# Kerberos 协议安全性的符号模型检验分析

郭云川　古天龙　董荣胜　李凤英
桂林电子工业学院计算机系，桂林 541004）
E-mail：guoyunchuan@hotmail.com

摘　要　基于模型检验的安全协议分析和验证是协议工程研究的一个新方向。该文建立了 Kerberos 协议的有限状态机模型，并用符号模型检验器（SMV）从安全属性的两个方面——认证性和保密性分析了 Kerberos 协议，指出了 Kerberos 协议的缺陷。

关键词　符号模型验证　安全性　重放攻击

文章编号 1002-8331-（2003）29-0177-04　文献标识码 A　中图分类号 TP393.04；TP393.08

# Symbolic Model Checking Analysis for Security of Kerberos Protocol

Guo Yunchuan　Gu Tianlong　Dong Rongsheng　Li Fengying
（School of Computer Science，Guilin University of Electronic Technology，Guilin 541004）

**Abstract**：Model-checking has been a new direction of analyzing and verifying security protocols.In this paper，Kerberos protocol is modeled as finite state machine，and analyzed in two aspects of security property-authentication and secret property using SMV.As a result，an attack upon the protocol is discovered.

**Keywords**：Symbolic Model Checking，Security，Replay Attack

## 1　引言

现在计算机网络得到了飞速发展，为了保护网络中信息安全，我们常常使用安全协议。用安全协议来保证信息传输安全，首先必须保证安全协议本身是安全的。但是安全协议却常常存在瑕疵。因此在协议设计后期，采取一定的方式对协议进行分析验证是必要的。现在验证的手段有多种，比如：直观的分析，实际攻击测试和形式化的验证分析技术[1]。其中形式化技术以其手段的严密和断言的普遍使用见长。目前常用的形式化协议分析和验证技术有逻辑验证（如 BAN 逻辑[2]）和模型验证（如符号模型验证（Symbolic Model Checking）[3]）。在模型验证中，为了分析协议的安全性，需要对协议进行建模，然后将协议模型和安全性要求输入到模型检验器中，模型检验器将分析系统所有的行为已确定协议是否满足安全性要求。如果系统的行为满足系统所需要的属性，模型检验器输出 TRUE，否则，输出 FALSE，并生成不满足系统所需要属性的反例。Kerberos 协议是给计算机网络提供身份验证的安全协议，其基础是基于信任第三方，集中地进行用户认证和发放电子身份凭证，它提供了在开放型网络中进行身份认证的方法，认证实体可以是用户或用户服务，这种认证不需要保证网络上所有主机的物理安全性。本文使用符号模型检验器（Symbolic Model Verifier）[6]对 Kerberos 协议的安全性进行了分析，并指出了协议中的瑕疵。

## 2　Kerberos 协议介绍

Kerberos 协议[4]主要目的在服务器和客户之间进行密钥交换和身份认证。Kerberos 协议共有 5 个版本，在基本的 Kerberos 协议中，协议参与者有 3 方：A，B，以及 A 和 B 都信任的认证服务器 S。Ta，Ts 分别为 A 和 S 的时间戳，L 是消息的生命期，Kab，Kas，Kbs 分别为 A 和 B，A 和 S，B 和 S 的共享密钥。其协议如下：

(1) A->S：A，B

(2) S->A：$\{Ts,L,Kab,B\}_{Kas}$，$\{Tb,L,Kab,A\}_{Kbs}$

(3) A->B：$\{Ts,L,Kab,A\}_{Kbs}$，$\{A,Ta\}_{Kab}$

(4) B->A：$\{Ta+1\}_{Kab}$

首先，A 向 S 发送协议参与者的初始方和响应方身份 A 和 B；然后，S 向 A 发送消息 2，消息 2 由两部分组成：第 1 部分由共享密钥 Kas 加密时间戳 Ts，生命期 L，共享密钥 Kab 和主体 B 组成，第 2 部分同第 1 部分类似。第 3 步中，A 用与 B 的共享密钥 Kab 加密 A 和 A 的时间戳 Ta，然后将这个消息，连同 A 向 B 转发 S 的票据一并发送给 B；最后，B 提取 Ta，将 Ta 加 1 之后用共享密钥 Kab 加密，最后转发给 A。依照 Kerberos 协议的设想，第 3 步中，当 B 获得票据时，那么 B 就可以确认 A 的身份并获得与 A 的共享密钥 Kab。第 4 步中，A 收到用 Kab 加密的 Ta+1 时，就可以确认 B 的身份，这样 A 和 B 之间就建立了一个秘密共享的通信信道。

