[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
H04L 12/24 (2006.01)
H04L 29/06 (2006.01)

[12] 发明专利说明书

专利号 ZL 200610033899.2

[45] 授权公告日 2008 年 3 月 26 日

[11] 授权公告号 CN 100377534C

[22] 申请日 2006.2.20
[21] 申请号 200610033899.2
[73] 专利权人 华为技术有限公司
地址 518129 广东省深圳市龙岗区坂田华为总部办公楼
[72] 发明人 董亚波 涂卫华 郑志彬
[56] 参考文献
US2004/0205474A1 2004.10.14
CN1697404A 2005.11.16
US2005/0216764A1 2005.9.29
CN1549126A 2004.11.24

Fast Portscan Detection Using Sequential Hypothesis Testing. Jaeyeon Jung, Vern Paxson, Arthur W. Berger, HariBalakrishnan. Proceedings of the 2004 IEEE Symposium on Security and Privacy. 2004

蠕虫扫描检测算法的注记. 王琦，杨莉莉，宋如顺. 兵工自动化，第 24 卷第 6 期. 2005
审查员 何永春

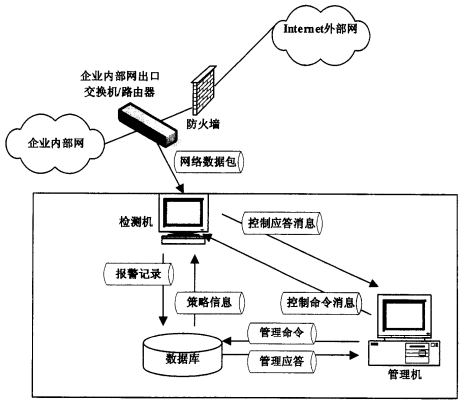
权利要求书 6 页 说明书 21 页 附图 3 页

[54] 发明名称

一种网络蠕虫检测系统及方法

[57] 摘要

一种网络蠕虫检测系统及方法，根据网络中主机向以前未连接过的 IP 地址发起的第一次连接的成功或失败情况和主机本次发起第一次连接与上次发起第一次连接的时间间隔，分别选择相应的概率计算公式计算主机感染蠕虫的概率；将计算所得概率值与预先设定的主机感染蠕虫判断阈值进行比较，若概率值大于主机感染蠕虫判断阈值，则认为该主机为异常主机。 若无法判断时，等待主机的下一个第一次连接，并将本次计算所得的条件概率作为下一次计算的先验概率，重新计算该主机感染蠕虫的概率。 本发明克服了现有的网络蠕虫检测技术不能准确、高效地检测到未知蠕虫和蠕虫病毒变种，检测结果存在较高的误报率和漏报率的问题，能准确、高效地检测到未知蠕虫和蠕虫病毒变种。



1、一种网络蠕虫检测方法，其特征在于，包括以下步骤：

根据网络中主机向以前未连接过的IP地址发起的第一次连接的成功或失败情况和主机本次发起第一次连接与上次发起第一次连接的时间间隔，分别选择相应的概率计算公式计算主机感染蠕虫的概率；

将计算所得概率值与预先设定的主机感染蠕虫判断阈值进行比较，若概率值大于主机感染蠕虫判断阈值，则认为该主机为异常主机。

2、根据权利要求1所述的网络蠕虫检测方法，其特征在于：还设置正常主机的判断阈值，若计算所得概率值小于正常主机判断阈值，则认为该连接的源地址主机为正常主机。

3、根据权利要求2所述的网络蠕虫检测方法，其特征在于：若无法判断该主机是否为感染蠕虫的异常主机或者为正常主机时，等待主机的下一个第一次连接，并将本次计算所得的条件概率作为下一次计算的基础，重新计算该主机感染蠕虫的概率。

4、根据权利要求1、2或3所述的网络蠕虫检测方法，其特征在于：所述的概率计算公式采用贝叶斯公式。

5、根据权利要求4所述的网络蠕虫检测方法，其特征在于：在第一次连接成功，且主机本次第一次连接与上次第一次连接的时间间隔大于预设时间时，所述主机感染蠕虫的条件概率计算公式为：

$$Prob(W|S,Itv>t)=\frac{Prob(W)*Prob(Itv>t|W)*Prob(S|W)}{Prob(W)*Prob(Itv>t|W)*Prob(S|W)+Prob(B)*Prob(Itv>t|B)*Prob(S|B)}$$

其中：

Prob（W）：主机感染蠕虫的先验概率，以前次第一次连接计算所得的条件概率作为本次计算的先验概率；

Prob（Itv>t|W)：预设的感染蠕虫的主机第一次连接时间间隔大于预设时间 t 的概率；

Prob（S|W)：预设的感染蠕虫的主机出现第一次连接成功的概率；

Prob（B）= 1-Prob（W)：主机未感染蠕虫的先验概率；

Prob（Itv>t|B)：预设的正常主机第一次连接时间间隔大于预设时间 t 的概率；

Prob（S|B)：预设的正常主机出现第一次连接成功的概率。

6、根据权利要求 4 所述的网络蠕虫检测方法，其特征在于：在第一次连接成功，且主机本次第一次连接与上次第一次连接的时间间隔不大于预设时间时，所述主机感染蠕虫的条件概率计算公式为：

$$Prob(W|S,Itv<=t)=\frac{Prob(W)*Prob(Itv<=t|W)*Prob(S|W)}{Prob(W)*Prob(Itv<=t|W)*Prob(S|W)+Prob(B)*Prob(Itv<=t|B)*Prob(S|B)}$$

其中：

Prob（W)：主机感染蠕虫的先验概率，以前次第一次连接计算所得的条件概率作为本次计算的先验概率；

Prob（Itv<=t|W)：预设的感染蠕虫的主机第一次连接时间间隔不大于预设时间 t 的概率；

Prob（S|W)：预设的感染蠕虫的主机出现第一次连接成功的概率；

Prob（B）= 1-Prob（W)：主机未感染蠕虫的先验概率；

Prob（Itv<=t|B)：预设的正常主机第一次连接时间间隔不大于预设时间 t 的概率；

Prob（S|B)：预设的正常主机出现第一次连接成功的概率。

7、根据权利要求 4 所述的网络蠕虫检测方法，其特征在于：在第一次连接失败，且主机本次第一次连接与上次第一次连接的时间间隔大于预设时间时，所述主机感染蠕虫的条件概率计算公式为：

$$Prob(W|F,Itv>t)=\frac{Prob(W)*Prob(Itv>t|W)*Prob(F|W)}{Prob(W)*Prob(Itv>t|W)*Prob(F|W)+Prob(B)*Prob(Itv>t|B)*Prob(F|B)}$$

其中：

Prob（W）：主机感染蠕虫的先验概率，以前次第一次连接计算所得的条件概率作为本次计算的先验概率；

Prob（Itv>t|W）：预设的感染蠕虫的主机第一次连接时间间隔大于预设时间 t 的概率；

Prob（F|W）：预设的感染蠕虫的主机出现第一次连接失败的概率；

Prob（B）= 1-Prob（W）：主机未感染蠕虫的先验概率；

Prob（Itv>t|B）：预设的正常主机第一次连接时间间隔大于预设时间 t 的概率；

Prob（F|B）：预设的正常主机出现第一次连接失败的概率。

8、根据权利要求 4 所述的网络蠕虫检测方法，其特征在于：在第一次连接失败，且主机本次第一次连接与上次第一次连接的时间间隔不大于预设时间时，所述主机感染蠕虫的条件概率计算公式为：

$$Prob(W|F,Itv<=t)=\frac{Prob(W)*Prob(Itv<=t|W)*Prob(F|W)}{Prob(W)*Prob(Itv<=t|W)*Prob(F|W)+Prob(B)*Prob(Itv<=t|B)*Prob(F|B)}$$

