在你的电脑上记录 cookie。通过这些 cookie，网站可以追踪你的上网行为（比如你多久访问一次这个网站）。

有了 BetterPrivacy，你就可以配置允许哪些网站记录 cookie。BetterPrivacy 的牛B之处在于：它不光可以控制传统的 cookie，还可以控制 Flash 的 cookie（LSO）。

局限性

此扩展只针对隐私保护，无法防范扩展脚本等攻击。

浏览器的安全类扩展，细分为很多领域，数量也很多。限于篇幅，俺仅挑选出每个领域最出名的代表。如果你还有补充的，可以到本文留言。

★结尾

关于 Web 的安全防范，本来只想写一篇。谁曾想，东扯西扯，居然写了三篇。可能有同学会问，为啥没提到杀毒软件和个人防火墙？俺觉得：把这两个话题放到 Web 安全防范中聊，不太合适——还是单独拿出来聊比较好。在本系列后续的博文，会说说杀毒软件和个人防火墙的那些事儿。

[如何防止黑客入侵8]：物理隔离的几种玩法

文章目录

★先插播一个安全通告

11月29日曝光了一个 Firefox 的高危漏洞，详情在“这里”（洋文）。

根据漏洞描述，此漏洞只影响 Windows 平台。照理说俺用的是 Linux，应该不受影响。但考虑到俺是高危险人士，为了保险起见，还是先静默几天——暂不使用“编程随想”这个身份进行网络活动。俗话说得好：【小心驶得万年船】。

本来俺计划在11月底发一篇博文谈 Linux，但是这个漏洞让俺改变主意——所以今天这篇来聊聊“物理隔离”的话题。

（本文发出后，俺已经到博客后台管理界面，把这几天被 Google 误判为垃圾广告的评论，全都恢复出来了）

★本文的目标读者是哪些人？

为了避免浪费大伙儿的时间，俺先声明一下本文的目标读者。

这篇博文是面向那些【对安全性要求非常高】的用户。对这些人而言，“虚拟机隔离”还是无法令他们放心。

既然聊到这里，顺便解释一下：什么情况下“虚拟机隔离”会出问题。

如果虚拟化软件（比如 VirtualBox、VMware、KVM ...）本身出现了安全漏洞，并且这个安全漏洞会导致虚拟机被穿透；然后，Guest OS 正好又感染了恶意软件；而相关恶意代码正好又能够利用这个漏洞进行穿透；那么，该恶意代码就有可能从 Guest OS 入侵 Host OS。一旦恶意代码能够侵入 Host OS，理论上它就可以访问这个 Host OS 上的所有 Guest OS。在这种情况下，“虚拟机隔离”的措施就【失效】了。

出现上述情况的概率是非常低的——需要好几个条件【同时】具备。因此，能够进行这种攻击的，必然是比较高级的入侵者，而且这个入侵者对入侵对象必须有深入的了解：

知道对方用的是什么类型的虚拟化软件（不同的虚拟化软件，其安全漏洞的情况全然不同），
知道对方用的 Guest OS 是什么类型的系统
知道对方用的 Host OS 是什么类型的系统

所以，普通的网民不需要考虑这种风险。为啥捏？能这么搞的入侵者都是比较高级的，他们根本不屑于去入侵普通网民。

★预备知识：关于“隔离性”和“攻击面”

◇几种不同的隔离级别

在本系列的前面3篇，俺连续聊了很多关于 Web 方面的防范措施。你看完这3篇之后应该会发现：俺提及的防范措施，全都是围绕“隔离”这个概念来展开的。不论是“多实例”还是“多用户”或者“多虚拟机”，说白了都是为了隔离“浏览器环境”。

“隔离”的好处在于：一旦某个环境被入侵，（只要你的隔离屏障没有被穿透）别的环境不会受影响。说到“隔离屏障被穿透”，自然就引出“隔离性的好坏”这个话题。

不同的“隔离措施”，其“隔离性”是不同的。“隔离性”越好，攻击者就越难穿透。

【常见的】隔离措施（从低到高）有如下几种。俺结合浏览器来加以说明

1. 进程级

（以浏览器为例）搞多个“浏览器实例”。不同的实例肯定是不同进程，但都在同一个用户下。

2. 用户级

（以浏览器为例）在不同的【普通】用户下分别运行浏览器。但都在同一个操作系统下。

3. 虚拟系统级

（以浏览器为例）创建多个虚拟系统（Guest OS），在不同的虚拟系统中分别运行浏览器。虽然处于不同的 Guest OS，但还是在同一个 Host OS 下。

4. 物理系统级

（以浏览器为例）在不同的物理系统中运行浏览器。

今天俺要介绍的就是最高级的“物理系统级的隔离”。

◇“物理隔离”的3种子类型

“物理隔离”还可以继续细分为3种子类型（这三种级别也是从低到高）：

物理隔离1型

多个物理系统之间【存在】网络连接。

物理隔离2型

多个物理系统之间【没有】网络连接，但【存在】存储介质的交换（比如：通过“U盘”交换数据）。

物理隔离3型

多个物理系统之间既【没有】网络连接，也【没有】存储介质的交换。

◇攻击面

关于这个概念，洋文称之为“attack surface”，相关的英文维基百科页面在“这里”。考虑到很多同学不喜欢看洋文，下面俺通俗地扫盲一下。

为了描述这个概念，暂且借用一下军事术语。在军事领域中，如果其它因素都一样，则防线越长就越容易被突破。

在信息安全领域，道理也类似。军事领域的“防线”就类似于本文所说的“攻击面”。

◇“攻击面”的2个维度

从概念上讲，“攻击面”可以再分成2个维度，分别是“时间维度”和“空间维度”。为了便于理解，下面俺举例说明。

时间维度

有两台软硬件配置一模一样电脑（A 和 B）。A 是一年到头365天都挂在网上，而 B 在一年里面只联网1分钟。很显然，A 被入侵的概率要大大高于 B。

空间维度

有两台服务器 A 和 B，都是始终联网的（时间维度一样）。A 上面【没有】装任何服务端软件；而 B 上面既装了 Apache（web 服务）又装了 MySQL（数据库）。而且这几个服务的端口都对外开放。假如某天，Apache 曝光了高危漏洞，B 就可能因此而被入侵，但是 A 就没事儿。

【通俗地说】，系统中装的软件越多，则包含的“代码”就越多，那么“代码”中出现安全漏洞的概率也就越大。

（上面这句只是“通俗”的说法。严格来讲，“攻击面”与“代码量”并【不是】简单的线性关系。这两者之间的关系牵扯的因素很多，比如:软件代码的质量，软件模块的类型、各个软件模块之间的依赖关系、开源 or 闭源 ……）