## 3　SMV 系统

SMV 是在有限状态模型的基础上，检测系统属性或断言的工具。SMV 程序由两部分组成：有限状态转换系统和计算树逻辑（Computation Tree Logic--CTL）[3]公式。将这两部分输入到 SMV 系统并运行 SMV 系统后，若有限状态系统满足 CTL 所

基金项目：广西十百千人才专项基金项目；广西自然科学基金项目（编号：0141046）资助

作者简介：郭云川，硕士研究生，研究方向：网络安全，形式化技术。古天龙，博士，教授，博士生导师，主要研究领域：形式化技术及工业应用，离散事件/混杂系统理论及应用，协议工程等。董荣胜，副教授，主要研究领域：形式化技术及工业应用，计算机科学与技术方法论，计算教育哲学。李凤英，助教，硕士，主要研究领域：实时系统，面向对象软件工程，协议验证技术。

描述的属性，则输出真，否则输出假，并同时生成不满足属性的反例。在 SMV 系统中，系统状态集合和迁移关系都是用布尔公式隐式地表示，并在符号状态空间上进行搜索。在大多数情况下，符号化表示一个集合比显示的表示更紧凑。在 SMV 中，布尔公式用高效率的 OBDDs (Ordered Binary Decision Diagrams)来表示和操作，而 OBDDs 有如下优点：对大多数布尔函数都比较紧凑；存在唯一的表示形式；由高效算法来直接进行各种布尔运算和不动点运算。因此，符号技术能有效地缓解了状态爆炸问题。现在符号模型验证技术可以验证多达 $10^{120}$ 状态的系统。SMV 原理如图 1 所示。

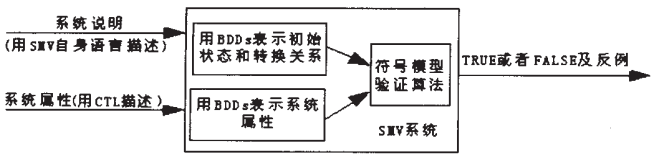


图 1　SMV 工作原理

## 4　Kerberos 协议分析

### 4.1　Kerberos 协议的数据结构

密钥 (key)记录加密和解密者，其数据结构如下：

```
typedef key struct{
  k1 :{A,B,S,I} ;//加密方，其中 A,B,S 是协议合法参与者，I 为入侵者
  k2 :{A,B,S,I} ;//解密方
}
```

子消息 (child_mess)纪录消息初始者，响应者，时间戳和各种密钥，其数据结构如下：

```
typedef child_mess struct{
    initiator :{A,I} ;//消息初始者
    responder :{B,I} ;//消息响应者
  ts 0..10 //时间戳 ts，为了简化，假设时间戳在 0 到 10 之间
  ta 0..10 //时间戳 ta，为了简化，假设时间戳在 0 到 10 之间
  key_session :key ;//会话密钥
  key_share :key ;//共享密钥
}
```

消息 (message)纪录消息的类型，消息发送方，接收方以及协议的各种数据。其数据结构如下：

```
typedef message struct{
  type :{idle,mess1,mess2,mess3,mess4,auth} //消息类型
  cmess1 :child_mess ;
  cmess2 :child_mess ;
  destination :{A,B,S,I} //消息接收方
  source :{A,B,S,I} //消息发送方
}
```

