软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2019,30(9):2733−2759 [doi: 10.13328/j.cnki.jos.005518]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

差分信息熵的网络时序型隐蔽信道检测*

张宇飞[1,2], 沈 瑶[1,3], 杨 威[1,3], 肖奘汉[1,2], 黄刘生[1,3]

[1](中国科学技术大学 苏州研究院,江苏 苏州 215123)

[2](中国科学技术大学 软件学院,江苏 苏州 215123)

[3](中国科学技术大学 计算机科学与技术学院,安徽 合肥 230026)

通讯作者: 张宇飞, E-mail: SA614137@mail.ustc.edu.cn

摘 要: 网络隐蔽信道是以合法网络通信信道作为载体建立的一种隐蔽通信技术.相比信息加密,网络隐蔽信道不仅隐藏了传输信息的内容,同时还隐藏了传输信息的行为,因而具有更强的隐蔽性.隐蔽信道技术的出现,使得网络通信中的信息安全和隐私保护受到了极大的威胁,尤其是间谍和其他不法分子可以利用隐蔽信道绕过系统的安全检查机制,窃取机密信息.因此,研究高效且准确率高的隐蔽信道检测技术势在必行.在分析和总结前人研究成果的基础上,提出了差分信息熵的概念,进而提出了基于差分熵的网络时序型隐蔽信道检测算法.首先给出了差分信息熵的定义和相关特性,然后给出了基于差分信息熵的隐蔽信道检测算法的实现原理,以及算法在具体实现过程中的参数设定,最后设计实验检测算法的性能和效果.实验结果表明,基于差分信息熵的检测算法可以有效检测IPCTC,TRCTC,JitterBug 时序型隐蔽信道.

关键词: 时序型隐蔽信道;差分;信息熵;隐蔽信道检测

中图法分类号: TP393

中文引用格式: 张宇飞,沈瑶,杨威,肖奘汉,黄刘生.差分信息熵的网络时序型隐蔽信道检测.软件学报,2019,30(9):2733−2759. http://www.jos.org.cn/1000-9825/5518.htm

英文引用格式: Zhang YF, Shen Y, Yang W, Xiao YH, Huang LS. Detecting covert timing channels based on difference entropy. Ruan Jian Xue Bao/Journal of Software, 2019,30(9):2733−2759 (in Chinese). http://www.jos.org.cn/1000-9825/5518.htm

Detecting Covert Timing Channels Based on Difference Entropy

ZHANG Yu-Fei[1,2], SHEN Yao[1,3], YANG Wei[1,3], XIAO Yan-Han[1,2], HUANG Liu-Sheng[1,3]

[1](Suzhou Research Institute, University of Science and Technology of China, Suzhou 215123, China)

[2](School of Software Engineering, University of Science and Technology of China, Suzhou 215123, China)

[3](School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China)

Abstract: Covert channel is a way to building confidential channels based on the legitimate channels (also named with 'overt channel'). Compared with the encryption technology, covert channel has stronger covertness because it conceals the behavior of covert communication as well as the transmitted message it contains. The emergence of covert channels has threatened the information security and personal privacy in public Internet. Some hackers and criminals, in particular, adopt covert channels to steal secret information bypassing the inspection of security facilities. It is, therefore, crucial to design and deploy more efficient and accurate detection algorithm for covert channels. In this study, a detection algorithm is proposed for covert timing channels based on the difference entropy. First, the definition of difference entropy is introduced, then, the principle of the algorithm is proposed, and the description of the implementation of this algorithm and parameter optimization is given. Lastly, the performance of the detection algorithm is evaluated through experiments,

* 基金项目: 国家自然科学基金(61572456); 江苏省自然科学基金(BK20151241)

Foundation item: National Natural Science Foundation of China (61572456); Natural Science Foundation of Jiangsu Province of China (BK20151241)

收稿时间: 2016-10-18; 修改时间: 2017-03-24, 2017-09-05; 采用时间: 2017-11-23

and experimental results show that proposed algorithm is effective on the detection of the IPCTC, TRCTC, JitterBug covert timing channels.

Key words: covert timing channel; difference; information entropy; convert channel detecting

网络隐蔽信道技术是指利用网络中的合法通信信道构建秘密信道,进行隐秘信息的传输.隐蔽信道在加密传输信息内容的同时,对传输信息行为也进行了加密.传统的加密技术很容易引起注意,并被加以破解.随着密码学等相关理论的发展,密文的破译技术得到了飞速发展,加密技术的可靠性将面临挑战.隐蔽信道技术相比加密技术具有更强的隐蔽性.自 1973 年 Lampson 提出隐蔽信道的概念至今,对于隐蔽信道技术的研究成果层出不穷;与此同时,隐蔽信道检测技术也随着隐蔽信道技术的发展而被研究.对于隐蔽信道领域的正向研究,主要目标是提出具有更高信道容量、更高传输效率、更强隐蔽性的隐蔽信道实现方案,以保障重要信息传输过程的安全;反向研究则是探索更通用、更可靠、检测率更高的隐蔽信道检测算法,及时发现并阻止不法分子利用隐蔽信道窃取机密信息.