其中：

Prob（W）：主机感染蠕虫的先验概率，以前次第一次连接计算所得的条件概率作为本次计算的先验概率；

Prob（Itv<=t|W）：预设的感染蠕虫的主机第一次连接时间间隔不大于预设时间 t 的概率；

Prob（F|W）：预设的感染蠕虫的主机出现第一次连接失败的概率；

Prob（B）= 1-Prob（W）：主机未感染蠕虫的先验概率；

Prob（Itv<=t|B）：预设的正常主机第一次连接时间间隔不大于预设时

间 t 的概率；

Prob（F|B）：预设的正常主机出现第一次连接失败的概率。

9、根据权利要求 4 所述的网络蠕虫检测方法，其特征在于：在第一次检测到主机的第一次连接时，如果连接成功，则所述主机感染蠕虫的条件概率计算公式为：

$$Prob(W|S)=\frac{Prob(W)*Prob(S|W)}{Prob(W)*Prob(S|W)+Prob(B)*Prob(S|B)}$$

如果连接失败，则所述主机感染蠕虫的条件概率计算公式为：

$$Prob(W|F)=\frac{Prob(W)*Prob(F|W)}{Prob(W)*Prob(F|W)+Prob(B)*Prob(F|B)}$$

其中：

Prob（W）：主机感染蠕虫的先验概率，预设初始值；

Prob（S|W）：预设的感染蠕虫的主机出现第一次连接成功的概率；

Prob（B）= 1-Prob（W）：主机未感染蠕虫的先验概率；

Prob（S|B）：预设的正常主机出现第一次连接成功的概率；

Prob（F|W）：预设的感染蠕虫的主机出现第一次连接失败的概率；

Prob（F|B）：预设的正常主机出现第一次连接失败的概率。

10、根据权利要求 4 所述的网络蠕虫检测方法，其特征在于：进行网络蠕虫检测的步骤包括：

a）设置以下参数：感染蠕虫的主机出现第一次连接失败的概率、正常主机出现第一次连接失败的概率、主机感染蠕虫判断阈值、正常主机判断阈值、主机感染蠕虫的初始概率、正常主机第一次连接时间间隔大于预设时间的概率、正常主机第一次连接时间间隔小于等于预设时间的概率、感染蠕虫的主机第一次连接时间间隔大于预设时间的概率、感染蠕虫的主机第一次连接时间间隔小于等于预设时间的概率；

b）从网络上获取数据包，获取一个连接的源地址、目标地址、连接状态；

c）检查源地址是否已经在蠕虫列表中，若存在，说明该地址的主机已经

感染蠕虫，则对该连接不作进一步的处理，转到 b)；若不在，则转到 d)；

d）检查该源地址是否是第一次出现；如果是则转到 e)；否则转到 f)；

e）判断该连接的连接状态，根据该连接是成功连接还是失败连接，计算该连接的后验概率；并记录该连接到达时间；然后转到 b)，监视下一个连接；

f）判断该连接的源地址和目标地址对是否第一次出现，如果不是，说明该连接不是第一次连接，则对该连接不作进一步处理，转到 b)；否则转到 g)；

g）判断该连接的连接状态，如果该连接是成功连接，则转到 h)，如果是失败连接，则转到 i)；

h）判断连接到达的时间间隔是否超过预设时间，分别计算主机感染蠕虫的条件概率；如果条件概率小于正常主机判断阈值，则认为该连接的源地址为正常主机，将其概率重新设置为初始预设概率值，继续对该主机的下一个连接进行监视；如果大于正常主机判断阈值，则将本次计算所得的条件概率作为下一次计算的先验概率，然后转到 b)，监视下一个连接；

i）判断连接到达的时间间隔是否超过预设时间，分别计算主机感染蠕虫的条件概率；如果条件概率大于主机感染蠕虫判断阈值，则该主机被认为已经感染了蠕虫，并将该主机加入蠕虫列表中；如果条件概率小于主机感染蠕虫判断阈值，则将本次计算所得的条件概率作为下一次计算的先验概率，然后转到 b)，监视下一个连接。

11、一种网络蠕虫检测系统，其特征在于：包括检测机，所述检测机连接网络出口交换机或路由器的镜像端口，用于监听网络出口处的网络数据包，并运行蠕虫检测端程序，从网络中捕获数据包，根据网络中主机向以前未连接过的 IP 地址发起的第一次连接的成功或失败情况和主机本次发起第一次连接与上次发起第一次连接的时间间隔，分别选择相应的概率值计算公式计算主机感染蠕虫的概率，判断是否存在感染蠕虫的主机。

12、根据权利要求 11 所述的网络蠕虫检测系统，其特征在于：所述的检测机至少包括相互连接的网络流量分析模块和蠕虫检测模块；

所述网络流量分析模块通过网络接口监听网络上的数据包，从中提取连接信息；

所述网络蠕虫检测模块利用蠕虫检测算法对网络流量分析模块产生的连接信息进行分析，判断网络中是否存在网络蠕虫。

13、根据权利要求 11 或 12 所述的网络蠕虫检测系统，其特征在于：网络蠕虫检测系统还包括与所述检测机连接的管理机和数据库服务器，所述管理机用于指定检测机的检测策略，所述检测机对感染蠕虫的主机生成报警记录写入所述数据库服务器中。

14、根据权利要求 13 所述的网络蠕虫检测系统，其特征在于：所述的管理机包括策略配置模块，以及与策略配置模块连接的检测端连接模块、数据库连接模块，所述检测端连接模块、数据库连接模块分别用于与所述检测端程序、所述数据库服务器建立连接；

所述策略配置模块使用户根据实际检测情况通过所述检测端连接模块对所述检测机的检测参数进行修改，通知所述检测机更新策略信息，并通过所述数据库连接模块更新所述数据库服务器中的策略配置数据。

15、根据权利要求 14 所述的网络蠕虫检测系统，其特征在于：所述的管理机还包括与所述数据库连接模块连接的实时报警模块，实时报警模块通过数据库连接模块从所述数据库服务器中提取报警记录信息，并通过用户界面进行显示。

16、根据权利要求 14 所述的网络蠕虫检测系统，其特征在于：所述的管理机还包括与所述数据库连接模块连接的报警查询模块，所述报警查询模块允许用户输入查询条件，从所述数据库服务器中检索符合条件的报警记录并显示。

一种网络蠕虫检测系统及方法

技术领域

本发明涉及计算机安全防护技术领域，尤其涉及一种网络蠕虫检测系统及方法。

背景技术

网络蠕虫是一种可独立运行的程序，它通过对网络的扫描，发现存在系统漏洞的计算机系统，获取该计算机系统的控制权，并进行传播；网络蠕虫的传播会导致信息的泄露，计算机系统的资源消耗，网络的拥塞等严重后果。

网络蠕虫的工作流程可以分为漏洞扫描、攻击、传染、事后处理四个阶段，如图 1 所示，蠕虫程序扫描到有漏洞的计算机系统后，将蠕虫主体迁移到目标主机。然后，蠕虫程序进入被感染的系统，对目标主机进行事后处理。同时，蠕虫程序生成多个副本，重复上述流程。

分析蠕虫的整个工作流程，可以归纳蠕虫的主要危害有：

a、主动入侵目标系统

蠕虫利用系统漏洞对目标系统进行主动攻击，从而可以获取目标系统的控制权限，对目标系统的安全造成威胁；同时，由于蠕虫在进行攻击时无需人为干预，因此其隐蔽性强，传播速度快。

b、造成网络拥塞

蠕虫在传播过程中，会对目标系统进行扫描，这就不可避免的产生大量的网络数据流量；同时蠕虫在向不同目标系统发出的攻击数据也会产生大量的网络数据流量，因此蠕虫的爆发很容易导致整个网络的瘫痪，造成巨大的经济损失。

c、消耗系统资源

蠕虫入侵目标系统后，会在被感染的计算机上产生多个副本，每个副本会自动搜索新的攻击目标，因此会大量消耗被感染计算机的系统资源，导致系统性能的下降。

由于网络蠕虫存在上述危害，因此已经成为目前影响网络安全的一个重大因素。

通过分析蠕虫的工作过程和行为特征，可以知道防止蠕虫泛滥的关键在于及早的发现蠕虫，然后对被感染的计算机系统采取相应措施，如清除病毒文件、隔离等。因此对蠕虫的检测成为防止蠕虫传播的首要步骤，对蠕虫检测技术的研究成为保证网络环境安全性，维护社会和个人利益的迫切需要。

目前对于网络蠕虫的检测主要还是通过传统的基于特征码的检测，首先分析捕获的蠕虫样本，得到该蠕虫的特征码，更新蠕虫检测软件的特征库，然后蠕虫检测程序根据这些新的特征码在网络流量或者主机文件中进行特征匹配，从而实现蠕虫检测。这种检测方法的缺点是如果有新的蠕虫出现，需要经过一段时间才能使特征库得到更新。这样对于新出现的蠕虫或者蠕虫的变种就不能做到及时发现。