★招数1：把不同的上网帐号放到不同的物理系统

这个招数是最容易想到的，所以俺先聊这招。

如今很多人都有不止一台 PC。但是 PC 再多也不会超过10台吧（除非你是开网吧的）。但是你的上网帐号很可能不止10个。所以，你不太可能给每一个上网帐号配一台单独的 PC。这就需要考虑一个“划分策略”：哪些帐号共用一台 PC。

“划分策略”通常有如下两种：

◇根据“重要程度”进行划分

比如说，你搞两台 PC，一台专门用来操作很重要的网络帐号，另一台用来操作不那么重要的网络帐号。

对于前者，安装靠谱的操作系统（Linux > Mac OS > Windows），装的软件要尽量【少】（减少“攻击面”），并采取尽可能多的措施来保护它（具体措施参见本系列前面几篇）。

◇根据“身份”划分

如果你对隐匿性的要求很高，可以考虑以“身份”划分上网帐号到不同的 PC。

为了便于理解，拿俺自个儿来举例：

显然，俺是非常看重“隐匿性”滴——如果让朝廷方面知道俺的真实身份，俺就废了。所以，俺不光注重【安全性】的需求（防止被入侵），更加要注重【隐匿性】的需求（防止俺的“虚拟身份”与“真实身份”被关联起来）。

如果俺仅仅使用“虚拟机隔离”（一个虚拟机用来操作“真实身份”的帐号，另一个虚拟机用来操作“编程随想这个身份”的帐号）。万一碰到“虚拟机穿透”（前面提到过），入侵者通过 Guest OS 进入 Host OS，就可以同时访问到这两个 Guest OS，就有可能发现：原来“编程随想”就是某某人。

反之，如果把这两个身份的帐号放到两台 PC（物理隔离）。即使“编程随想”这个身份对应的 PC 被彻底入侵，入侵者看到的帐号全部都是“编程随想”相关的帐号，不会看到“真实身份的帐号”（因为这些帐号在另一台 PC 上）。

★招数2：在【专用的】物理系统中操作“重置密码的邮箱”

这个招数很简单，仅仅从章节标题，你基本上就能猜到是怎么玩的。所以俺就节省点口水，不展开了。

下面简单说说这个招数的注意事项：

\1. 用来“重置密码”的邮箱，最好是专用的——也就是说：除了“重置密码”，不再作其它用途（以降低“攻击面”）

\2. 操作这个邮箱的 PC，最好也是专用的（以降低“攻击面”）

\3. 这台电脑只要很低的硬件配置就可以了。所以，你可以拿以前淘汰的旧电脑来用。

★招数3：在【专用的】物理系统中运行密码生成器

可能很多同学是第一次听说“密码生成器”这个玩意儿。所以俺来解释一下：

在本系列的第3篇《如何构造安全的口令/密码》中，俺介绍了好几种设置密码的技巧，那篇教程的最后一招就是“用散列算法构造密码”。

“用散列算法构造密码”，有“简单用法”，也有“高级用法”（对这两种，那篇博文都有介绍）。由于本文针对的是“安全性要求很高的读者”，所以你当然要用“高级用法”。“高级用法”通常需要依靠一个小程序/小脚本来辅助你进行 N 迭代（计算 N 次散列）。这个“小程序/小脚本”就是本章节所说的“密码生成器”。

有了“密码生成器”，你可以轻松构造出许许多多超长密码（可以长达几百个字符，前提是：网站要支持这么长的密码）。而且你【不需要】花很多脑细胞来记住这些变态的长密码。你只需记住自己的“种子串”以及“迭代次数”。如果你把“种子串”及“迭代次数”写死在生成器的脚本中，那你甚至连这两个信息也【不用】记忆。

这台运行“密码生成器”的电脑完全【不用联网】。所以这台 PC 的隔离属于前面提到的“物理隔离3型”——这是物理隔离中最严格的（没有之一）。

★招数4：对重要帐号划分“操作机”和“登录机”

这个招数与前面几个完全不同——是采用【2台】PC 来保护【单个】帐号。通常而言，只有很重要的帐号才值得你动用两台物理电脑来保护。

这个招数相对比较难懂，俺需要多费点口水。在开讲之前，先来解释一下什么是“记住登录状态”的功能。

◇“记住登录状态”功能

如今大部分网络帐号都有【记住登录状态】这个功能。当你登录该帐号，输入完密码后，如果勾选了“记住登录状态”这个选项（不同的网站，界面上的叫法可能略有不同），那么你下次访问该帐号就【不用】再输入密码了。

◇“记住登录状态”的技术原理

网站是如何做到这个效果的捏？通常是利用浏览器 cookie。当你输入密码并登录成功之后，会在 cookie 中保存一个“安全令牌”（洋文叫“token”）。这个“安全令牌”不是密码，但是比密码更复杂更难暴力破解。而且，**根据令牌是【无法】逆向推算出密码的**（只要是比较靠谱的网站，技术上肯定能保证这点）。

当你下一次又访问了这个网站，浏览器会把 cookie 中的这个令牌发送到网站服务器，服务器端会采用某种安全可靠的算法来验证这个令牌是否合法，如果合法，就认为你是之前登录过的那个用户。

由于 cookie 有时间限制（期限），过了期限这个 cookie 就失效了。这时候你如果又访问这个网站，它又会要求你输入密码。另外，如果你在浏览器中清空了所有 cookie，那么下一次访问该网站也要重新输入密码。

◇物理隔离的玩法

明白了这个功能的原理之后，开始来说“物理隔离”怎么玩。

你需要专门准备2台 PC——“登录机”和“操作机”。记住：这2台电脑都是用来操作【同一个】帐号滴。

每次需要登录该帐号，你一定要在“登录机”上输入密码。登录成功之后，把登录机上的【cookie 文件】复制到“操作机”上。然后你就可以在“操作机”上【免登录】访问该帐号。

（由于需要把 cookie 从“登录机” copy 到“操作机”。所以这种隔离属于前面提到的“物理隔离2型”）

◇这个招数的优点

首先，因为你从来【不】在“操作机”上输入该帐号的密码。所以，即使“操作机”被植入木马，木马也不可能知道该帐号的密码是啥。

从“时间维度”而言

由于“登录机”仅仅用来输入密码，其它的日常操作都在“操作机”。所以，（从“时间维度”上讲）“登录机”的攻击面远远小于“操作机”。

从“空间维度”而言

由于这个“登录机”仅仅用来输入密码。你只需安装非常少的几个软件（比如 浏览器 之类的）。如果你的帐号名称是洋文（大部分应该都是吧），这台“登录机”甚至连输入法都【不用装】。

所以，从“空间维度”而言，“登录机”的攻击面也会远远小于“操作机”。

综上所述，入侵者很难搞定“登录机”。

然后，前面俺又说了，根据（cookie 中的）“安全令牌”是无法逆向推算出帐号密码滴。所以，即使“操作机”被【彻底】入侵了，入侵者虽然可以进入你的帐号，但是他/她依然无法知道你的密码（因此也就无法改密码）。