根据存储载体的不同,现有的网络隐蔽信道实现方式主要分为两类:网络存储型隐蔽信道利用网络数据包中的某些字段编码隐秘信息进行传输;网络时序型隐蔽信道则将隐秘信息编码到相邻数据包的时间间隔中,利用网络中数据包的传输速率、发送时间和到达时间进行隐秘信息的传输.

时序型隐蔽信道利用数据包的时间间隔传输信息,而数据包的时间间隔容易受到网络环境影响.当网络环境不稳定时,数据包的时间间隔也会发生较大的抖动,造成隐蔽信道传输信息的错误.因此相比存储型隐蔽信道,时序型隐蔽信道的可靠性较差.但是时序型隐蔽信道的优点在于其隐蔽性很强,实现机理和信息编码过程的复杂多变,使得目前尚无方法有效检测所有类型的网络时序型隐蔽信道;并且随着网络和通信技术的发展,网络环境也在逐步改善,这也使得时序型隐蔽信道的准确率和可靠性在逐步提升.因此,时序型隐蔽信道检测技术的研究成为了目前隐蔽信道检测技术研究工作的重点.

针对上述问题,本文提出一种基于差分运算和信息熵的网络时序型隐蔽信道检测算法,其主要贡献如下:

(1) 提出了差分信息熵(difference entropy)的概念,通过理论研究得到差分熵的特性,为检测算法的有效性提供理论基础;

(2) 提出了基于差分信息熵的时序型隐蔽信道检测算法,并通过一系列实验确定算法的最优参数设定;

(3) 设计了对照实验,比较了本文提出的差分熵算法和相似度算法、随机性检测算法、CCE 算法以及熵评估算法对于 IPCTC,TRCTC,JitterBug 这 3 种时序型隐蔽信道的检测效果,着重研究了对于 JitterBug 隐蔽信道的检测效果.

本文第 1 节介绍研究背景,简要描述本文中涉及到的一些预备知识.第 2 节提出差分信息熵的概念并描述其特性.第 3 节介绍基于差分信息熵的网络时序型隐蔽信道检测算法的原理.第 4 节介绍算法的实现过程,包括数据的收集、处理方式以及算法在程序实现过程中涉及到的参数设定.第 5 节通过设计实验来检验算法的性能和效果.最后,第 6 节说明目前研究过程中存在的问题以及之后的研究方向,并且总结全文.

1 预备知识

1.1 隐蔽信道

隐蔽信道的概念最初由 Lampson 在 1973 年提出,他将隐蔽信道定义为本意不是用来传输信息的通信信道[1].即:如果采用某种手段或方法,利用一个本不是用于通信的系统或过程来传输信息,那么这种手段或方法就构建了一条隐蔽信道.文献[1]中,作者将两个具有调用关系的程序称为 Customer 和 Service,并给出了在 Customer 调用 Service 时,程序 Service 可能将 Customer 的信息泄露给第三方的 6 种可能的情况,进而表明:即使 Customer 在调用 Service 时为 Service 规定了严格的数据访问权限,Customer 的数据依然有可能被泄漏.通过对 6 种情况的阐述,作者认为,Service 程序将产生 3 种类型的通信信道.

- 存储信道:Service 可以将数据存储在可以被第三方访问到的存储空间中来实现数据的隐秘传输;
- 加密信道:Service 利用某种方式,在合法信道中嵌入机密信息进行传输;

- 隐蔽信道:通过某种方式,利用本不是用来传输信息的过程或机制来传输信息.

Lampson 隐蔽信道的定义后来被 Tsai 等人进一步完善.Tsai 给出的隐蔽信道的定义是:给定一个强制安全策略模型 M 以及其在操作系统中的解释 $I(M)$,$I(M)$中的两个主体 $I(S_h)$和 $I(S_l)$之间的通信是隐蔽的,当且仅当模型 M 中的对应主体 S_h 和 S_l 之间的任何通信都是非法的[2].该定义认为:隐蔽信道只与系统的强制访问控制策略模型相关,并且广泛存在于部署了强制访问控制机制的安全操作系统、安全网络和安全数据库中[3].