目前蠕虫检测方法研究的主流方向是通过对蠕虫传播过程中的导致的网络异常特征的分析，来检测蠕虫的爆发。常用的方法有通过对连接数的累计，判断累计值是否超过设置的阈值，来检测蠕虫；通过对 ICMP 消息异常的统计来检测蠕虫的发生等。但是上述方法由于缺乏对蠕虫传播特性的建模，检测策略简单，导致检测的误报率和漏报率较高。

现有技术列举如下：

一、现有技术一

现有技术一公开了一种入侵检测方法，入侵检测系统按照检测规则对访问受护网络或主机的每个事件进行检测，还包括以下步骤：a）判断当前检测到的事件是否为入侵事件，如果不是，返回步骤 a 继续检测下一个事件，如

果是，则取得检测当前事件所使用的入侵检测规则；b）根据入侵检测规则与漏洞之间预先设置的对应关系确定当前检测到的入侵事件所要攻击的漏洞；c）在该受护网络或主机进行漏洞扫描得到的漏洞扫描结果库中检索该受护网络或主机是否存在步骤 b 所确定的漏洞；并根据入侵事件的危害度和所要攻击的漏洞与漏洞扫描结果库间存在漏洞信息的匹配度进行入侵事件风险度评估。

现有技术一的缺点在于：

该方法对于通过检测每个访问事件，产生该事件对应的入侵规则，再将该规则与预先设置的漏洞事件库进行匹配，来判断入侵事件，其本质上还是基于规则的检测方法。基于规则的入侵检测方法，需要依赖预先设置的检测规则的精度和密度，而检测规则的设置总是在入侵事件发生后才能得到更新，所以对于新出现的攻击会有很高的漏报率。

二、现有技术二

现有技术二公开了一种检测蠕虫病毒方法，该方法利用设置在网络中的病毒监测程序监测任一与网络联接的计算机与其它联网计算机的连接数量，并设置阀值限制其连接数量，对超过阀值的连接将强行丢弃，并向入侵检测系统 IDS 发出报警。其步骤包括：

a）截获与网络连接的计算机应用层到 TCP/IP 核心的数据；

b）分析目的地址，统计该机与其它计算机的连接，包括该机发起到其它计算机的 TCP 连接和该机发送的 UDP 包；

c)判断目的地址是不是入侵检测系统 IDS 的地址，对到入侵检测系统 IDS 的数据，则转发到网络接口；

d）对不是到入侵检测系统 IDS 的数据，判断是否超过规定的阈值，超过阈值则将数据丢弃，并向入侵检测系统 IDS 发出报警信息，否则转发到网络接口。

现有技术二的缺点在于：

该蠕虫检测方法是通过判断连接数量是否超过阈值来确定蠕虫。这种判断方法由于是通过统计在一定时间段内的连接数量，会产生较高的误报率和漏报率。如果正常主机比较频繁地访问网络，而检测阈值又设置得较低，则会产生大量的误报；如果将检测阈值设得较高，则对于扫描速度比较慢的蠕虫，在寻找目标系统的时候，在一段时间内不会产生大量的连接，使用该方法会产生漏报。同时，如果蠕虫的开发者获知连接数阈值，则可以通过调整蠕虫的程序，来控制一定时间段内的蠕虫发送的连接请求数，来回避检测规则，从而会导致漏报。此外，该方法在实施时，需要在网络内的每一台计算机上安装一个病毒检测程序，用于监视从本计算机发起到其它计算机的连接。这种实施方案加大了成本投入，实际部署也会遇到困难。

三、现有技术三

现有技术三提出在大规模高速网络中，入侵检测系统采用分层的分布式结构，通过分散采集、分布处理和集中管理，满足了大规模高速网络的需求。在一个大型网络中可以配置多个入侵检测系统，每个入侵检测系统负责网络的一部分，还可以配置一些如防火墙等其它安全部件。为了获得入侵的全局视图，要求这些安全部件能够协同工作。该专利通过“聚类—合并—关联”三个步骤实现对告警的融合，产生大规模环境下的告警，同时提高单个入侵检测的检测率，降低它们的虚警率，最终为安全管理人员提供简练精确的告警。

现有技术三的缺点在于：

该技术对入侵检测系统对整个大规模环境的入侵检测作了说明，但是其在提高单个入侵检测的准确率方面没有作更大的改进。

四、现有技术四

现有技术四提出了一种基于相关特征聚类的层次入侵检测系统，其中控制台、数据收集模块、预处理模块、数据存储模块、响应模块、通信模块和事件分析模块中的检测器的结构组成、连接关系及功能都与现有的误用入侵

检测系统相同；其创新的关键之处是在事件分析模块中增设了由相关特征分析器、数据重组器和大类轮廓分析器构成的对初始化数据流进行相关特征分析、提取和重组的构件来替代原来的攻击轮廓分析器，从而构成一种新的层次入侵检测系统。该系统能够正确识别和检测新的攻击，较好地解决了现有的误用检测系统无法检测新的攻击方式和检测概率较低的缺陷，为保障计算机系统、网络系统及整个信息基础设施的安全的一种新技术装备。

现有技术四的缺点在于：

该技术对于入侵检测技术本身仍然采用了误用检测系统的方法，所以其本质上还是一种基于规则的入侵检测技术。

五、现有技术五

现有技术五提出一种高性能网络入侵检测系统和检测方法，其检测系统由一台转发器、至少一台交换机和多台检测引擎通过信息传输线路连接组成，在转发器内安装有分流系统。其检测方法对一个数据包的处理过程包括抓取数据包、修改目的 MAC 地址、转发数据包、分流数据包和入侵检测等步骤。该发明的高性能网络入侵检测系统和检测方法实行了数据分流，提高了检测性能，部署方便、具有良好的伸缩性，易于使用和维护，性价比高。适用于各种需要多台检测引擎才能实现高性能入侵检测的高速网络。

现有技术五的缺点在于：

该技术提出了利用多个检测引擎来实现高速网络的入侵检测，但是其对于入侵检测技术的本身没有提出更新的方法。

六、现有技术六

Bakos,George （ISTS Dartmouth College）;Berk，Vincent H.. Early detection of internet worm activity by metering ICMP destination unreachable messages. Proceedings of SPIE- The International Society for Optical Engineering, v4708, 2002, p33-42.

Bakos 等在上述文献中提出了一种新的蠕虫检测方法，利用 ICMP 目标主

机不可到达消息来识辨蠕虫的随机扫描行为。无论是在蠕虫真正传播前，还是在蠕虫传播过程中，蠕虫都会扫描许多随机的主机地址，当目标主机不存在的时候，网络中的路由器会返回给发出连接请求的机器一个 ICMP 目标主机不可到达消息，这个消息包含了源地址、源端口、目标地址、目标端口、网络协议等信息。通过部署一个专门的中央收集点，来收集互联网中许多路由器产生这种 ICMP 目标主机不可到达消息，然后再通过分析器来集中统计分析这些消息，按源地址、目标地址来统计消息的个数，并与设定的参数值和阈值进行比较，来判断蠕虫。

现有技术六的缺点在于：

这种方法对蠕虫检测的精度很大程度依赖于参与蠕虫检测的路由器的多少。如果路由器的个数不多，那么收集到的 ICMP 目标主机不可到达消息会比较少，会影响蠕虫检测的效果和效率。然而，如果参与的路由器比较多，则可能会因产生大量的 ICMP 消息而导致收集点的网络发生拥塞，影响收集点的正常工作。

七、现有技术七

Jaeyeon Jung, Vern Paxson, Arthur W. Berger, and Hari Balakrishnan. Fast portscan detection using sequential hypothesis testing. In Proceedings of the IEEE Symposium on Security and Privacy, 2004.