这种情况下，你可以很轻松地把入侵者踢掉——只需通过“登录机”进入帐号，修改密码；然后入侵者在“操作机”上就进不了这个帐号了（一旦修改过密码，原有的安全令牌就作废了）。

★招数5：“物理隔离”搭配“多重代理”

◇“物理电脑”与“虚拟机”在“网络隔离”方面的差异

俺博客上有一个很受欢迎的系列叫做《如何隐藏你的踪迹，避免跨省追捕》。在这个系列中，俺介绍“多重代理”，也介绍了“用虚拟机防止公网 IP 暴露”。

当初俺写那篇《用虚拟机隐匿公网 IP》的时候，好几个懂技术的读者跳出来质疑。因为他们光看标题没看内容 :)

所谓的“用虚拟机防止公网 IP 暴露”，是因为：可以利用虚拟网卡的不同模式（NAT 及 Host-Only）来设置网络隔离，使得 Guest OS 中的软件【无法直接联网】（强迫这些软件走代理）。这样就可以避免：“因为某些软件【直连网络】导致的公网 IP 暴露”。

但如果出现“虚拟机被穿透”，那“公网 IP”还是有暴露的风险。因为虚拟网卡只能用来限制 Guest OS，而无法限制 Host OS。而 Host OS 中的软件通常是可以直连公网的。当某个恶意代码从 Guest OS 穿透进入 Host OS，该恶意代码就有可能尝试“网络直连”，并导致你的公网 IP 暴露。

看到这里，聪明的读者已经猜到解决方案了。那就是把原先“单虚拟机方案 或 双虚拟机方案”中那个“隔离的虚拟机”换成“隔离的物理电脑”。

话虽这么说，但操作起来会有点复杂。

原先的“虚拟机方案”，你只需设置虚拟网卡的“网卡模式”（设置成 Host-Only 或 Internal），就可以把 Guest OS 的网络访问隔离起来。但是物理电脑是【没有】虚拟网卡的，物理电脑是物理网卡。而“物理网卡”是【没有】“网卡模式”让你修改的。所以你需要在网络拓扑上下点功夫。具体的玩法有很多种，考虑到本文的篇幅有限，俺只介绍其中一种。

◇“物理电脑”进行“网络隔离”的玩法（方法之一）

下面俺【简单】介绍一下步骤（考虑到本文面向的是安全要求很高的读者。既然你的安全要求很高，当然不能是技术菜鸟，肚子里要有点货。所以，俺在下面只提几个要点，操作的细节靠你自己摸索了）

1.
准备两台物理电脑，分别称为“隔离机”和“网关机”。
“隔离机”要有一个【有线】的网卡；“网关机”要有2个网卡，至少有一个是【有线】的（用来与“隔离机”对接）
为啥俺要强调“有线网卡”，因为有线网卡才能确保其只连接到一个对端；而无线网卡就难说了。
所以隔离机只能保留仅有的一个有线网卡。如果隔离机有多余的无线网卡，在 BIOS 里面禁掉。

2.
“隔离机”上运行的软件是：
有可能遭遇高级入侵的软件（比如你的浏览器），或者是那些本身就不太安分的软件（比如 QQ）
“网关机”上运行的软件是：
TOR
TOR 的前置代理（如果你在墙外，可以不用前置代理。但是有前置代理会更安全——“双重代理”优于“单重代理”）
防火墙

3.
前面说了，“网关机”至少有一个有线网卡，用来对接“隔离机”。
因此，你需要配置“网关机”的防火墙，使得【在这块网卡上】只允许 TOR 的监听端口对外暴露（这条一定设置）
另一块网卡是用来接入公网的，你可以配置防火墙，禁止该网卡上的任何监听端口（这条可设可不设，设置了更保险一些）

4.
用一根网线把这2台电脑各自的【有线网卡】对接起来。

5.
设置“隔离机”上的软件，让它们以“网关机”的 TOR 作为代理。

6.
设置“网关机”上的 TOR，让它通过“网关机”上的前置代理联网。

★总结

几个主要招数聊得差不多了。俺在最后提醒一下：
\1. 本文的这几个招数，相互之间是可以【组合使用】的
\2. 本文的这几个招数是可以跟“虚拟机隔离”结合起来用的
\3. 组合得越复杂，需要的物理电脑也越多 :)
\4. 如今的 PC 已经很便宜啦（两三千就可以买到硬件配置还凑合的）。如果你是“高危人士”或“高价值目标”，【不要】吝啬那几千块钱！

[扫盲“社会工程学”0]：基本常识

文章目录

最近几年，信息安全方面的问题日益严重，许多同学深受其害（比如网络钓鱼、盗用银行卡、蠕虫木马泛滥、僵尸网络盛行等等）。俺窃以为，很大一部分原因在于相应的扫盲教育没有跟上。且不说普通的电脑菜鸟对信息安全一无所知，即便是很多 IT 公司的专业技术人员，对此也知之甚少。其后果就是：很多菜鸟级的攻击手法屡试不爽，很多平庸的攻击者屡屡得手。有鉴于此，俺打算抽空普及一下信息安全相关的东东，或许能对某些同学有所帮助。

其实信息安全方面的话题非常之多，俺经过左思右想之后，决定先拿社会工程学来扫盲一下。至于为啥先说它，后面会解释原因。

★社会工程学是啥玩意儿?

俺喜欢把信息安全分为【硬安全】和【软安全】两部分。所谓“硬安全”主要包括具体的 IT 安全技术（比如：防火墙、入侵检测、漏洞扫描、拒绝服务攻击、缓冲区溢出攻击 ……）；而“软安全”主要涉及管理、心理学、文化、人际交往等方面，与具体的 IT 技术无关。今天所说的社会工程学，实际上就是“软安全”的范畴。

通俗地说，社会工程就是：攻击者利用【**人**】自身的弱点（往往是心理学层面）来获取信息、影响他人，从而达到自己不可告人的目的。光这么说稍显简单，更详细的定义可以参见“这里”。不懂洋文的同学可以看“这里”。

★为啥要了解社会工程学?