把消息分解成两条子消息的原因是：消息 3 需要转发消息 $\{Ts,L,Kab,A\}_{Kbs}$。

### 4.2　Kerberos 协议的有限状态系统

#### 4.2.1　合法主体

在 Kerberos 协议中，合法主体 A、B 和 S 将严格按照协议的要求进行通信，协议中的三个参与主体在 SMV 系统中都对应了一个有限状态自动机，这三个有限状态自动机都在一个 MODULE node 之内，node 的结构为如图 2 所示，每个主体都是这个 MODULE 的一个实例 (instance)，这三个实例分别为 A,B,S。在协议运行的时候，每个主体选择各自相应的自动机运行，即分别选择图 2 中的 FSM_A、FSM_B 和 FSM_S，FSM_A、FSM_B 和 FSM_S 的有限状态系统转换图如图 3、图 4 和图 5 所示。A 的状态为{idle,G_mes1,W_mes2,G_mes3,W_mes4,Auth},B 的状态为{idle,W_mes3,G_mes4,Auth},身份认证服务器的状态为{idle,W_mes1,G_mes2,Auth} (W、G 分别表示等待、生成，如 W_mes3 表示等待消息 3 状态)。A,B,S 的初始状态均为 idle，如果主体 A 要同主体 B 通信，那么自动进入 G_mes1 状态，选择 B，生成消息 1，向认证服务 S 发送消息 1 后，进入 W_mes2 状态，等待接收消息 2，当接收到消息 2 后，判断消息 2 是否符合协议要求，如果符合协议要求则进入 G_mes3 状态，按协议要求生成消息 3，向 B 发送消息 3，然后转到 W_mes4 状态，等待并验证消息 4，如果符合协议要求，进入 Auth 状态，结束一次协议运行。如果 A 进入 Auth 状态表示 A 完成一次对 B 的认证。B 和 S 的状态转换图与 A 类似。

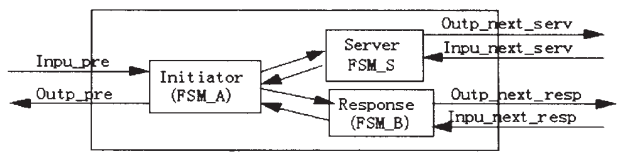


图 2　合法交易主体模块 (MODULE node)

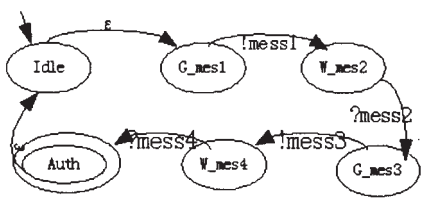


图 3　初始者 A 的状态转换图

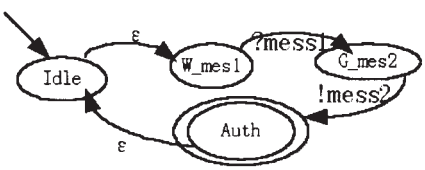


图 4　认证服务器 S 的状态转换图

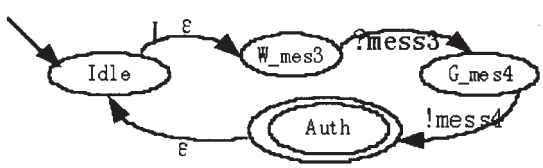


图 5　响应者 B 的状态转换图

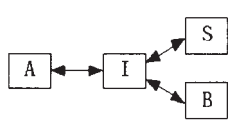


图 6　通信模型

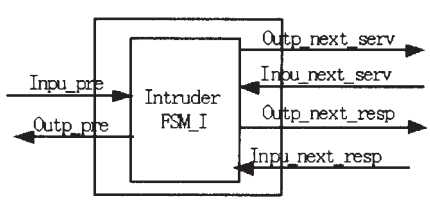


图 7　入侵者模块 (MODULE Intruder)

为了模拟协议运行，初始方 A 发送消息 1 之后，通过 init_save_name 来保存响应方的名称 B；服务器 S 在接收到消息 1 后，通过变量 serv_memo_init 和 serv_memo_resp 来记录协议运行的初始方 A 和 B；初始方 A 在接收到消息 2 后，通过变量 init_memo_mess 来记录消息 2 的第 2 部分；当 B 接收到消息 3 之后，通过分别用变量 resp_memo_ta、resp_memo_init 来记录时间戳 ta 和协议 A。在 A 收到消息 4 之后，通过变量 init_memo_ta 来记录 A 收到的 ta。为了便于验证，针对不同的主体引入不同的计数器：A_session，B_session 分别表示 A 或 B 作为初始者或响应者开始认证的次数，A_count_auth 和 B_count_auth 分别表示 A 或 B 完成认证的次数。所有计数器的初始值均为 0，每当开始或者结束一次协议认证时，相应的计数器加 1。