Jaeyeon Jung 等提出了一种利用连续假设检验来检测扫描的 TRW 检测技术。蠕虫在传播过程中，蠕虫主机会对许多随机地址发起连接请求，这些连接请求可能会得到回应，可能得不到回应，蠕虫主机发起的连接往往是失败连接。而正常主机发起连接能够成功建立的可能性更大。该技术检测被检测网络中每台主机的每个连接的状态，判断其是成功还是失败，采用连续假设检验的方法对失败连接的次数与成功连接的次数做比较，如果失败连接的次数比成功连接次数多指定的次数，则判断该主机存在异常。

现有技术七的缺点在于：

该技术实质上是通过比较失败连接次数与成功连接次数来判断主机是否存在异常。但是，在一定时间段内，正常主机可能也会产生更多的失败连接，因此该技术会产生较高的误报率。

发明内容

本发明所要解决的技术问题是：克服现有的网络蠕虫检测技术不能准确、高效地检测到未知蠕虫和蠕虫病毒变种，检测结果存在较高的误报率和漏报率的问题，提出一种网络蠕虫检测系统及方法，准确、高效地检测到未知蠕虫和蠕虫病毒变种。

本发明为解决上述技术问题所采用的技术方案为：

一种网络蠕虫检测方法，包括以下步骤：

根据网络中主机向以前未连接过的 IP 地址发起的第一次连接的成功或失败情况和主机本次发起第一次连接与上次发起第一次连接的时间间隔，分别选择相应的概率计算公式计算主机感染蠕虫的概率；

将计算所得概率值与预先设定的主机感染蠕虫判断阈值进行比较，若概率值大于主机感染蠕虫判断阈值，则认为该主机为异常主机。

还设置正常主机的判断阈值，若计算所得概率值小于正常主机判断阈值，则认为该连接的源地址主机为正常主机。

若无法判断该主机是否为感染蠕虫的异常主机或者为正常主机时，等待主机的下一个第一次连接，并将本次计算所得的条件概率作为下一次计算的基础，重新计算该主机感染蠕虫的概率。

所述的概率计算公式可以采用贝叶斯公式。

在第一次连接成功，且主机本次第一次连接与上次第一次连接的时间间隔大于预设时间时，所述主机感染蠕虫的条件概率计算公式为：

$$Prob(W|S,Itv>t)=\frac{Prob(W)*Prob(Itv>t|W)*Prob(S|W)}{Prob(W)*Prob(Itv>t|W)*Prob(S|W)+Prob(B)*Prob(Itv>t|B)*Prob(S|B)}$$

其中：

Prob（W）：主机感染蠕虫的先验概率，以前次第一次连接计算所得的条件概率作为本次计算的先验概率；

Prob（Itv>t|W）：预设的感染蠕虫的主机第一次连接时间间隔大于预设时间 t 的概率；

Prob（S|W）：预设的感染蠕虫的主机出现第一次连接成功的概率；

Prob（B）= 1-Prob（W）：主机未感染蠕虫的先验概率；

Prob（Itv>t|B）：预设的正常主机第一次连接时间间隔大于预设时间 t 的概率；

Prob（S|B）：预设的正常主机出现第一次连接成功的概率。

在第一次连接成功，且主机本次第一次连接与上次第一次连接的时间间隔不大于预设时间时，所述主机感染蠕虫的条件概率计算公式为：

$$Prob(W|S,Itv<=t)=\frac{Prob(W)*Prob(Itv<=t|W)*Prob(S|W)}{Prob(W)*Prob(Itv<=t|W)*Prob(S|W)+Prob(B)*Prob(Itv<=t|B)*Prob(S|B)}$$

其中：

Prob（W）：主机感染蠕虫的先验概率，以前次第一次连接计算所得的条件概率作为本次计算的先验概率；

Prob（Itv<=t|W）：预设的感染蠕虫的主机第一次连接时间间隔不大于预设时间 t 的概率；

Prob（S|W）：预设的感染蠕虫的主机出现第一次连接成功的概率；

Prob（B）= 1-Prob（W）：主机未感染蠕虫的先验概率；

Prob（Itv<=t|B）：预设的正常主机第一次连接时间间隔不大于预设时间 t 的概率；

Prob（S|B）：预设的正常主机出现第一次连接成功的概率。

在第一次连接失败，且主机本次第一次连接与上次第一次连接的时间间

隔大于预设时间时，所述主机感染蠕虫的条件概率计算公式为：

$$Prob(W|F,Itv>t)=\frac{Prob(W)*Prob(Itv>t|W)*Prob(F|W)}{Prob(W)*Prob(Itv>t|W)*Prob(F|W)+Prob(B)*Prob(Itv>t|B)*Prob(F|B)}$$

其中：

Prob（W）：主机感染蠕虫的先验概率，以前次第一次连接计算所得的条件概率作为本次计算的先验概率；

Prob（Itv>t|W）：预设的感染蠕虫的主机第一次连接时间间隔大于预设时间 t 的概率；

Prob（F|W）：预设的感染蠕虫的主机出现第一次连接失败的概率；

Prob（B）= 1-Prob（W）：主机未感染蠕虫的先验概率；

Prob（Itv>t|B）：预设的正常主机第一次连接时间间隔大于预设时间 t 的概率；

Prob（F|B）：预设的正常主机出现第一次连接失败的概率。

在第一次连接失败，且主机本次第一次连接与上次第一次连接的时间间隔不大于预设时间时，所述主机感染蠕虫的条件概率计算公式为：

$$Prob(W|F,Itv<=t)=\frac{Prob(W)*Prob(Itv<=t|W)*Prob(F|W)}{Prob(W)*Prob(Itv<=t|W)*Prob(F|W)+Prob(B)*Prob(Itv<=t|B)*Prob(F|B)}$$

其中：

Prob（W）：主机感染蠕虫的先验概率，以前次第一次连接计算所得的条件概率作为本次计算的先验概率；

Prob（Itv<=t|W）：预设的感染蠕虫的主机第一次连接时间间隔不大于预设时间 t 的概率；

Prob（F|W）：预设的感染蠕虫的主机出现第一次连接失败的概率；

Prob（B）= 1-Prob（W）：主机未感染蠕虫的先验概率；

Prob（Itv<=t|B）：预设的正常主机第一次连接时间间隔不大于预设时

间 t 的概率；

Prob（F|B）：预设的正常主机出现第一次连接失败的概率。

在第一次检测到主机的第一次连接时，如果连接成功，则所述主机感染蠕虫的条件概率计算公式为：

$$Prob(W|S)=\frac{Prob(W)*Prob(S|W)}{Prob(W)*Prob(S|W)+Prob(B)*Prob(S|B)}$$

如果连接失败，则所述主机感染蠕虫的条件概率计算公式为：

$$Prob(W|F)=\frac{Prob(W)*Prob(F|W)}{Prob(W)*Prob(F|W)+Prob(B)*Prob(F|B)}$$

其中：

Prob（W）：主机感染蠕虫的先验概率，预设初始值；

Prob（S|W）：预设的感染蠕虫的主机出现第一次连接成功的概率；

Prob（B）= 1-Prob（W）：主机未感染蠕虫的先验概率；

Prob（S|B）：预设的正常主机出现第一次连接成功的概率；

Prob（F|W）：预设的感染蠕虫的主机出现第一次连接失败的概率；

Prob（F|B）：预设的正常主机出现第一次连接失败的概率。

进行网络蠕虫检测的步骤包括：

a）设置以下参数：感染蠕虫的主机出现第一次连接失败的概率、正常主机出现第一次连接失败的概率、主机感染蠕虫判断阈值、正常主机判断阈值、主机感染蠕虫的初始概率、正常主机第一次连接时间间隔大于预设时间的概率、正常主机第一次连接时间间隔小于等于预设时间的概率、感染蠕虫的主机第一次连接时间间隔大于预设时间的概率、感染蠕虫的主机第一次连接时间间隔小于等于预设时间的概率；

b）从网络上获取数据包，获取一个连接的源地址、目标地址、连接状态；

c）检查源地址是否已经在蠕虫列表中，若存在，说明该地址的主机已经感染蠕虫，则对该连接不作进一步的处理，转到 b）；若不在，则转到 d）；

d）检查该源地址是否是第一次出现，如果是则转到 e）；否则转到 f）；

e）判断该连接的连接状态，根据该连接是成功连接还是失败连接，计算该连接的后验概率；并记录该连接到达时间；然后转到 b），监视下一个连接；

f）判断该连接的源地址和目标地址对是否第一次出现，如果不是，说明该连接不是第一次连接，则对该连接不作进一步处理，转到 b）；否则转到 g）；

g）判断该连接的连接状态，如果该连接是成功连接，则转到 h），如果是失败连接，则转到 i）；

h）判断连接到达的时间间隔是否超过预设时间，分别计算主机感染蠕虫的条件概率；如果条件概率小于正常主机判断阈值，则认为该连接的源地址为正常主机，将其概率重新设置为初始预设概率值，继续对该主机的下一个连接进行监视；如果大于正常主机判断阈值，则将本次计算所得的条件概率作为下一次计算的先验概率，然后转到 b），监视下一个连接；

i）判断连接到达的时间间隔是否超过预设时间，分别计算主机感染蠕虫的条件概率；如果条件概率大于主机感染蠕虫判断阈值，则该主机被认为已经感染了蠕虫，并将该主机加入蠕虫列表中；如果条件概率小于主机感染蠕虫判断阈值，则将本次计算所得的条件概率作为下一次计算的先验概率，然后转到 b），监视下一个连接。