开头已经提到了安全基础知识的普及度不够。那为啥俺要先介绍社会工程学捏？主要有如下几点原因：

◇普及度不够

首先，社会工程是信息安全中一个经常被忽视的偏僻角落。即便很多 IT 安全领域的从业人员，往往也缺少社会工程学的相关常识。比如很多人都知道什么是防火墙、杀毒软件，但却从来没有听说过“社会工程学”这个词。

◇重视不够

大部分的安全厂商都把注意力集中在“硬安全”方面（比如现在防火墙厂商、杀毒厂商多如牛毛），很少有安全厂商把社会工程挂在嘴边的。以此**相反**的是：现有的信息安全攻击，大都以“软安全”作为攻击者的突破口，只有一小部分是纯粹通过“硬安全”来进行的。（这又是一个二八原理的生动例子）

为啥攻击者喜欢从“软安全”层面进行突破捏？因为人性的弱点是很难在短时间内得到改善的（尤其是

人多大公司、大机构，更是如此）。所以，“软安全”方面会遗留很多可以利用的漏洞，攻击者只要善于利用这些漏洞，就可以轻易侵入。

◇用处大大滴

不过捏，光是鲜为人知、重视不足，还不至于让俺花这么多口水大力忽悠。还有另一个原因是：社会工程学的常识非常有用，而且它的用处不限于信息产业（几乎所有行业都用得着）。具体有些啥用处捏？

首先，了解起码的社会工程学常识能够让你对相关的攻击手法（具体参见“这里”和“这里”）有**基本的**防范，不至于轻易上当。要知道，有很多人被攻击者利用了之后，自己还浑然不觉。

其次，如果你是公司的老板或者某个管理层的头头，你可以在自己的职权范围内进行相关的扫盲培训（后面的帖子会介绍如何防范）。

最后，假如你看完本系列后，发现自己在社会工程方面很有天赋，那或许可以考虑朝这个方向发展。比如搞个商业间谍之类的工作干干，没准也很有前途哦。不过捏，一旦将来被抓被关、被杀被剐，本博主是概不负责滴 **:-)**

★本系列帖子能给你啥帮助？

如果你从来没有听说过社会工程学，仅仅想扫盲，那只需看本帖即可，后续的内容无需多看。
如果你希望对社会工程攻击能够有**基本的**防范，建议看看后续的“攻击手法”、“如何防范”。
如果你对社会工程学这门学问很有兴趣，建议看完本系列所有帖子。
如果你已经是社会工程学的老手，请不吝赐教，本系列帖子您老就不用看了。

★本系列帖子不能给你哪些帮助？

本系列帖子【**不能**】帮你成为社会工程学的高手。如果你真想达到这个目标，请先【**确保**】自己有这方面的天赋，接着再通过《欺骗的艺术》（凯文·米特尼克所著）进行深造。

本系列帖子【不能】帮你化解【所有的】社会工程攻击。毕竟社会工程学的手法太多、涉及的面太广。有些新颖的手法，其设计之巧妙、用心之险恶，估计连俺都会入套。

为了方便阅读，把本系列帖子的目录整理如下（需翻墙）：

[扫盲“社会工程学”1]：攻击手法之【信息收集】

文章目录

上一个帖子普及了一些基本概念和常识，接下来就得说点实在的货色：介绍一下攻击者常用的套路。攻击者的套路大致可以分为如下几个步骤：信息收集、假冒身份、施加影响、实施最终的攻击。由于每个步骤介绍起来都蛮长的，俺今天先来介绍“信息收集”这个步骤。

★什么是信息收集?

信息收集就是通过各种手段去获取机构、组织、公司（以下统一简称“机构”）的一些【不敏感】信息。为啥特地强调“不敏感”捏？如果信息不敏感，就不会有特别严格的访问限制，攻击者也就容易得手。而且在获取这种信息的过程中，不易引起别人的注意，降低了攻击者自身的风险。

★收集的信息有啥用捏?

大部分社会工程攻击者都会从信息收集入手。但信息收集往往【不是】攻击者的最终目的，仅仅是攻击者进入下一个阶段的前期准备工作。大多数攻击者拿到这些信息之后，多半会用来包装自己，以便进行后续的身份假冒。具体如何该包装和冒充，俺会在下一个帖子里介绍。

★哪些信息属于不敏感信息?

典型的**不敏感**信息有如下几种：

◇某些关键人物的资料

这里说的“资料”包括该人物所处的部门、担任的职位、电子邮箱、手机号、座机分机号等。大伙儿注意一下，此处的【关键人物】，不一定是名气大或位高权重的人，而是指这些人位于攻击路线上的关键点。攻击者必须利用这些人来达到某种目的。

◇机构内部某些操作流程的步骤

每个机构内部都有若干操作流程（比如报销流程、审批流程等），这些流程对于攻击者非常有用。一旦摸清了这些流程的细节，攻击者就能知道每一个攻击环节会涉及哪些对象，这些对象分别处于什么部门？担任什么职务？具有什么授权？

◇机构内部的组织结构关系

机构的组织结构关系包括如下几个方面：各个部门的隶属关系、部门之间的业务往来、职权的划分、某个部门是强势还是弱势等等信息。

组织结构图的用处类似于操作流程，俺就不再多啰嗦了。

◇机构内部常用的一些术语和行话

大部分攻击者都会收集一些机构内部的术语和行话。当攻击者在和机构内的其他人员交流时，如果能熟练地使用各种专用的术语和行话，就可以有效打消其他人的疑虑，并获得信任。

上述这些信息似乎蛮普通的，在大伙儿看来好像没啥价值。但是这些信息到了攻击者手中就能发挥出巨大的作用（具体细节，可以看"这里的示例"）。

★如何收集到不敏感信息?

收集这些普通信息的途径大致有如下几种：

◇通过网站和搜索引擎

比如，很多机构的内部操作流程直接放在官方网站上，可以轻易获取。还有很多不敏感信息，攻击者通过 Google 就能找到一大把。

◇通过离职员工

有些时候，某个员工（哪怕是一个很小的角色）跳槽到竞争对手那里，就可以带来很丰富的信息。保本的话，至少能拿到原公司的通讯录；稍好一些的话，还能拿到组织结构图以及更深层次的一些东东。

◇通过垃圾分析

很多机构对于一些普通的打印材料，直接丢到垃圾桶，不会经过碎纸机处理。所以攻击者可以从办公垃圾中找到很多有用的信息。

举一个简单的例子：很多公司每当有新员工入职，人事或者行政人员都会打印一张清单给新员工。清单上面**可能**会有如下内容：

公司内部常用服务器（比如打印服务器、文件服务器）的*IP*地址
新员工外部邮箱的名称和默认口令
公司内部系统（比如 *ERP* 系统、*MIS* 系统等）的用户名和默认口令
某些内部系统的简单使用说明

如果某个新员工没有【立即】修改默认口令（有相当比例的新员工不会在入职当天立即修改【所有的】默认口令），并且把这个清单直接丢到垃圾桶。那对于垃圾分析者来说，可就捡了大便宜啦！

不过捏，垃圾分析方法属于苦差事。使用此招数，每次都要捏着鼻子，在垃圾箱里翻上好几个小时。但还是有很多商业间谍乐此不疲。

◇通过电话问讯

某些攻击者直接打电话给前台或者客户服务部，通过某些技巧（后面的帖子详述），就能套出很多有价值的信息。

为啥攻击者特别偏爱于前台和客服人员捏？这里面可是大有讲究啊！一般来说，前台和客户服务人员都属于机构内的服务支撑部门。这些部门的员工经常被培训成具有如下特质：不怨其烦、热情好客、乐于助人。所以，这类员工会比较有耐心，也比较能满足攻击者的一些（哪怕是有点无理的）要求。