从 1973 年至今,对于隐蔽信道方面的研究成果层出不穷.Wang 等人、Zander 等人对于目前已有隐蔽信道方面的研究成果进行了充分的总结和归纳[3,4].Wang 在其综述中描述了 4 种类型的隐蔽信道[3].

(1) 数据库信道:利用数据库系统对外传输数据,主要利用数据库中的存储资源、管理资源以及事务并发控制机制构建隐蔽信道;

(2) 阈下信道:基于公钥密码技术的数字签名、认证等应用密码体制的输出密码数据中建立起来的一种隐蔽信道;

(3) 网络信道:网络信道可以分为两种:第 1 种存在于多级安全系统中,是一种用于从高安全级向低安全级传输信息的隐蔽信道;第 2 种则不涉及多级安全的概念,在普通的通信信道中嵌入一层隐蔽的通信信道,这里普通信道是隐蔽信道的载体;

(4) 推理信道:严格意义上讲,推理信道并不是一种通信信道,它利用某种方式,通过对公开数据的查看来推理出隐私数据的内容.这种技术通常用于获取数据库中的私有信息,例如:利用数据库中的某些聚集查询和函数查询来推断某一条数据记录的具体内容.

从文献[3]可以看出,对于隐蔽信道的研究已经扩展了 Lampson 和 Tsai 最初对于隐蔽信道的定义. Lampson 提出的 3 类信道,在现今的研究中都可以归于隐蔽信道的范畴;Wang 所描述的 4 类隐蔽信道中,数据库信道对应于 Lampson 描述的存储信道,阈下信道和网络信道则对应于 Lampson 提出的加密信道,推理信道对应于 Lampson 的隐蔽信道.目前研究的隐蔽信道主要针对利用网络协议规定的数据帧或数据报来构建的隐蔽信道,对应于 Wang 在其综述中提到的第 2 类网络信道.

1.2 网络隐蔽信道

网络隐蔽信道是指利用计算机网络中的各种协议,以合法通信信道作为载体,构建用于传输隐秘信息的通信信道.这里关于术语隐蔽信道(covert channel)的意义和指代的范畴,Zander 在其综述中做出了明确的界定[4]:利用网络中的各种协议帧的格式来嵌入隐秘信息的技术称为隐蔽信道;通过加密技术将隐秘信息嵌入到数据明文中的方式则称为隐写术(steganography);术语信息隐藏(information hiding)涵盖了以上两者.作为载体的合法信道则称为公开信道(overt channel).

对于网络隐蔽信道技术,我们可以从多个角度进行分类.

(1) 按照隐秘信息编码原理划分,网络隐蔽信道可以划分为网络存储型隐蔽信道和网络时序型隐蔽信道:前者主要利用网络协议中的一些控制字段编码隐秘信息,实现隐秘传输;后者则利用了数据包的发送时间、接收时间、时间间隔来进行隐秘信息的编码;

(2) 按照传输载体划分,网络隐蔽信道可以分为网络协议隐蔽信道和网络应用隐蔽信道:前者利用的是涉及到网络基础建设的底层协议,诸如 TCP,UDP,IP,ICMP 以及其他的链路层协议;后者利用的则是涉及到高级网络应用的应用层协议,诸如 HTTP,HTTPS,SMTP 以及 SSH 等;

(3) 按照与合法信道的关系划分,网络隐蔽信道可以分为主动式隐蔽信道和被动式隐蔽信道.主动式隐蔽信道的发送方和接受方与合法信道的发送方和接收方是相同的,因此,主动式隐蔽信道可以直接操纵信道的容量和传输速率;被动式隐蔽信道的发送方、接受方与合法信道的发送方、接收方不同,而是介于合法信道发送方和接收方之间.被动式隐蔽信道仅仅是借助于合法信道作为载体,对于信道的控制力较弱,隐蔽信道的传输能力和性能也主要取决于作为载体的合法信道的情况.被动式隐蔽信道和主动式隐蔽信道的区别可以用图 1、图 2 来描述.

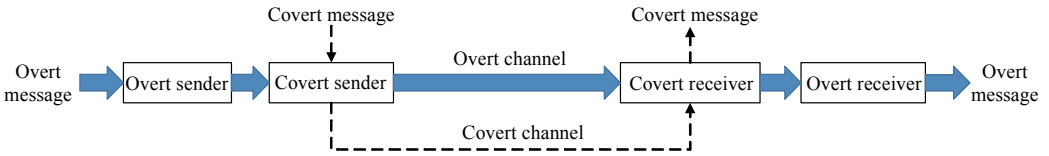

Fig.1 Passive covert channel

图 1 被动式隐蔽信道

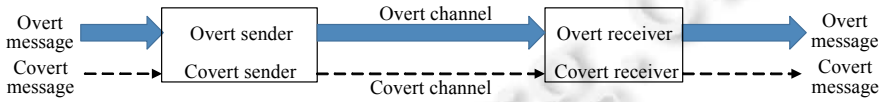

Fig.2 Active covert channel

图 2 主动式隐蔽信道

以上的分类方式中,第 1 种分类方式最能体现隐蔽信道的本质,下文中的讨论都基于该分类方式.