相应的一种网络蠕虫检测系统，包括检测机，所述检测机连接网络出口交换机或路由器的镜像端口，用于监听网络出口处的网络数据包，并运行蠕虫检测端程序，从网络中捕获数据包，根据网络中主机向以前未连接过的 IP 地址发起的第一次连接的成功或失败情况和主机本次发起第一次连接与上次发起第一次连接的时间间隔，分别选择相应的概率值计算公式计算主机感染蠕虫的概率，判断是否存在感染蠕虫的主机。

所述的检测机至少包括相互连接的网络流量分析模块和蠕虫检测模块；所述网络流量分析模块通过网络接口监听网络上的数据包，从中提取连接信息；所述网络蠕虫检测模块利用蠕虫检测算法对网络流量分析模块产生的连接信息进行分析，判断网络中是否存在网络蠕虫。

网络蠕虫检测系统还可以包括与所述检测机连接的管理机和数据库服务器，所述管理机用于指定检测机的检测策略，所述检测机对感染蠕虫的主机生成报警记录写入所述数据库服务器中。

所述的管理机包括策略配置模块，以及与策略配置模块连接的检测端连接模块、数据库连接模块，所述检测端连接模块、数据库连接模块分别用于与所述检测端、所述数据库服务器建立连接；所述策略配置模块使用户根据实际检测情况通过所述检测端连接模块对所述检测机的检测参数进行修改，通知所述检测机更新策略信息，并通过所述数据库连接模块更新所述数据库服务器中的策略配置数据。

所述的管理机还可以包括与所述数据库连接模块连接的实时报警模块，实时报警模块通过数据库连接模块从所述数据库服务器中提取报警记录信息，并通过用户界面进行显示。

所述的管理机还可以包括与所述数据库连接模块连接的报警查询模块，所述报警查询模块允许用户输入查询条件，从所述数据库服务器中检索符合条件的报警记录并显示。

本发明的有益效果为：本发明提供了一种网络蠕虫检测系统及方法，对蠕虫的检测是通过贝叶斯方法根据第一次连接的状态计算某主机的后验概率，比较该值与设定阈值之间的关系来判断该连接的源地址为正常主机还是蠕虫。本发明蠕虫检测方法并非简单地对某一时间段内连接数进行统计，不仅考虑了当前的蠕虫行为，也考虑历史状态对当前检测的影响，从而提高了蠕虫检测效率和精度；本发明蠕虫检测方法考察了网络蠕虫攻击行为的一般过程，针对的是蠕虫传播过程中的具有代表性的随机扫描行为的检测，因此可以更加全面地实现对未知网络蠕虫的检测；利用本发明能够精确检测到蠕虫主机，产生的误报记录更少。

为了验证本专利所研究的蠕虫检测技术的性能，从校园网络上采集了一个宿舍楼出口处的网络流量数据，采用包括本专利技术在内的四种不同的检

测技术进行蠕虫检测，并对检测结果进行了分析。

与本专利技术比较的检测技术包括：

1、检测技术一：即采用现有技术七的 TRW 算法，设置 $P_D = 0.9$，$P_F = 0.01$，$\theta_1 = 015$，$\theta_0 = 08$；

2、检测技术二：10 秒内失败连接数超过 15 次判断为异常，产生报警；

3、检测技术三：30 秒内失败连接数超过 15 次判断为异常，产生报警。

测试结果如表 1 所示：

表 1、蠕虫检测结果比较

	记录总数	有效报警记录			误报记录			报警有效率
		蠕虫	扫描 Web 服务器	扫描 FTP 服务器	P2P 共享软件	游戏软件	不能确定	
本专利技术	40	18	6	1	6	5	4	62.5%
检测技术一	70	18	7	1	19	12	13	37.1%
检测技术二	42	16	7	1	4	9	5	57.1%
检测技术三	50	18	6	1	6	9	10	50%

实验结果表明，本蠕虫检测方法是在四种检测技术里产生的误报记录最少的。由于扫描 Web 服务器和 FTP 服务器也属于存在威胁的网络行为，对它们的报警也是合理的，因此本次检测的报警有效率（有效报警记录/记录总数）可以达到 62.5%。检测技术二在检测蠕虫时会产生漏报，而且蠕虫可以很容易通过调整扫描策略来规避检测技术二和检测技术三的检测。虽然蠕虫比较难以规避检测技术一：TRW 算法的检测，但是该算法的检测有效率仅有 37.1%，误报过多，因此也不具有实用性。

本发明蠕虫检测系统只需要部署在企业内部网络的出口处，就可以实现对整个内部网络的蠕虫检测，这样节省了资源和产品升级、维护的费用。

附图说明

图 1 为蠕虫工作流程示意图；

图 2 为本发明蠕虫检测系统结构示意图；

图 3 为本发明蠕虫检测流程图；

图 4 为本发明检测端模块示意图；

图 5 为本发明控制管理端模块示意图。

具体实施方式

下面根据附图和实施例对本发明作进一步详细说明：

本发明通过对蠕虫本质特征的分析，提出一种蠕虫检测系统和方法，可以检测出企业内部网是否存在感染网络蠕虫的主机，并提供实时报警。

一、蠕虫检测系统

蠕虫检测系统部署方式如图 2 所示，蠕虫检测系统包括了检测机、数据库服务器、管理机；其中检测机安装两块网卡，其中一个网口连接企业内部网的出口交换机或路由器的镜像端口，将连接外部网的端口的流量映射到检测机上，用于监听企业内部网出口处的网络数据包，另一个网口连接包含数据库和管理机的网络，用于接受管理和发送报警信息。

检测机运行蠕虫检测端程序，执行管理机指定的检测策略，从实际网络中捕获数据包，采用蠕虫检测算法进行分析，判断是否存在感染蠕虫的主机，生成报警记录写入数据库中。管理机上运行管理端程序，可以调整检测机的检测策略，以及从数据库中查询报警记录。

二、蠕虫检测方法

基于随机扫描的网络蠕虫在进行传播时，会向随机地址发送大量的扫描数据包，以判断该地址的主机是否开机，以及是否可以被感染。然而，在 Internet 上，多数 IP 地址可能没有主机，或者主机没有开机，或者主机被隐藏在防火墙、NAT 等设备之后，因此由感染蠕虫的主机发起的一次扫描行为可以建立起一个连接的概率比较低，因此在连接成功概率这一特征上，网络蠕虫的扫描行为和正常主机的网络访问行为存在很明显的差异，因此可以利用这一特征作为检测随机扫描型蠕虫的依据。网络蠕虫为了达到在短时间内感染尽可

能多的目标系统的目的，它在传播时发送连接请求的时间间隔一般会很短，而正常主机在对外发起连接时，其间隔时间会比较长，因此，可以把连接请求发起时间间隔也作为其中的一个影响因素。

基于随机扫描的网络蠕虫为了实现在整个互联网范围内的传播，需要对整个 IP 地址段进行扫描，因此企业内部网中存在的蠕虫主机也必然会向企业外部的网络发起扫描行为。基于这个原理，只要将蠕虫检测机部署在企业内部网的出口处，就可以检测企业内部网是否存在网络蠕虫。

本发明以第一次连接（First Contact Connection，FCC）状态作为蠕虫行为判据的基准，综合考虑算法精确度和效率，采用贝叶斯公式作为决策算法进行网络蠕虫的检测。某 IP 地址的 FCC 是指以此 IP 地址为源地址，向以前未连接过的 IP 地址发起的请求的连接。在蠕虫检测中，需要考虑其行为的历史性，即通过主机一系列的行为对其是否为蠕虫或者正常主机进行判断。采用贝叶斯公式作为决策算法，可以保证在检测中充分考虑历史数据的影响，使检测结果更加精确，并且其具有成熟的理论基础，运算简单，能够保证很好的实时性。