上述就是社会工程学中，信息收集的基本常识。本系列的下一个帖子，咱们来聊一下"假冒身份"的话题。

[扫盲“社会工程学”2]：攻击手法之【假冒身份】

文章目录

在前一个帖子里，咱们介绍了“信息收集”，今天咱们来讲一讲“假冒身份”的手法。

为了避免某些同学误解，有必要事先澄清一下：“信息收集”、“假冒身份”、“施加影响”这三个手法不是孤立存在的，而是有机结合的。攻击者在干坏事的时候，总会混用这三个手法以达到最终目的。俺只是限于时间和篇幅，所以才大卸三块，分开来介绍。

★为啥要假冒？

假冒身份说白了就是“包装”。攻击者又不是傻冒，他们当然不会轻易暴露自己的真实身份，自然要找一个马甲来伪装一下。一般来说，攻击者会根据面对的目标来选取**针对性**的马甲。选好马甲之后，还要在某些细节上稍微粉饰一下，让人觉得更加逼真。

总而言之，包装要为后续的“施加影响”埋下伏笔，打好基础。

★包装要达到啥效果？

按照二八原理，大部分人都是感性的。包装的效果，就是要充分利用和挖掘人【感性的弱点】。

◇博取信任

还记得上一个帖子提到的那些“不敏感信息”吗？攻击者会利用这些信息来证明自己是机构内的人，从而得到信任（具体看文本后面的实例）。博取信任是先决条件，只有先取得信任，攻击者才能再接再厉，继续博取好感、博取同情、树立权威等等。

◇博取好感

博取好感显然是没啥坏处的。如果对方产生了好感，攻击者就便于提出更进一步的要求。比如很多保险推销员就善于利用各种手段来博取好感。

◇博取同情

大部分人或多或少都有一点同情心，某些攻击者会刻意示弱，从而让对方产生一些同情心，然后借机提出一些要求。从这个角度来讲，很多乞丐也利用了社会工程学的技巧。

◇树立权威性

很多人都会对权威人物有一种轻信和盲从。所以，树立权威性也有助于攻击者后续的“施加影响”。

★如何包装?

◇选择身份

要达到上述的效果，首先要选择特定的身份。选择身份是很有讲究的，要综合考虑多方面的因素。由于俺不是教你如何搞社会工程攻击，所以俺只能是简单说一说。

要博取好感，攻击者可以通过建立认同感来达到。比如对方是某个秘书，攻击者会谎称自己是另一个部门的秘书（职务上的认同）。关于认同感，后面的帖子会详细介绍。

要树立权威性，可以通过冒充公司内更高级别的人物（或者和高层相关的人，比如某领导的秘书）。这个招数对于那些等级森严的公司，效果挺好。

要博取同情的话，可以看本文后面举的例子。

◇外貌的粉饰

除了选取身份，一些外貌的细节也很重要。由于大多数攻击者采用电话的方式沟通，那些嗓音略带磁性（仅限于男性）或者充满柔情（仅限于女性）的家伙，就很占优势啦。

大多数攻击者都不会贸然现身（现真身的风险可大了）。万一在特殊情况下需要亲自出马，到对方的机构去拜访，有经验的攻击者都会选取得体的着装，以便和假冒的身份相称。在这种情况下，攻击者的长相也是一个关键因素。那些相貌堂堂、一表人才、玉树临风的家伙，第一眼就会让对方产生好感并放松警惕。

顺便跑题一下。我在本系列开篇的扫盲帖里面不是强调过**天赋**的重要性吗？所谓的社会工程学天赋，不光是脑瓜子机灵，嗓音和相貌也不能太差哦（尤其是嗓音）。俗话说得好：天生嗓音差不是你的错，但跑出来混社会工程就是你的不对啦！

★一个实例

前面忽悠了一大堆理论，为了加深同学们的印象，咱来看个简单的例子（灵感来自凯文·米特尼克所著的《欺骗的艺术》）。在此例子中，攻击者的主要目的是更进一步的“信息收集”。在该过程中，攻击者使用了“假冒身份”的手法。

◇主要人物介绍

某社会工程攻击者，简称小黑。
某公司客服人员，简称小白。

◇背景介绍

小黑想打探这家公司某客户（张三）的银行帐号。小黑先进行了一些初步的信息收集（通过Google），了解到如下信息：
1、公司内部有一个商业客户资料系统，里面包含有客户的银行帐号
2、该系统简称BCIS
3、该公司的客户服务人员有BCIS的查询权限
准备妥当之后，小黑**打电话**到该公司客户服务部。

◇对话过程

小白：你好，哪位?
小黑：我是客户资料部的，我的电脑中了该死的病毒，没法启动了。偏偏有个总裁办的秘书让我查一个客户的资料，还催得很急。听说你们客服部也能登录到BCIS，麻烦你帮我查一下吧。谢谢啦!
小白：哦。你要查什么资料?
小黑：我需要一个客户的银行帐号。
小白：这个客户的ID是多少?
小黑：客户ID在我电脑里，可是我的电脑打不开了。麻烦你根据姓名进行模糊查找，应该能找到的。这个客户叫"张三"。
小白：稍等，我查询一下。
……
小白：找到了，你拿笔记一下，他的银行帐号是1415926535。
小黑：好的，我记下了。你可帮了我大忙啦！太谢谢你了!
小白：不客气。

◇案例分析

首先，攻击者通过信息收集中打听到"商业客户资料系统"简称BCIS。另外，攻击者还了解到"客服部门"有BCIS的查询权限。当小黑很自然地说出这两个信息，就会让小白相信自己是公司内的人员。

接着，小黑通过谎称自己的电脑中毒，来进行示弱并博取小白的同情。

有了上面这两条，小黑成功的把握就很大啦。如果再辅助一些特定的嗓音和语调，并且在言谈中流露出焦急的心情，那基本上就大功告成了。

关于"假冒身份"的话题，就暂时聊到这。本系列的下一个帖子，咱们来聊一下"施加影响"的话题。

[扫盲"社会工程学"3]：攻击手法之【施加影响】

文章目录

上一个帖子咱们介绍了社会工程中**包装**的技巧。按照本系列的计划，今天要讨论的内容是："如何施加影响"。

★关于《影响力》

说到施加影响以及相关的技巧，就不得不提及《影响力》这本书。这真是一本好书啊，很值得俺专门写一个书评来忽悠一下。不过捏，刘未鹏同学已经抢先 N 步，写了此书的评论（在"这里"），所以俺就不再浪费打字的力气去做重复劳动了。