1.2.1 网络存储型隐蔽信道

网络存储隐蔽信道主要利用网络协议帧中的字段编码隐秘信息,构建隐蔽通信信道.常用的载体有 IP 数据报中的服务类型(TOS)[5]、标志位(flag)[6]、数据报编号(identification)[7,8]、生存时间(TTL)[9,10]、数据报长度等字段以及 TCP 报文中的紧急指针(URG)[11]、报文序号(sequence)[12,13]等字段.另外,以上协议中的选项部分和填充字节也可以作为隐秘信息的载体.

网络存储型隐蔽信道的构建不仅可以利用底层的网络协议,还可以利用一些应用层的协议,如 HTTP[14−16]和 DNS[17]协议.一些 TCP/IP 层之下的协议,诸如以太网、令牌环网络中涉及的相关协议也可以作为网络存储型隐蔽信道的载体[18−20].

本文提出的检测算法主要针对网络时序型隐蔽信道,因此对于存储型隐蔽信道,这里仅简要说明.

1.2.2 网络时序型隐蔽信道

网络时序型隐蔽信道将隐秘信息编码为数据包传输过程中的时间间隔,达到隐秘传输的目的.对于主动式的网络时序型隐蔽信道,发送端在发送数据包时就将隐秘信息编码到了数据包的时间间隔中,发送端发送数据包时,发送的时间间隔就根据要传输的隐秘信息进行了调制;对于被动式时序型隐蔽信道,则主要通过拦截、抓取正常网络信道中正在传输的数据包,根据要传送的隐秘信息,将数据包的时间间隔延长或缩短,达到传输隐秘信息的目的.

以下是部分网络时序型隐蔽信道类型的简要介绍.

- IPCTC (IP covert timing channel)

2004 年,Cabuk 提出了一种时序型隐蔽信道称为 IPCTC[21],在其实现中,发送方和接受方需要事先约定一个时间间隔 t.发送开始后,对于每一个时间间隔 t,如果发送方需要发送比特 1,就在该时间间隔内发送一个数据包;否则,发送方保持静默,接收方观察数据包到达的时间间隔,就可以得到信道中包含的隐秘信息.如果在时间间隔 t 之内收到了发送方发送的数据包,传输的数据被识别为比特 1;否则,识别为比特 0.IPCTC 的传输原理如图 3 所示.

Cabuk 还在文中给出了发送方和接收方的同步机制,并指出了 4 种可能影响 IPCTC 隐蔽信道传输性能的因素.其中,时间间隔 t 直接决定了信道的容量和传输效率:t 值越小,信道的传输速率越高;但是时间间隔的减小,会导致传输过程中的错误率提高.理论上,该信道的极限传输速率等同于发送端处理数据的速率.

IPCTC 利用在约定的时间区间中是否发送数据包来表示传输比特 0 或比特 1.该信道并不是直接将信息编码到数据包时间间隔的数值中,而是利用在时间区间内的发送或不发送数据包的事件,在发送方和接收方之间建立了一个可以共享的布尔值,接收方通过观察这个布尔值的结果来接收隐秘信息.在 Zander 的综述中, IPCTC 又被称为 ON/OFF 隐蔽信道[4].

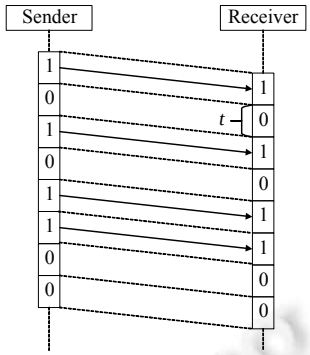

Fig.3 IP covert timing channel (IPCTC)

图 3 IPCTC 网络时序型隐蔽信道

- TRCTC (time replay covert timing channel)

2006 年,Cabuk 对其提出的 IPCTC 隐蔽信道进行了改进[22],提出了一种新的时序型隐蔽信道——TRCTC. 随意确定的时间间隔 t 很有可能会改变包间隔数据的统计特征,从而使得 IPCTC 的隐蔽性和抗检测性降低.TRCTC 首先对正常网络信道中的数据包时间间隔进行采样,然后将采样得到的数据划分为两个集合 S_0 和 S_1,传输数据时,如果需要传输比特 0,就从集合 S_0 中选择一个时间间隔,发送数据包;如果需要传输比特 1,就从 S_1 中选择一个时间间隔,发送数据包.由于通信中用到的时间间隔采样自合法通信中的时间间隔,TRCTC 和正常通信产生的包间隔具有相似的数值特性.相比 IPCTC,TRCTC 的隐蔽性更好.