4.2.2 非法主体

为了分析协议的安全性，引入入侵者并且假设密钥是“良好的”，即入侵者能够解密密文消息，当且仅当入侵者拥有解密密钥；如果入侵者能够加密消息，当且仅当入侵者拥有加密密钥。在协议运行中，入侵者将不依照协议的要求，而是根据自己拥有知识的情况，进行偷听、存储、删除、生成、转发和重放消息，其通信模型如图 6，任何主体之间的通信都要通过入侵者 I。入侵者也对应了一个非常复杂的有限状态机，这个有限状态机在 Module intruder 如图 7）之中，其实例为 I。为了记录入侵者拥有和存储的知识，引入了一个一维数组 known_mess，在初始状态 know_mess 的内容为 $K_{IS}$，随着协议的运行，know_mess 的内容逐渐增加，协议运行的第一步，入侵者可以截获消息 1，从而使 A，B 将成为 know_mess 的内容。协议运行的第 2、3、4 步，入侵者分别截获 $\{Ts,L,Kab,A\}_{Kls}$，$\{Ts,L,Kab,B\}_{Kas}$，$\{A,Ta\}_{Kab}$，$\{Ta+1\}_{Kab}$。用 SMV 编程的时候，入侵者生成消息就是入侵者将 know_mess 中的内容按照消息的结构进行重新组合；删除消息是将消息的 destination 的内容修改为 I，转发消息是修改消息的 source。重放过程是将 know_mess 中的内容不作修改地再次发送。为了便于检验，同样在入侵者中引入计数器 I_get_kab 来表示入侵者 I 获取 A 和 B 之间共享密钥 kab 次数，其初值为 0，每当 I 取得一次 Kab，那么 I_get_kab 加 1。

## 4.3 协议的安全属性要求的 CTL 描述

安全协议的安全性要求：（1）认证性：确定对方的真实身份与对方所称的身份是否一致；（2）保密性：信息存取和信息在传输过程中不能被非法窃取；（3）完整性：要求数据在传输或存储过程中，不会受到非法地生成、修改和删除。（4）不可否认性：要求发送方对自己发出的信息不可抵赖，同时接收方对自己收到的信息无法否认。用如下 CTL 公式来描述认证性和保密性：

R1：AG（(A.init_state=idle & A.init_save_name=B)->AF (B.resp_state=Auth & B.resp_memo_init=A))&AG!(A.A_session=1 & B.B_count_auth=2)

R2：AG（(A.init_state=idle & A.init_save_name=B)->AF (A.init_state=Auth & A.init_memo_ta=ta))&AG!(B.B_session=1 & A.A_count_auth=2)

R3：AG(I_get_kab=0)

R1 表示 B 对 A 的身份认证，即如果 A 需要同 B 通信，那么最后 A 和 B 会完成认证，并且在每次完成认证后 B 都认为初始方为 A，不存在 B 认为 B 与 A 进行了两次通信，而 A 实际上只与 B 进行了一次通信的情况。R2 同 R1 类似，表示 A 对 B 的身份认证。R3 为协议的保密性，即 I 不能截取到会话密钥。其中 AG，AF 是 CTL 的量化符，公式 AG p 表示在所有路径上的所有状态下公式 p 成立，AF p 表示在所有路径上的公式 p 最后成立，&、->和！分别表示逻辑与、逻辑蕴含和逻辑非。

在 Kerberos 协议中，如果协议具有保密性，即不会泄漏任何密钥，那么入侵者解密消息中任何内容，因此，不可能删除、篡改用加密过的任何内容，因此，在 Kerberos 协议中保密性蕴含了完整性。由于 Kerberos 协议是采用对称密钥加密，因而不具备非否性，因此不对非否认性进行验证。

## 4.4 协议验证结果分析

将上述协议模型转化成 SMV 语言并将此 SMV 语言和安全性要求输入到 SMV 中，运行 SMV 之后，发现 Kerberos 协议模型并不符合安全协议的要求，经分析，协议存在如下的攻击：

（1）A->S：A，B

（2）S->A：$\{Ts,L,Kab,B\}_{Kas}$，$\{Tb,L,Kab,A\}_{Kbs}$

（3）A->B：$\{Ts,L,Kab,A\}_{Kls}$，$\{A,Ta\}_{Kab}$

（3′）I（A）->B：$\{Ts,L,Kab,A\}_{Kls}$，$\{A,Ta\}_{Kab}$

（4）B->A：$\{Ta+1\}_{Kab}$

（4′）B->I（A）：$\{Ta+1\}_{Kab}$

这样在第 3 步 I 偷听 A 向 B 发送的消息 3，并在 3′中将消息 3 转发给 B，在第 4′步，拦截 B 向 A 响应消息。这样 B 将认为 A 请求了两次会话连接，而实际上 A 仅有一次会话请求。从而造成了入侵者可以进行重放攻击。