《影响力》这本书高屋建瓴地总结了"对他人施加影响"的种种伎俩。这些伎俩似乎不够光明正大，但常常能收到奇效。如果你从来没有读过此书，强烈建议你先去拜读一下（俺的网盘提供了本书电子版的下载，参见"心理学"大类），再接着来看本帖后续的内容。

……

……

……

现在，不妨假设你已经拜读过《影响力》。接着，咱们来看看书上的那些技巧是如何运用在社会工程学当中的。

★博取好感

博取好感是施加影响的手法中，最基本的招数。具体的技巧有很多种，咱今天只介绍常见的几种。

◇通过外在特征的"光环效应"

此处所说的外在特征，包括相貌、嗓音、着装、甚至姓名等诸多方面；此处所说的"**光环效应**"（也叫光晕效应、晕轮效应），是指对某人的某个局部特征的看法被扩大化，变成对此人整体的看法。这么说比较抽象，咱来看下面几个例子。另外，俺单独写了一个帖子详细介绍"光环效应"，在"这里"。

举例1（以貌取人）

据说当年马云创业时，出去推销产品，别人一看到他都觉得他是坏人。显然，相貌和人品没有必然联系。但是很多人在潜意识里，都会把长得歪瓜裂枣的人当成坏人。

举例2（以名取人）

比如很多歌星、影星仅仅由于演技好，其 fans 就把演技扩大化，认为他们/她们样样都好。其实演技好和人品好没有必然联系。

所以俺在前一个帖子里强调社会工程的攻击者需要有好的嗓音（有时甚至需要有好的相貌），就是为了能发挥光环效应。

◇通过相似性来博取好感

所谓的"**相似性**"，范畴很广，常见的有如下一些：同学、同乡、同校（校友）、爱好相同（比如都喜欢看球，甚至都喜欢某个球星）、经历相同、等。

很多攻击者善于通过看似不经意的闲聊，和被攻击者扯上某种关系，让被攻击者的好感油然而生。

★通过互惠原理来骗取好处

俺在看了《影响力》之后，才意识到互惠原则的效果竟然如此巨大。真是不看不知道，一看吓一跳。具体的例子书上举了很多，俺这里主要总结互惠原则的两种运用招式。

◇初级招式："投桃报李"式

"投桃报李"式比较好理解，简单说就是给予对方一点小甜头，然后再索取点小回报。

为了形象点，举例说明：

比如有个攻击者在信息收集阶段，想了解某个连锁商店店长的信息。攻击者打电话给该商店（接电话的是某店员），谎称自己是一位长期客户，由于该店的服务很好，想写封表扬信给店长。店员一听就很爽，立马就把店长的详细信息告知对方。

◇高级招式："拒绝-退让"式

"拒绝-退让"式比"投桃报李"式要高级一些。这个招式实际上包含了互惠原理和**对比原理**，如果把握得当，效果比"投桃报李"要好很多。具体的实施分两个步骤进行：先提出一个很高（比较过分）的要求（以下简称 A），对方多半会拒绝；然后，攻击者主动作出让步（撤回该要求），再提一个（相对 A 来说）比较低的要求（以下简称 B），这时对方多半会答应。其实 A 仅仅是一个烟雾弹，并不是攻击者的真实意图。攻击者真正想达成的是 B。

这个招式的难点在于把握A的尺度。A必须和B形成比较明显的反差（利用对比原理），通过A来衬托出B的微不足道。这样，对方拒绝了A之后，潜意识里觉得B反正很微不足道，再加上互惠原理的作用，就会很容易地接受B。

比如有些攻击者在收集信息时，可以先索取某个比较敏感的信息，如果对方拒绝了，就转而索取一个不敏感的信息。

★通过社会认同来施加影响

所谓的"社会认同"，通俗地说就是人云亦云、随大流。大多数人都有这个毛病，否则也不会有那么多跟风、赶时髦的家伙了。

那社会工程者如何运用这个伎俩捏？一个常见的方法就是"造势"。通过制造某种舆论来引导（或者叫"误导"）被攻击者，从而达到目的。这种方式有两个要点：

首先，要达成某种规模效应。一旦规模形成，由于"社会认同"的影响，就会变成正反馈，导致越来越多的人被卷入。

其次，要注意引导的技巧。具体要如何"引导"捏？常见的有："制造狂热"、"制造恐慌"、"制造愤怒"、"制造反感"等方式。当人们处于狂热、恐慌、愤怒、反感等状态时，会变得情绪化。这时候，感性的因素就会占主导，同时会丧失理性的判断，从而被一小撮人所利用。

从本质上分析，这两个要点依然是借助了心理学层面的因素来起作用。关于造势的例子，大伙儿可以看看源自 IT 行业的 FUD（Fear, Uncertainty, Doubt，具体解释见"这里"）手法。

写到这里，突然联想到：其实天朝的毛太祖，就是造势、造舆论的高手啊！上述的两个要点发挥得炉火纯青，不得不令人佩服啊！

★通过权威来施加压力

大部分人都有服从权威的倾向。因此攻击者可以通过树立或借助权威，让对方服从自己的一些不太合理的要求。

比如有的攻击者假冒成某 VP（Vice President）的秘书，声称该 VP 急需某某文件或资料，那么对方就会迫于压力而答应。这个招数在等级森严的组织机构，效果特别好。

★总结

有句话俺必须再啰嗦一下：按照二八原理，**大部分人都是感性的。**为啥上述的这些伎俩能够屡试不爽？就因为这些技巧充分利用了人们感性的弱点。如果你是一个感性的人，那可要小心啦：你可能会容易入上述这些圈套，平时须得小心防范。

不过捏，凡事总有两面性滴。你一方面要提防别人通过这些招数影响你，另一个方面，你也可以利用这些东东去影响别人。虽说今天是为了介绍社会工程学才扯了这么多施加影响的招数，但这些玩意儿可不仅仅限于社会工程学哦。在很多很多不同的领域（比如：管理、谈判、社交、追 MM/GG、推销……），今天讲的这些东西都是非常有用滴。大伙儿一定要活学活用、举一反三啊，才不枉费俺打了这么多字！

本系列的下一个帖子，咱们搞几个综合的示例来分析一下。

[扫盲“社会工程学”4]：【综合运用】举例

文章目录

前面的几个帖子已经介绍了社会工程学的一些常见伎俩（主要是“信息收集”、“假冒身份”、“施加影响”这三个手法），今天俺要来举几个综合性的例子。通过这些例子，大伙儿可以见识一下那些社会工程学的老手是如何把各种伎俩有机结合起来，并达到最终的目的。

为了避免引起不必要的误解，俺事先声明如下：

由于本人才疏学浅，难以凭空捏造出各种社会工程学案例的场景，因此后面有些例子的灵感，是来自于凯文·米特尼克所著的《欺骗的艺术》。另外，俺举这些例子只是教大家如何防范，决无教唆的意图。如果有人企图追究俺教唆犯罪的责任，拜托去找米特尼克先生，别来找俺滴麻烦 **:-)**