- JitterBug

2006 年,Shah 等人提出一种利用用户敲击键盘的时间间隔来编码信息的时序型隐蔽信道——JitterBug[23]. JitterBug 设定了一个时间参数 ω,如果需要传输比特 1,就增加用户两次敲击键盘的时间间隔,使该间隔可以被参数 ω 整除;如果要传输比特 0,就增加时间间隔使其可以被 $\omega/2$ 整除,但不能被 ω 整除.显然,参数 ω 的选择直接决定了信道的容量和传输效率:ω 的值越小,信道的传输速率越高;并且,ω 的值越小,产生的时间间隔数据和合法信道的数据的相似度就会越高,信道的隐蔽性就越强.Jitterbug 信道的缺点在于它是一个被动式隐蔽信道,不论 ω 的取值为多少,信道的极限容量和传输效率本质上取决于载体信道的容量和传输效率.

以上 3 种方式为时序型隐蔽信道的主要实现方式,除此之外,时序型隐蔽信道的种类还有 MBCTC[24],DM[25] 等.下文中的研究主要涉及到了上面提到的 3 种类型的隐蔽信道,因此对于其他的类型,这里不再赘述.

相比存储型隐蔽信道,时序型隐蔽信道隐蔽性更强,检测时序型隐蔽信道的难度也更大.时序型隐蔽信道是一种即时通信,发送方发送数据后,接收方必须及时接收,否则,数据包传输完毕后,传输的信息也随之丢失.另外,时序型隐蔽信道利用数据包时间间隔进行隐秘信息的编码,所以对于网络环境有一定的要求,当前网络环境的稳定性直接影响到时序型隐蔽信道传输过程的可靠性和传输信息的正确性.

1.2.3 其他类型的网络隐蔽信道

近几年来,对于网络隐蔽信道技术的研究提出了更多的网络隐蔽信道实现方案.存储型和时序型的分类已经无法完全涵盖所有的隐蔽信道类型,例如:利用虚拟机之间共享的硬件资源在两个虚拟机实例之间构建隐蔽信道[26];利用数据包的排序方式来构建隐蔽信道[27];利用多条通信流的构建隐蔽信道[28]等.这些隐蔽信道无法从严格意义上讲其属于存储型或时序型的类型,也不在本文的研究范围.

1.3 网络时序型隐蔽信道的检测技术

时序型隐蔽信道将隐秘信息编码到网络数据包传输的时间间隔中.对于合法信道,数据包的时间间隔仅取决于当前的网络环境;而对于时序型隐蔽信道,数据包的时间间隔则主要取决于要传输的隐秘信息.因此,时序型隐蔽信道的包时间间隔数据会在一些统计特征上呈现出和合法信道的差别.所以目前比较成熟的时序型隐

蔽信道检测技术主要通过评估待测数据的统计特征,包括均值、方差等统计量,来达到检测和识别时序型隐蔽信道的目的.

Cabuk 认为:由于正常数据包的传输时间间隔仅仅由网络环境决定,因此时间间隔数据的随机性较高;而时序隐蔽信道中的时间间隔用于传输隐秘信息而被刻意操纵,因此随机性比较低[21].基于这样的观点,Cabuk 提出了两种时序型隐蔽信道检测方式.

- 第 1 种方式称为随机性检测算法,该算法检测数据包时间间隔的标准差.通过将待检测数据划分 N 个窗口,分别计算每个窗口的标准差,结果分别记为:$\sigma_1,\sigma_2,...,\sigma_N$.然后计算不同窗口标准差的差距的标准差,如下:

$$regularity = STDEV\left(\frac{|\sigma_i-\sigma_j|}{\sigma_i}, i<j<N, \forall i,j\right),$$

  其中,*STDEV* 表示计算标准差.Cabuk 认为:合法信道数据的随机性高,因此数据的标准差不是恒定的,因此 *regularity* 的值也因标准差数据的波动较大;而隐蔽信道数据的随机性低,在传输数据的过程中,包时间间隔的标准差是恒定的,所以 *regularity* 的值会很小;

- Cabuk 提出的第 2 种检测方式称为 ε相似度(ε-similarity),基本思想是:合法通信中的各个数据包时间间隔是相互独立的;而隐蔽信道的数据包间隔编码了隐秘信息,因此数据包时间间隔之间是否独立取决于要传输的数据.ε相似度算法计算排序后的相邻包间隔数据之间的相似度.对于排序后的包间隔数据序列 $P_1,P_2,...,P_N$,相邻数据包间相似度的计算方式如下:

$$similarity(i) = \frac{|P_i - P_{i+1}|}{P_i}.$$

计算所有数据对的相似度后,统计相似度小于ε值的数据的比例.比例越大,待测信道中存在隐蔽信道的可能性就越高.