贝叶斯公式如下式所示：

$$P(\omega_i \mid x) = \frac{p(x \mid \omega_i)P(\omega_i)}{p(x)}$$

其中：

$P(\omega_i \mid x)$ 为后验概率；

$p(x \mid \omega_i)$ 为概率密度；

$P(\omega_i)$ 为先验概率；

$p(x) = \sum_{i=1}^{c} p(x \mid \omega_i)P(\omega_i)$ 为 x 的总概率。

贝叶斯公式表明了在事件 ω_i 的发生概率为 $P(\omega_i)$ 的前提下，在事件 ω_i 发生的条件下 x 发生的条件概率为 $p(x \mid \omega_i)$，则在 x 发生的条件下事件 ω_i 发生的条件概率。

根据 TCP 连接标准定义，如果完成了三次握手的过程，则称该连接为一

个成功连接，否则为失败连接。对于一个 UDP 连接，如果源端发送了 UDP 请求包后，在超时时间间隔内获得了目标端的反馈包，则称该连接为成功连接，否则为失败连接。

首先假定某个主机感染了蠕虫的概率为 *Prob（W）*，其中 W 表示主机感染蠕虫的事件。如果当前该主机产生了一个 FCC，则需要根据这个 FCC 是失败连接还是成功连接，并且这次 FCC 与上次 FCC 的时间间隔是大于 0.2 秒还是小于等于 0.2 秒，来分别计算该主机感染了蠕虫的条件概率 *Prob（W|F, Itv>0.2）*、*Prob（W|F, Itv<=0.2）*、*Prob（W|S, Itv>02）*和 *Prob（W|S, Itv<=0.2）*，其中 F 表示该次 FCC 为失败连接，S 表示该次 FCC 为成功连接，Itv 表示这次 FCC 与上次 FCC 的时间间隔（单位为秒）。如果条件概率大于感染判断阈值，如 0.999995，则可以报警提示该主机已被蠕虫感染。如果条件概率小于正常判断阈值，如 0.0001，则可以认为该主机没有感染蠕虫。当该主机再次产生一个 FCC 时，可以将上一次计算出的条件概率 *Prob（W|F, Itv>02）*（或 *Prob（W|F, Itv<=02）* 或 *Prob（W|S, Itv>02）* 或 *Prob（W|S, Itv<=0.2）*）作为这次计算的先验概率 *Prob（W）*，重新计算该主机感染蠕虫的概率。

根据贝叶斯公式有：

公式（1）

$$Prob(W|S,Itv>0.2)=\frac{Prob(W)*Prob(Itv>0.2|W)*Prob(S|W)}{Prob(W)*Prob(Itv>0.2|W)*Prob(S|W)+Prob(B)*Prob(Itv>0.2|B)*Prob(S|B)}$$

公式（2）

$$Prob(W|S,Itv<=0.2)=\frac{Prob(W)*Prob(Itv<=0.2|W)*Prob(S|W)}{Prob(W)*Prob(Itv<=0.2|W)*Prob(S|W)+Prob(B)*Prob(Itv<=0.2|B)*Prob(S|B)}$$

公式（3）

$$Prob(W|F,Itv>0.2)=\frac{Prob(W)*Prob(Itv>0.2|W)*Prob(F|W)}{Prob(W)*Prob(Itv>0.2|W)*Prob(F|W)+Prob(B)*Prob(Itv>0.2|B)*Prob(F|B)}$$

公式（4）

$$Prob(W|F,Itv<=0.2)=\frac{Prob(W)*Prob(Itv<=0.2|W)*Prob(F|W)}{Prob(W)*Prob(Itv<=0.2|W)*Prob(F|W)+Prob(B)*Prob(Itv<=0.2|B)*Prob(F|B)}$$

其中，W 表示主机感染蠕虫，B 表示主机未感染蠕虫，为正常主机，F 表示该次 FCC 为失败连接，S 表示该次 FCC 为成功连接；Itv 为连接请求发

起的时间间隔；

Prob（W）：主机感染蠕虫的概率，初值取 0.5；

Prob（B） = 1-*Prob（W）*：主机未感染蠕虫的概率；

Prob（F|W）：感染蠕虫的主机出现失败 FCC 的概率；

Prob（F|B）：正常主机出现失败 FCC 的概率；

Prob（S|W） = 1- *Prob（F|W）*：感染蠕虫的主机出现成功 FCC 的概率；

Prob（S|B） = 1- *Prob（F|B）*：正常主机出现成功 FCC 的概率；

Prob（Itv>02|W）：感染蠕虫的 FCC 时间间隔大于 0.2 秒的概率

Prob（Itv<=0.2|W）：感染蠕虫的 FCC 时间间隔小于等于 0.2 秒的概率

Prob（Itv>02|B）：正常主机 FCC 时间间隔大于 0.2 秒的概率

Prob（Itv<=02|B）：正常主机 FCC 时间间隔小于等于 0.2 秒的概率

前两个公式表明了，当 FCC 成功时，将对该主机进行是否为正常主机的判断，当计算的条件概率小于正常主机判断阈值 β 时，则认为该主机是正常主机。后两个公式表明了，当 FCC 失败时，将对该主机进行是否为蠕虫的判断，当所计算出的条件概率大于感染判断阈值 α 时，则认为该主机已被感染。若无法判断该主机是否为蠕虫或者正常主机时，则等待下一个 FCC，并将本次计算所得的条件概率作为下一次计算的先验概率。

通过大量的数据分析后，得到了一些概率值：

a）感染蠕虫的主机出现失败 FCC 的概率：*Prob（F|W）* = 08；

b）正常主机出现失败 FCC 的概率：*Prob（F|B）* = 01；

c）正常主机 FCC 时间间隔大于 0.2 秒的概率：*Prob（Itv>02|B）*=0.7；

d）正常主机 FCC 时间间隔小于等于 0.2 秒的概率：*Prob（Itv<=02|B）* =0.3；

e）感染蠕虫的 FCC 时间间隔大于 0.2 秒的概率：*Prob（Itv>02|W）*=01；

f）感染蠕虫的 FCC 时间间隔小于等于 0.2 秒的概率：*Prob（Itv<=02|W）* =0.9；

g）主机感染蠕虫判断阈值为：α = 0999995；

h）正常主机判断阈值为：β = 00001；

检测算法的基本流程是：根据连接的成功（或失败）情况和连接时间间隔，选择相应的概率值计算公式，再把计算所得概率值与设定的判断阈值进行比较，若概率值大于主机感染蠕虫判断阈值，则认为该连接的源地址为异常主机，若概率值小于正常主机判断阈值，则认为该连接的源地址为正常主机，其它情况则用该概率值更新计算公式里的先验概率，继续考察下一次连接情况。

蠕虫检测算法具体流程图如图 3 所示，步骤如下：

a）设置参数，感染蠕虫的主机出现失败 FCC 的概率 *Prob（F|W）* = 08；正常主机出现失败 FCC 的概率 *Prob（F|B）* = 01；主机感染蠕虫判断阈值 α = 0999995；正常主机判断阈值 β =0.0001；主机感染蠕虫的初始概率 *Prob（W）* = 05；正常主机 FCC 时间间隔大于 0.2 秒概率 *Prob（Itv>0.2|B）* = 07；正常主机 FCC 时间间隔小于等于 0.2 秒概率 *Prob（Itv<=0.2|B）* = 03；感染蠕虫的主机 FCC 时间间隔大于 0.2 秒概率 *Prob（Itv>02|W）* = 0.1；感染蠕虫的主机 FCC 时间间隔小于等于 0.2 秒概率 *Prob（Itv<=02|W）* = 09；