文献[2]用 BAN 逻辑对 Kerberos 协议进行了验证，结果表明 Kerberos 是安全的，这是因为文献[2]假设“时间戳 Ta 是新鲜的”，即“在本次协议运行之前，Ta 没有被发送过”。但目前没有任何协议描述支持这种假设[5]。文献[5]在对 Kerberos 协议的分析中假设了 B 在时间戳允许的时间内能够检测到消息是否重放。然而，正如文献[7]指出了用时间戳来防止重放攻击时，需要假设主体的时钟与服务器的时钟同步。虽然可以通过时间服务器来实现时钟同步，但是在具体实现的时候非常复杂，只能保证较弱的安全性。因此，这种攻击是可能发生的。为了防止这种重放攻击，最新版的 Kerberos 协议中，在消息 3 中加入了临时值选项来保证消息 3 的新鲜性。

# 5 结论

该文用 SMV 从认证性和保密性两个方面对 Kerberos 协议安全性进行了验证，指出了协议的瑕疵。在该文修改稿完成之际，发现 M.pati 等人[8]也分析出了这种缺陷，他们侧重在对一致性的扩展，而笔者则侧重在协议的分析过程。与其他的形式化验证技术相比较，模型验证具有更多的优点：能够自动地进行，当系统模型不满足系统要求时，模型检验器会自动生成不满足系统规格的反例，这些反例反映了模型中的瑕疵，指出了如何进一步修改模型以满足所需求的属性。（为了便于交流，笔者愿意提供源程序）（收稿日期：2003 年 1 月）

# 参考文献

1 Stefanos Gritzalis，Diomidis Spinellis，Panagiotios Georgiadis Security

Protocols over Open Networks and Distributed Systems:Formal Method for Their Analysis,Design and Verification[J].Computer Communications,1999,22(8):695~707
2.M Burrows,M Abadi,R Needham.A Logic of Authentication[J].ACM Transactions on Computer Systems,1990,8(1):18~36
3.K L McMillan.Symbolic Model Checking[M].Kluwer Academic Publishers,1993
4.B Clifford Neunman,Theodore Ts,o Kerberos.An Authentication Service for Computer Networks[J].IEEE Communications,1994,32(9):33~38
5.S G Stubblebine,R N Wright.An Authentication Logic with Formal Semantics Supporting Synchronization,Revocation,and Regency[J].IEEE Transactions on Software Engineering,2002,28(3):256~285
6.SMV.http://www-cad.eecs.Berkeley.edu/~kenmcmil
7.王育民,刘建伟.通信网的安全:理论与技术[M].西安电子科技大学出版社,1998
8.M Panti,L Spalazzi,S Tacconi.Using the NuSMV Model Checker to Verify the Kerberos Protocol[C].In:Proceeding of the 3rd Collaborative Technologies Symposium,2002

(上接 160 页)

毒软件有时候可能无法完全解决问题。中国软件市场中的反病毒软件种类繁多但也良莠不齐,他们某些名不副实的广告使得广大用户对他们产生了过大的依赖性。但实际上他们的真实水平大多却并非如此,甚至相差甚远。譬如,CIH 病毒爆发前期,不少反病毒公司声称他们可以轻松解除 CIH 病毒,但是 CIH 病毒还是毫不留情地破坏了他们用户的宝贵数据。某杀毒软件居然还直接导致了用户数据的丢失。

众所周知,病毒是交叉并呈几何级数向外扩散传播的。如果能够控制其中大部分传染源,切断其大面积的传播途径,那么它的传播速度和范围会大大缩小。CIH 首次爆发是在 1998 年 4 月 26 日,但是真正引起世界性大灾难是在 1999 年 4 月 26 日,如果事先有一种专门克制 CIH 病毒的杀毒程序在 Internet 上传播并发挥它的监测、杀毒和免疫作用的话,那么在某种程度上,可以大大缩小它的传播范围,避免广大用户遭受损失。