检测时序型隐蔽信道的另一种较为成熟的技术是评估待测数据的熵率.熵(entropy)是随机变量随机程度的定量度量,它的定义来自于 Shannon 提出的信息论[29],而熵率(entropy rate)则是随机变量序列的平均信息熵,或者等价定义为无限个随机变量序列的极限条件熵.熵(记作 *EN*)和熵率(记作 *ER*)的计算方式如下:

$$EN(X) = -\sum_{i=1}^{n} p_i \log_2 p_i,$$

$$ER(x) = \frac{1}{n}\lim_{n\to\infty} it\, EN(X_1,X_2,...,X_n) = \lim_{n\to\infty} it\, EN(X_n \mid X_1,X_2,...,X_{n-1}).$$

由于熵率是一个极限定义,对于有限的数据无法计算.Gianvecchio 提出一种带修正值的条件信息熵(correct condition entropy),使得可以通过有限的数据来估计熵率[30].计算方式如下:

$$CCE(\boldsymbol{X}_n)=CE(X_1,X_2,...,X_n)+perc(\boldsymbol{X}_n)\cdot EN(\boldsymbol{X}_1),$$

其中,$\boldsymbol{X}_n$ 表示长度为 n 的随机变量的序列,$CE(X_1,X_2,...,X_n)$表示 $X_1,X_2,...,X_n$ 的条件信息熵,$perc(\boldsymbol{X}_n)$则表示测试数据中所有唯一的 n 长度序列占所有 n 长度序列的比例,$EN(\boldsymbol{X}_1)$就是随机变量 X 的熵值.而熵率的估计值就是取不同的 n 值得到的 $CCE(\boldsymbol{X}_n)$的最小值.

Gianvecchio 的观点和 Cabuk 是相同的,同样认为隐蔽信道的数据的随机性要比正常数据低,熵率值越小,信道中含有隐蔽信道的可能性就越高.

检测时序型隐蔽信道还有其他的算法,诸如基于 SVM 的隐蔽信道分类器[31−33]、基于神经网络[34−36]的检测算法、基于熵率和标准差的检测方法[37]等.本文提出的检测算法主要基于ε相似度和熵,因此对于其他类型的检测算法,这里不再详述.

2 差分信息熵

本节主要对差分信息熵理论进行阐述.通过下文中的理论分析可以看到,差分信息熵可以同时对待检测数据的分布特性和数值特性进行评估.下文中为了便于分析,首先给出了一些定义和定理,然后对差分信息熵理论

进行说明.

2.1 分布特性与数值特性

信息熵是一种对随机变量随机性程度的定量度量.熵的概念源于热力学,由 Shannon 在 1948 年提出信息论后引入计算机和通信领域[29].离散型随机变量信息熵的计算方式前文中已经给出.Gianvecchio 通过评估数据的熵率来检测时序型隐蔽信道.文献[30]作者用 4 种隐蔽信道组织实验验证其提出的 CCE 算法的有效性[30]:IPCTC,TRCTC,MBCTC,Jitterbug.根据作者提供的实验数据,CCE 算法对于 JitterBug 隐蔽信道的检测效果很差,这主要是因为 Jitterbug 隐蔽信道对于数据包间隔做出的修改很小,使得产生的包间隔数据和正常数据在统计特性上的差别很小.

我们认为,利用熵率来对数据进行评估仅仅考虑到了数据的统计特性,而没有考虑数据的数值特性.例如,对于两个离散型随机变量 X,Y,假设 X 可能的取值为 1,3,5,7,取到每个值的概率为 1/4;Y 可能取值是 1,5,6,11,取到每一个值的概率同样为 1/4.那么代入信息熵计算公式,X 和 Y 得到的熵值是相同的:

$$EN(X)=EN(Y)=(-4)\times\frac{1}{4}\times\log_2\frac{1}{4}=2.$$

但是很显然,从数值上来看,变量 X 的规律性更强.对于相互独立的随机变量序列,熵率的值跟熵值是相同的;对于非独立的随机变量序列,熵率的大小低于熵值,此时,熵率和熵值的差距取决于序列中各个元素关联度的大小.Gianvecchio 提出的 CCE 算法之所以对于 JitterBug 隐蔽信道的检测效果不好,我们认为,主要是因为熵率仅仅评估了待测数据的分布特性,而没有评估待测数据数值特性.根据构建原理,JitterBug 中的时间序列的要么是参数 ω 的倍数,要么是 $\omega/2$ 的倍数.因此,Jitterbug 中的任意两个数据的差值一定是一个 $\omega/2$ 的倍数.就数值特性而言,JitterBug 隐蔽信道中的数据表现出了和正常数据不同的特性.