b）从网络上抓包，获取一个连接的源地址 Orig，目标地址 Dest，连接状态（成功/失败）；

c）检查源地址 Orig 是否已经在蠕虫列表中，若存在，说明该地址的主机已经感染蠕虫，则对该连接不作进一步的处理，转到 b）；若不在，则转到 d）；

d）检查该 Orig 是第一次出现，如果是则转到 e）；否则转到 f）；

e）由于第一次接收到 Orig 的连接时还不存在 FCC 的时间间隔，所以需要采用以下公式计算第一次 FCC 的后验概率：

公式（5）

$$Prob(W|S)=\frac{Prob(W)*Prob(S|W)}{Prob(W)*Prob(S|W)+Prob(B)*Prob(S|B)}$$

公式（6）

$$Prob(W|F)=\frac{Prob(W)*Prob(F|W)}{Prob(W)*Prob(F|W)+Prob(B)*Prob(F|B)}$$

判断该连接的连接状态，如果该连接是成功连接，则按公式（5）计算Prob(W|S)，如果是失败连接，则按公式（6）计算Prob(W|F)，作为Prob(W)；并记录该连接到达时间；然后转到b)，监视下一个连接；

f）判断该连接是否为第一次连接，如果不是则转到b)；否则转到g)；

g）判断该连接的连接状态，如果该连接是成功连接，则转到h)，如果是失败连接，则转到i)；

h）判断连接到达的时间间隔是否超过设置的阈值（0.2秒），如果大于0.2秒，则按公式（1）计算 *Prob（W|SItv）*；如果小于等于0.2秒，则按公式（2）计算 *Prob（W|SItv）*；如果 *Prob（W|SItv）* $<\beta$，则该主机（Orig）被认为是正常主机，且其 *Prob（W）* 被重新设置为0.5；如果 *Prob（W|SItv）* $>\beta$ 则令 *Prob（W）= Pob（W|SItv）*；然后转到b)，监视下一个连接；

i）判断连接到达的时间间隔是否超过设置的阈值（0.2秒），如果大于0.2秒，则按公式（3）计算 *Prob（W|FItv）*；如果小于等于0.2秒，则按公式（4）计算 *Prob（W|FItv）*；如果 *Prob（W|FItv）* $>\alpha$ 则该主机（Orig）被认为已经感染了蠕虫，并将该主机（Orig）加入蠕虫列表中；如果 *Prob（W|FItv）* $<\alpha$ 则令 *Prob（W）= Pob（W|FItv）*；然后转到b)，监视下一个连接。

三、检测端（即检测机）的实现

检测端模块如图 4 所示，网络流量分析模块通过网络接口监听网络上的数据包，进行数据包重组、超时检测、数据分析，从中提取TCP连接和UDP通信的状态、流量信息，并实时更新，提供蠕虫检测模块感兴趣的连接信息。

网络蠕虫检测模块利用贝叶斯蠕虫检测算法对网络流量分析模块产生的连接信息进行分析，判断内部网中是否存在网络蠕虫，如果存在则通知数据库访问模块记录报警信息。

数据库访问模块建立并维持检测端与数据库之间的连接，将检测模块的报警记录保存在数据库中，或从数据库中读出管理端对检测端的配置策略。

配置模块根据从数据库中读出的配置策略，对检测端的各个模块进行参数配置。

通信模块实现检测端与管理端之间的相互通信，接收管理端对检测端的管理控制命令，如启停检测端、修改检测策略等。

控制模块根据管理端的控制命令，调度检测模块和数据库访问模块获取并执行控制命令和配置策略。

四、管理端（即管理机）的实现

如图 5 所示，管理端程序在 Windows 环境下实现，具有友好的用户界面；其中包括了图 5 所示的下列组成模块：

策略配置模块可以让用户根据实际检测情况对检测机的检测参数进行修改，并通过数据库连接模块更新数据库中的策略配置表数据，同时通过检测端连接模块通知检测端程序更新策略信息；

数据库连接模块用于与数据库进行连接；

检测端连接模块通过 Socket 与检测端程序建立连接，实现与指定检测机建立通讯；

实时报警模块通过数据库连接模块定时从数据库中提取最新的报警记录信息，并通过用户界面进行显示；

报警查询模块允许用户输入查询条件，从数据库中检索符合条件的报警记录，并显示。

本发明采用第一次连接（FCC）的成功和失败的比例的差异以及 FCC 的到达时间间隔分布作为判断随机扫描型蠕虫存在的依据，抓住了这类蠕虫行为的本质特征，因此有很高的检测精度；利用贝叶斯方法计算每个主机在 FCC 连接成功或失败情况下的条件概率，比较该概率值与预先设定阈值，判断其是否为蠕虫；本发明算法不仅考虑当前的蠕虫行为，而且考虑了蠕虫历史活动行为，提高了蠕虫检测的精度。

本领域技术人员不脱离本发明的实质和精神，可以有多种变形方案实现

本发明，以上所述仅为本发明较佳可行的实施例而已，并非因此局限本发明的权利范围，凡运用本发明说明书及附图内容所作的等效变化，均包含于本发明的权利范围之内。

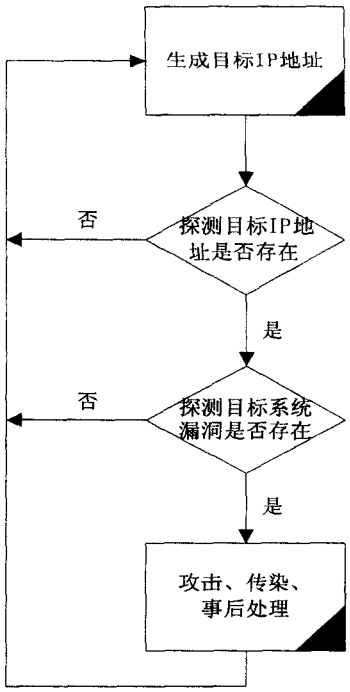

生成目标IP地址
探测目标IP地址是否存在
否
是
探测目标系统漏洞是否存在
否
是
攻击、传染、事后处理

图 1

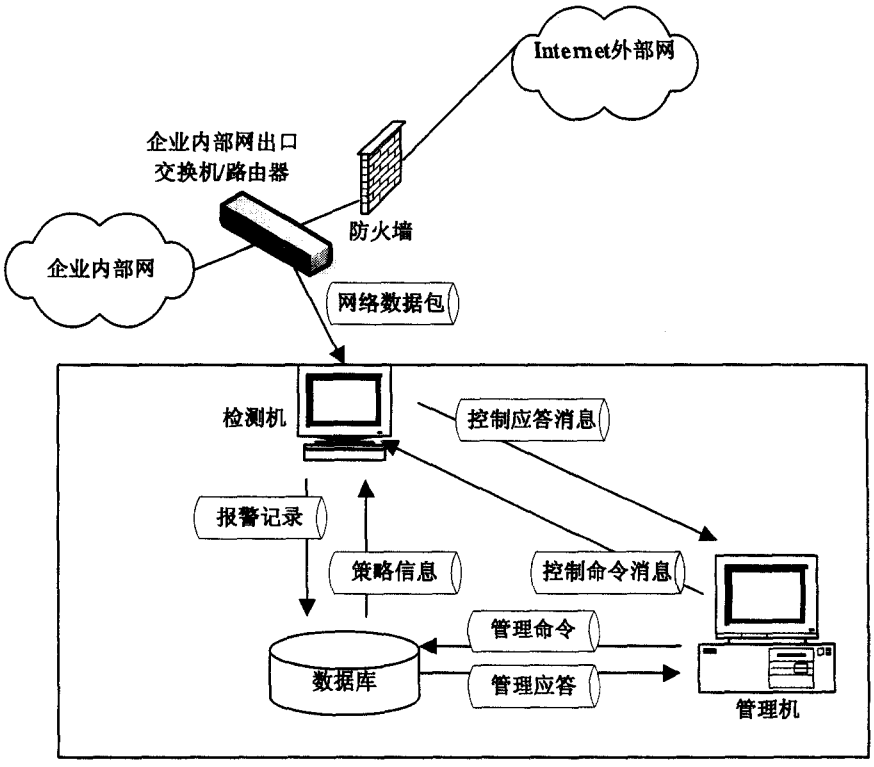

Internet外部网
企业内部网出口
交换机/路由器
防火墙
企业内部网
网络数据包
检测机
控制应答消息
报警记录
策略信息
控制命令消息
管理命令
数据库
管理应答
管理机