★举例1：获取通讯录

某个聪明的猎头（按照前面帖子的惯例，不妨称之为小黑）需要搞到一家大公司研发部门的通讯录。为了达到目的，小黑决定采取一些社会工程学的技巧。

首先，要选定突破口——也就是容易被利用的人。在这个案例中，小黑决定从前台和研发部秘书作为突破口。为啥要选择两个人捏？有一个原因在于，这两人的工作性质决定了他们会比较乐于助人，也就比较容易被小黑利用。接下来，咱们看看小黑是如何达到目的。

◇步骤1：获取前台的Email地址

此步骤就是之前的帖子介绍的“信息收集”。由于前台的电子邮件地址不是敏感信息，不会有严格的访问控制，可以比较容易获取。比如想办法拿到前台的名片或者在打电话跟前台套近乎。具体细节俺就不多啰嗦了。

◇步骤2：搞定研发的秘书

接着，小黑打电话给研发的秘书（搞到研发秘书的分机号也不是什么难题），然后谎称自己是总裁办的秘书，**急需**一份研发人员的清单。然后，小黑让研发的秘书把整理好的人员清单发送到 xxxx 邮件地址（也就是步骤1获取的前台 email 地址）。

这个步骤是整个计划的关键点。为了达成此步骤，需要用到“假冒身份”和“施加影响”这两个手法。通过冒充总裁办的人，造成一种潜在的威慑。而且小黑在通话的过程中自然流露出焦急的情绪，显得更加逼真。

对于研发秘书而言：虽然研发人员的清单比较敏感，但由于索要清单的是总裁办的人，也就不好拒绝了。而且对方留得email地址是本公司的邮箱，想想也就没啥好顾虑的了。

◇步骤3：搞定前台

打完步骤2的电话之后，小黑就赶紧打第3个电话给前台。下面是双方的对话。

> 前台：你好，哪位？
> 小黑：我是总裁办的XXX秘书。
> 前台：你有什么事情吗？
> 小黑：我正陪同XX副总裁在某处开会，XX副总裁需要一份资料。我已经找人整理好了，等一下会发到你的邮箱。你收到之后，请帮忙传真到XXXXXXXX号码。
> 前台：好的。还有其它事情吗？
> 小黑：没有了，多谢！

对于前台而言，她先接到一个电话让她收邮件，紧接着确实收到一份从公司内部邮箱发出的材料。所以她自然也就不会起疑心了。

★举例2：获取财务报表

前面的那个例子稍显简单，再来说一个稍微复杂点的例子。

某商业间谍兼资深黑客（还是简称为小黑）需要搞到某大公司内部的财务报表（可以卖大价钱哦）。由于这个财务报表是很敏感的资料，一般员工是接触不到滴，只有财务部的少数主管才能看到这些报表。而财务部的主管，肯定都知道这些报表的重要性。所以，小黑再想用"案例1"的伎俩是行不通滴。

小黑冥思苦想之后，决定采用“木马计”，在财务主管的电脑中植入木马（如果你不晓得“木马”是啥，自己先上网查一下）。一旦木马植入成功，那财务报表就是手到擒来、不费吹灰之力了。具体的实施步骤如下：

◇步骤1：准备阶段

准备阶段主要办三件事：首先，想办法搞到公司的通讯簿。通过案例1，大伙儿应该知道这个不难办到；然后，通过各种途径（具体的途径，请看之前的“信息收集”）了解该公司内部的一些情况（尤其是IT 支持部和财务部的人员情况）；最后，用**化名**去开通一个手机（有经验的攻击者肯定用假名，以免被抓）。

◇步骤2：忽悠财务主管

由于前面的准备工作，小黑了解到财务部某主管（不妨叫小白）的姓名和分机号。然后小黑打给该主管。下面是双方对话。

> 小白：你好，哪位？
> 小黑：我是IT支持部的张三。你是财务部的主管小白吧？
> 小白：对的。有啥事儿？
> 小黑：最近几天，你们财务部的网络正常吗？有没有感觉网络时断时续的？
> 小白：好像没有嘛。
> 小黑：有几个其它的部门反映网络不正常，所以我来问问你们的情况。如果这几天你碰到网络异常，请打电话给我。我最近忙着处理电脑网络的故障，不经常在座位上。你可以打我的手机，号码是13901234567。
> 小白：好的，我记一下。

> 小黑：另外，我想确认一下你电脑的网络端口号。
> 小白：什么是“网络端口号”？
> 小黑：你先找到你电脑的网线，在网线插在墙上的地方应该贴个标签，那上面的写的号码就是你电脑的“网络端口号”。你把上面的号码告诉我。
> 小白：等一下，我看一下……哦，看到了，上面写着“A1B2C3”。
> 小黑：嗯，很好。我只是例行确认一下。祝你工作愉快。再见。

◇步骤3：欺骗IT支持部

接着小黑耐着性子等待2到3天，然后打电话给 IT 支持部的某工程师（不妨称李四）。由于之前的准备工作，小黑知道李四管理公司的某些路由器和交换机。

小黑谎称自己是新来的网络工程师，正在财务办公室帮小白排查网络问题，请李四帮忙把网络端口号为“A1B2C3”的网络连接断开。

对李四而言，由于对方能准确说出小白的姓名以及小白电脑的网络端口号，所以李四就相信了他的话，并按照要求把对应的网络连接断开。

◇步骤4：等待鱼儿上钩

打完这个电话之后，接下来小黑就稍息片刻，等着小白的电话。果然，不出几分钟，小白就打了他的手机。

> 小黑：你好，我是IT支持部的张三。你是哪位？
> 小白：我是财务部的小白主管。前几天你给我打过电话的，还记得吗？今天网络果然出问题了。所以打你电话找你帮忙。
> 小黑：哦，是吗？那我帮你查一下，应该很快能搞定的。

大约十分钟之后，小黑重新打给IT支持部的李四，让他把端口号为“A1B2C3”的网络连接重新开通。

◇步骤5：大功告成

网络重新开通之后，小黑又打给小白。

> 小白：你好，哪位？
> 小黑：我是IT支持部的张三。刚才已经帮你把网络故障解决了。你现在试试看，网络应该通了。
> 小白：我看一下，嗯，果然通了！太好了！太谢谢你了！
> 小黑：不过，最近几天这个问题可能还会反复出现。
> 小白：啊！那可咋办？我们财务部月底正忙着呢？可经不起这个折腾啊！
> 小黑：办法倒是有一个，你需要安装一个网络模块的补丁，基本上就可以解决这个问题了。我等一下发到你邮箱中。你收到之后，把邮件附件中的程序安装一下就行了。
> 小白：哦，好的。
> 小黑：顺便提醒你一下，有些杀毒软件可能会把这个补丁误报为有害程序。你如果碰上这种情况，可以先把杀毒软件关闭，再重新安装一次就可以了。
> 小白：哦，我晓得了，谢谢。