为了同时对数据的分布特性和数值特性进行评估,我们结合 Cabuk 提出的 ε 相似度算法和 Gianvecchio 提出的 CCE 算法,提出差分信息熵的概念,进而提出差分信息熵的网络时序型隐蔽信道检测算法.

2.2 离散型随机变量的拆分与合并

为了便于下文中对于差分信息熵的详细说明,首先需要说明下面的定理:

定理 1. 对于任意的正数序列 $p_1,p_2,\ldots,p_n$,可以得到:

$$\left(\sum_{i=1}^{n}p_i\right)\times\log_2\left(\sum_{i=1}^{n}p_i\right)>\sum_{i=1}^{n}(p_i\log_2 p_i).$$

证明:只要将不等式两边的式子求差即可:

$$\left(\sum_{i=1}^{n}p_i\right)\cdot\log_2\left(\sum_{i=1}^{n}p_i\right)-\sum_{i=1}^{n}p_i\log_2 p_i=\sum_{i=1}^{n}\left[p_i\cdot\log_2\left(\frac{\sum_{i=1}^{n}p_i}{p_i}\right)\right].$$

由于序列中的每个 p_i 都是正数,因此,$\left(\sum_{i=1}^{n}p_i\right)\Big/p_i$ 一定是大于 1 的数,所以结果和式中的每一项都是正数,也就说明上面的做差结果是大于 0 的,原式成立. □

对于一个可能的取值有 $v_1,v_2,\ldots,v_n$、对应的概率分别是 $p_1,p_2,\ldots,p_n$ 的离散型随机变量 X,我们可以为其定义两种操作:

定义 1. 通过随机变量 X 定义随机变量 Y,Y 的取值规则如下:

$$Y=\begin{cases}v_i, & X=v_i\wedge 1\leqslant i\leqslant n-2\\ v_n, & X=v_n\vee X=v_{n-1}\end{cases}.$$

此时,我们可以说随机变量 Y 由随机变量 X 通过合并(merge)操作得到.

定义 2. 定义二值的随机变量 a,a 以概率 q 取值为 1,以概率 $(1-q)$ 取值为 0,然后通过随机变量 X 和 a 定义一个随机变量 Z,Z 的取值规则如下:

$$Z=\begin{cases}v_i, & X=v_i, 1\leqslant i\leqslant n-1\\ v_n, & X=v_n\wedge a=0\\ v_{n+1}, & X=v_n\wedge a=1\end{cases},$$

其中,$\forall 1\leqslant i\leqslant n, v_{n+1}\neq v_i$.此时,我们可以说随机变量 Z 由随机变量 X 通过拆分(split)操作得到.

现在考虑通过合并和拆分 X 得到的随机变量 Y 和 Z 的熵值.随机变量 X 具有 n 个可能取值,而随机变量 Y 在 X 的值为 v_{n-1} 或 v_n 时,取值都是 v_n,因此 Y 只有 $n-1$ 种可能的取值;而通过拆分操作得到随机变量 Z 在 X 的值为 v_n 时根据随机变量 a 的取值情况有两种取值:v_n 和 v_{n+1},因此,Z 共有 $n+1$ 种可能的取值.通过代入熵计算公式,Y 和 Z 的熵值分别为

$$EN(Y)=-\sum_{i=1}^{n-2}p_i\log_2 p_i-(p_{n-1}+p_n)\log_2(p_{n-1}+p_n),$$

$$EN(Z)=-\sum_{i=1}^{n-1}p_i\log_2 p_i-(p_n\cdot q)\log_2(p_n\cdot q)-[p_n\cdot(1-q)]\log_2[p_n\cdot(1-q)].$$