这在对付蠕虫上效果会更加明显。譬如,席卷整个欧美大陆的美丽莎病毒,他主要是通过向 outlook 地址簿中的 50 个邮件地址发送电子邮件疯狂传播自己的,如果一个病毒源传播 n 代,那么它可以感染 50 的 n 次方台计算机。但是,如果能够尽快截获其中一部分上级传染源的话,那么受感染的计算机会大大减少。这样,在相同的时间内,它产生的邮件数量会大大减少,它产生的邮件风暴规模和导致的损失也会大大减小。

虽然国际上反病毒技术得到了很大的发展,特别是随着启发式扫描和虚拟机技术的应用,清除未知病毒成为可能。但是,这两种技术的发展还是很不成熟的,特别是在人工智能技术没有取得关键突破的情况下,这两种技术的最终实现还有很长一段路要走。在这种情况下,利用以‘毒’攻毒作为杀毒软件的一种辅助杀毒方法,至少对于广大用户来说,是非常有利的并且有它的存在价值。

## 7 以‘毒’攻毒面临的一些问题

虽然以毒攻毒不是一种新观点,类似目的的病毒至少也出现过大约几十个[5]。但是这种方法似乎一直没有引起足够的重视。或许,人们对它可能导致的问题有些担心。

(1)计算机病毒是否会像生物病毒那样具有非常大的不可确定性[5]?计算机病毒毕竟和生物病毒还是有很多不一样的地方。一个生物病毒可能会在离开特定的温度、培养基或其他条件后发生变异,由治病变成一种恶性的致病菌。但是电脑病毒没有这么多的不确定性因素,或许它会被某些病毒制造者加以修改。但是这并不妨碍人们用它去解除病毒。

(2)一些人担心病毒制造者会对其传播感染机制感兴趣,并以此作为新病毒模板,而做出破坏性更大的病毒。的确,病毒的传染技术一直是病毒技术的重点。在这点上,可以考虑利用比较传统的传播方式。对程序的传播机制或许需要下一番功夫,每一种先进的技术都存在被用在非法途径上的可能。这一点有待进一步研究解决。

(3)关于变种的问题。变种是无法避免的。但相对于自己程序的变种,或许比较好对付一些。当然,也可以对该程序作一定的加密处理,以避免轻易产生变种。

(4)杀毒软件的认可。若要使程序畅通无阻,首先必须通过杀毒软件这一关。如何保证程序不被杀毒软件给杀掉(虽然,程序并非严格意义上的病毒,但是很难保证某些杀毒软件为了自身的利益而痛下杀手)。这可能需要政府的干预了。政府在给予杀毒软件公司权利的同时,必须赋予他们一定的义务。为了防止其他病毒程序效仿自己的程序而躲避杀毒软件。可以采取一些技术上的措施,譬如,在确认杀毒程序的标志以后,再对杀毒软件作一定的校验,等等。

(5)没有授权的数据更改。计算机病毒疫苗要正常工作,首先它必须要修改用户的数据以便自己留在用户计算机内监测病毒。这样,就离不开国家政策法规的支持。

(6)用户的信任。由于计算机病毒的破坏性和不可控制性给广大用户带来了巨大的损失和不快。因此,他们有可能在最初无法接受‘利用具有以毒攻毒特性的计算机病毒疫苗杀毒’这个事实。但是,随着具有以毒攻毒特性的计算机病毒疫苗的成功应用和对它的了解,相信广大用户是会慢慢接受它的。

可见,以‘毒’攻毒并不是一种异想天开,相反,它是一种值得推荐的杀毒方法,同时,在用户防毒意识薄弱和反病毒技术不是非常理想的情况下,它作为一种辅助杀毒方法还是非常必要的。但是,这种杀毒技术的实现,也决非哪一个人或单位能够做到并作得很好的工作,它必须依靠各国政府和广大杀毒软件公司和用户的支持。(收稿日期:2002 年 12 月)

## 参考文献

1.Cyrus Peikari.how virus writers can save the world?.www.virusmd.com
2.国务院令.中华人民共和国计算机信息系统安全保护条例.1994-02-18
3.Fred Cohen.Computer Viruses— Theory and Experiments[J].Computer &Security,1987;6(1):22~35
4.反病毒蠕虫程序问世 是喜是忧孰可知.计算机世界网,2001-09-05
5.江海客.以毒攻毒是一种异想天开.http://www.yesky.com/20010801/190926.shtml.2001.7.29