图2

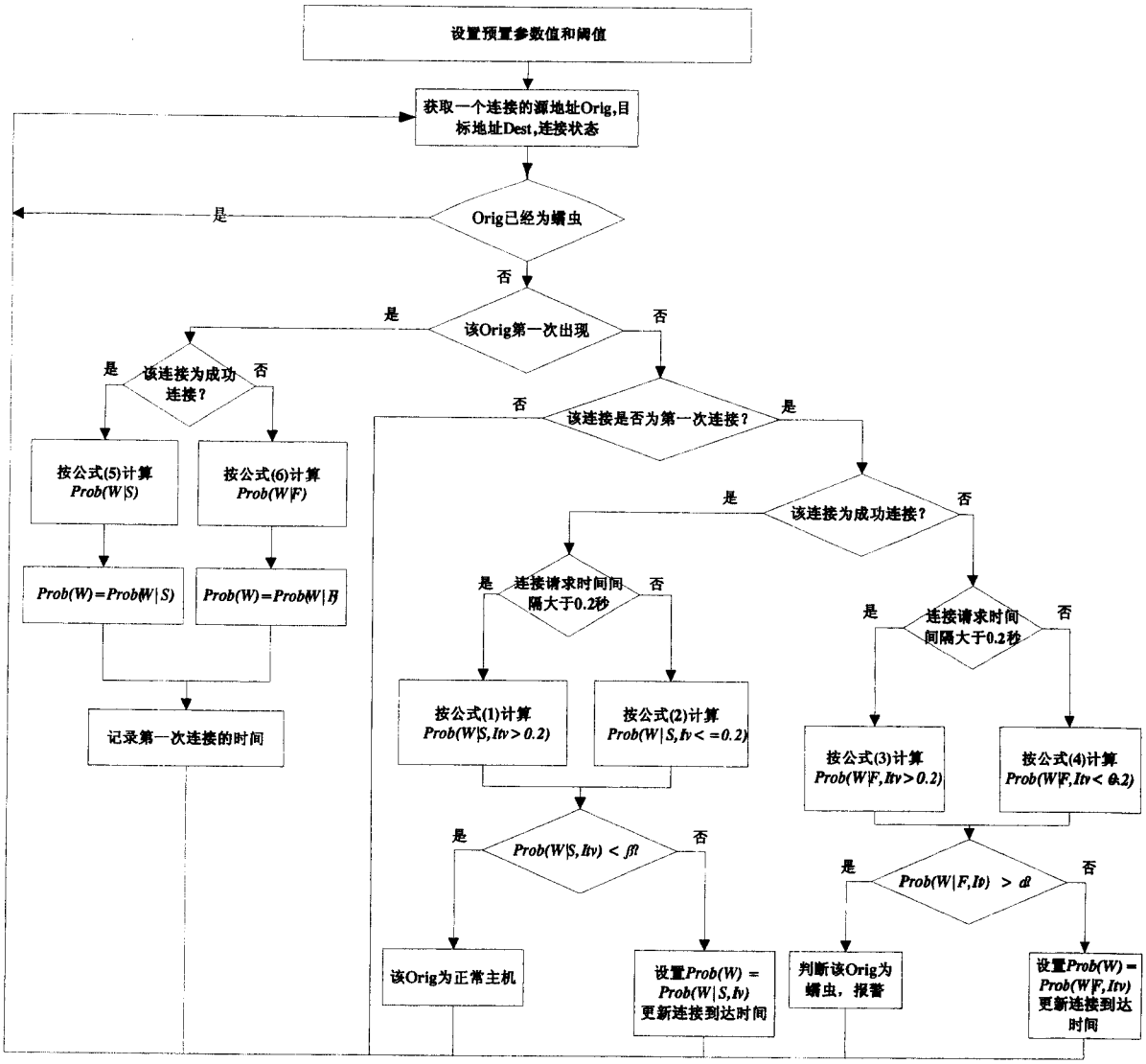

设置预置参数值和阈值
获取一个连接的源地址Orig,目标地址Dest,连接状态
Orig已经为蠕虫
是
否
该Orig第一次出现
是
否
该连接为成功连接?
是
否
按公式(5)计算 Prob(W|S)
按公式(6)计算 Prob(W|F)
Prob(W)=Prob(W|S)
Prob(W)=Prob(W|F)
记录第一次连接的时间
该连接是否为第一次连接?
是
否
该连接为成功连接?
是
否
连接请求时间间隔大于0.2秒
是
否
按公式(1)计算 Prob(W|S,Itv>0.2)
按公式(2)计算 Prob(W|S,Itv<=0.2)
连接请求时间间隔大于0.2秒
是
否
按公式(3)计算 Prob(W|F,Itv>0.2)
按公式(4)计算 Prob(W|F,Itv<=0.2)
Prob(W|S,Itv) < β1
是
否
Prob(W|F,Itv) > α2
是
否
该Orig为正常主机
设置Prob(W)=Prob(W|S,Itv)更新连接到达时间
判断该Orig为蠕虫，报警
设置Prob(W)=Prob(W|F,Itv)更新连接到达时间

图3

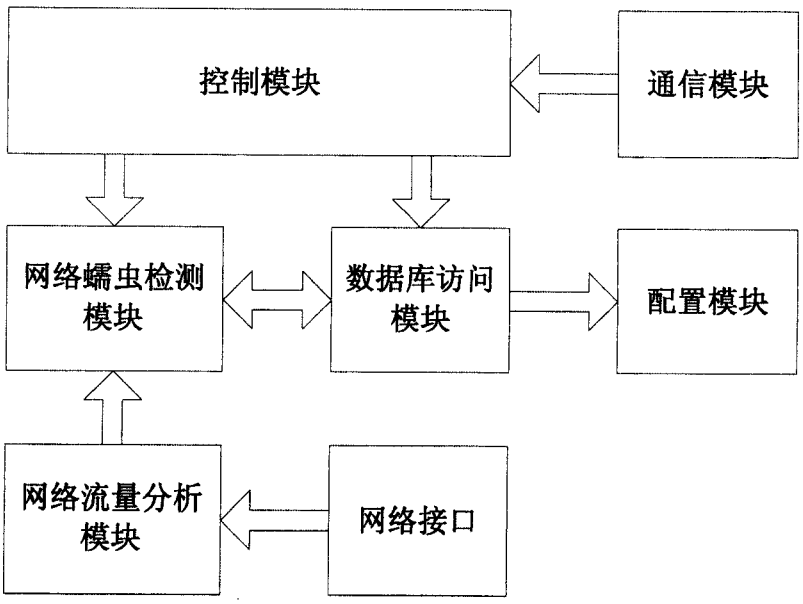

控制模块
通信模块
网络蠕虫检测模块
数据库访问模块
配置模块
网络流量分析模块
网络接口

图4

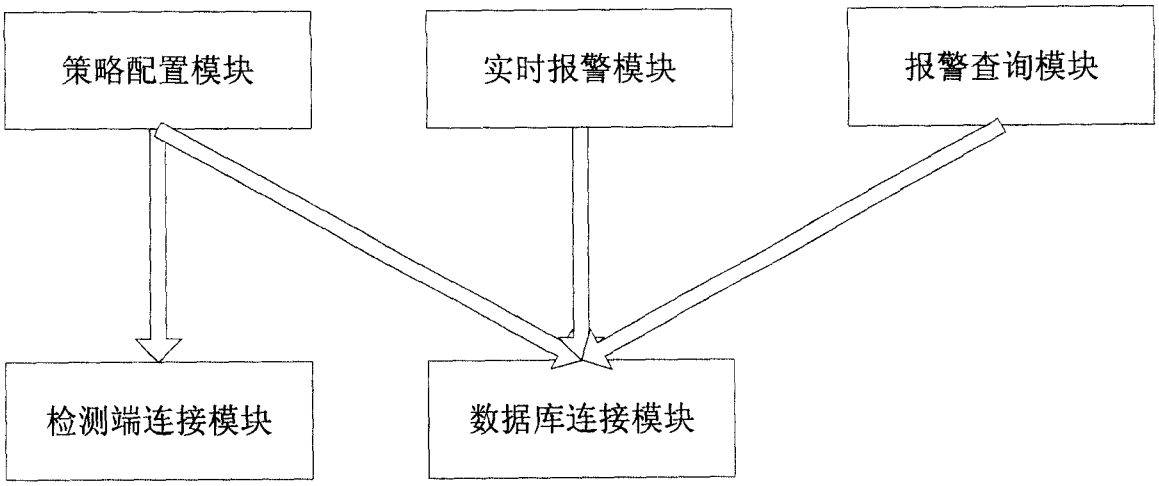


图5