然后，小黑就往小白的邮箱发了一个木马，并且把邮件的发件人地址伪装成 IT 支持部张三的地址，免得引起怀疑。

对于小白而言，张三（冒充的）刚刚帮他解决了网络故障。所以小白根本不会怀疑此人的身份。自然也不会怀疑邮件有诈。

★总结

由于篇幅有限，俺就不多举例了。从上述案例来看，社会工程高手在搞定复杂问题之前，一般会制定好一个计划，并且在计划的每一个步骤都会充分利用前面几个帖子提到的技巧。另外，在整个攻击过程中，攻击者无非就是做一些调研，打几个电话，成本非常低，被抓的风险也很小；而他们一旦得手，获益却很大。可能就是由于这种较大的反差，导致社会工程攻击在整个信息安全领域的比重不断增加。

本系列的下一个帖子，俺来介绍一下如何防范社会工程学攻击。

[扫盲“社会工程学”5]：你该如何【防范】？

文章目录

经过前面几个帖子的介绍，大伙儿应该能看出来，社会工程学的应用范围是很广泛滴。它的应用会涉及日常生活的许多领域，绝不仅限于信息安全。所以，如何防范就是一个重要的话题了。今年咱们就来聊一下如何防范。

★组织机构该如何做？

如果你是某公司/某机构里的一个小头目或大头目、甚至老板，那就得多看看这一节；否则的话，直接跳过本节，看下一个章节（个人该如何做）。

◇普及教育

最要紧的一条就是普及教育了。否则俺也不会在电脑前吭哧吭哧打这么多字，写这么个系列了。一些常识性的基础培训是很重要滴。按照二八原理，20%的简单培训就可以防范80%的潜在攻击。由于“**人**”是社会工程攻击的主要对象，并且有经验的攻击者都善于寻找组织机构的弱点，所以普及教育务必要涵盖到每一个人（连公司的扫地阿姨也不要放过哦**:-**）。

另外要强调的一点是：要重视对新员工的培训。很多时候，新员工往往是攻击者的突破点。首先，新员工初来乍到，跟周围的同事不熟，容易把攻击者误认为同事；其次，新员工往往怕得罪人，容易答应攻击者的各种要求。

◇严格的认证

认证（Authentication）是一个信息安全的常用术语。通俗地说，认证就是解决**某人到底是谁**?

由于大部分的攻击者都会用到“身份冒充”这个步骤，所以认证就显得非常必要。只要进行一些简单的身份确认，就能够识破大多数假冒者。比如碰到公司内不认识的人找你索要敏感资料（参见“这里”的示例），你可以把电话打回去进行确认（最好是打回公司内部的座机）。

◇严格的授权

授权（Authorization）和认证一样，也是一个常用的信息安全术语。通俗地说，授权就是解决**某人到底能干啥**？

对于组织机构来说，授权要尽量细化、尽量最小化。

举个例子。如果某软件公司中，**所有的**程序员都可以访问**所有的**源代码，那源代码泄漏的风险就很大。只要有一个人出问题，攻击者就可以得逞。反之，如果每个人只能访问自己开发的那部分代码，那安全风险就会小很多。即使某人上当受骗，也只会泄漏部分代码。

◇信息分类

在组织机构中，最好要有信息分类的制度。根据信息的重要程度，定出若干级别。越是机密的信息，知道的人越少。

比如在我负责的团队中，源代码的敏感度高于软件安装包。因此，源代码服务器只有开发人员能够访问；而放置安装包的发布服务器，大部分人（比如测试人员、产品人员）都可以访问。

◇别乱丢办公垃圾

看完信息收集的帖子，大伙儿应该明白，乱扔垃圾可不光是砸到花花草草的问题，更危险的是给垃圾分析者提供了大量有价值的素材。这也就是为啥要给扫地阿姨培训社会工程学的道理。

◇文化

最后再来说一下企业文化对社会工程攻击的影响。

在之前的帖子，俺已经介绍了“通过权威来施加压力”的攻击手法。如果某个组织机构的等级很森严，就容易给攻击者留下利用的机会。还有一些组织机构，里面的人员都是好好先生，每个角落都是一团和气。这种机构和等级森严的组织一样，容易被攻击者利用。

所以，假如你碰巧是组织机构内部的一个实权人物，或许可以尝试改变一下现状。不过俺要提醒一句，一个组织机构（尤其是政府机构）的文化是很难轻易改变滴。所以，别对这个招数报太大希望 :-(

★个人该如何做?

前面介绍了企业内部的防范措施，接着就该说说个人该如何应对了。

◇多了解一些社会工程学的手法

俗话说：知己知彼，百战不殆。如果你不想被人坑蒙拐骗，那就得多了解一些坑蒙拐骗的招数。除了俺提到过好几次的《欺骗的艺术》（凯文·米特尼克所著），你还可以通过互联网找到很多类似的资料。这些资料有助于你了解各种新出现的社会工程的手法。

另外，很多文学作品、影视节目也会掺杂社会工程学的情节。比如前段时间热播的《潜伏》，里面的主人公余则成显然是一个社会工程学老手。细心的同学应该能从中窥探到不少奥妙。

◇保持理性

在如何施加影响的帖子里，俺已经列举了很多种手法。这些手法不外乎都是利用人【**感性**】的弱点，然后施加影响。所以，尽量保持理性的思维（尤其在和陌生人沟通时）有助于减少你被攻击者忽悠的概率。不过捏，保持理性，**说起来简单，做起来未必简单** :-(以后俺有空再来聊聊这方面的话题。

◇保持冷静

还有一些"社会工程学"的惯用伎俩是【制造恐慌】。大部分人在慌乱之中就容易入套。
所以，保持冷静也很重要。不过捏，还是刚才那句话——**说起来简单，做起来未必简单。**

◇保持一颗怀疑的心

这年头，除了骗子是真的，啥都可能是假的。比如，你收到的邮件，发件人地址是很容易伪造滴；比如，你公司座机上看到的来电显示，也可以被伪造；比如，你收到的手机短信，发短信的号码也可以伪造。
所以，保持一颗怀疑的心，也是非常必要的啊！

◇别乱丢生活垃圾 :)

不光上述提到的办公垃圾有潜在风险，生活垃圾一样也会被垃圾分析者利用。比如有些粗心的同学会把帐单、发票、取款机凭条等东西随意丢在垃圾桶中。一旦碰上有经验的垃圾分析者，你没准就麻烦了。

◇ （其它）

肯定还有俺没提及的防范措施，欢迎大伙儿（到博客评论区）补充。