然后,将其和 X的熵值做差:

$$EN(Y)-EN(X)=p_{n-1}\log_2 p_{n-1}+p_n\log_2 p_n-(p_{n-1}+p_n)\log_2(p_{n-1}+p_n),$$

$$\begin{aligned}EN(Z)-EN(X)&=p_n\log_2 p_n-(p_n\cdot q)\log_2(p_n\cdot q)-[p_n\cdot(1-q)]\log_2[p_n\cdot(1-q)]\\&=[p_nq+p_n\cdot(1-q)]\log_2[p_nq+p_n\cdot(1-q)]-(p_n\cdot q)\log_2(p_n\cdot q)-[p_n\cdot(1-q)]\log_2[p_n\cdot(1-q)].\end{aligned}$$

根据定理 1,我们可以得到 $EN(Y)-EN(X)$的结果小于 0,$EN(Z)-EN(X)$的结果大于 0,因此可以得到如下定理:

定理 2. 拆分一个随机变量会使熵增大,合并一个随机变量会使熵减小.

2.3 差分值矩阵与差分概率矩阵

本节定义两个辅助矩阵来帮助下文中对于差分信息熵的讨论.

定义 3. 设离散型随机变量 X 可能取值有 $v_1,v_2,...,v_n$,对应的概率分别是 $p_1,p_2,...,p_n$,这里为了便于分析,将数值 $v_1,v_2,...,v_n$ 递增排序.然后定义一个随机变量 $Y=X_1-X_2$,其中,X_1 和 X_2 相互独立并且和 X 具有完全相同的概率分布.此时可以定义一个 $n\times n$ 的矩阵 D_v,矩阵中的元素 v_{ij} 表示 X_1 取值 v_i 并且 X_2 取值 v_j 时的 Y 值;同时还可以定义另一个 $n\times n$ 的矩阵 D_p,矩阵中的第 i 行第 j 列元素 p_{ij} 表示表示 X_1 取 v_i、X_2 取 v_j 时的联合概率.此时称矩阵 D_v 为变量 X 的差分值矩阵(difference value matrix),矩阵 D_p 为变量 X 的差分概率矩阵(difference probability matrix).

由于 X_1,X_2 独立,所以 p_{ij} 就是 p_i 和 p_j 的乘积.X 的差分值矩阵和差分概率矩阵如下:

$$D_v=\begin{pmatrix}v_{1,1}&v_{1,2}&\cdots&v_{1,n}\\ v_{2,1}&v_{2,2}&\cdots&v_{2,n}\\ \cdots&\cdots&\cdots&\cdots\\ v_{n,1}&v_{n,2}&\cdots&v_{n,n}\end{pmatrix},D_p=\begin{pmatrix}p_1^2&p_1p_2&\cdots&p_1p_n\\ p_2p_1&p_2^2&\cdots&p_1p_n\\ \cdots&\cdots&\cdots&\cdots\\ p_np_1&p_np_2&\cdots&p_n^2\end{pmatrix}.$$

通过观察可以发现,差分值矩阵 D_v 具有如下的性质.

(1) 主对角线上的值均为 X_1,X_2 取相同值的情况,因此主对角线上的元素 $v_{1,1},v_{2,2},...,v_{n,n}$ 的值均为 0;

(2) 关于主对角线互相对称的两个元素的结果互为相反数.显然,对于任意元素 $v_{i,j}=v_i-v_j$,它关于主对角线的对称位置为 $v_{j,i}=v_j-v_i$,和 $v_{i,j}$ 互为相反数;

(3) 每一个元素一定都比它上方和右方的元素大.这个结果是显而易见的.任意一个元素 $v_{i,j}=v_i-v_j$,它右边的元素是 $v_{i,j+1}=v_i-v_{j+1}$,上边的元素是 $v_{i-1,j}=v_{i-1}-v_j$,由于 $v_{j+1}>v_j, v_i>v_{i-1}$,所以 $v_{i,j}>v_{i-1,j}$;同时,$v_{i,j}>v_{i,j+1}$;

(4) 矩阵中数值最大的元素在左下角,最小的则在右上角;并且任意一行、任意一列中一定不存在相同的元素;

(5) 整个矩阵中最多包含 n^2-n+1 个不同的值,最少包含 $2n-1$ 个不同的值.对于前者,由于整个矩阵包含 n^2 个元素,而主对角线上的元素都是 0,共有 n 个 0,如果其余的元素任意两个都不相同,那么此时,矩阵包含的不同元素最多,这个最大值就是 n^2-n+1;对于后者,则基于这样的事实:不论 X 具有怎样的数值特

征,矩阵 D_v 中,总可以找到一条从左下角的最大元素到达右上角的最小元素的路径,使得沿着该条路径移动时,方向总是向上或向右,此时这条路径一定包含 $2n-1$ 个元素.图 4 给出了对于 X 有 4 个不同取值的情况,沿着路径遍历这些元素时,得到的一定是严格递减的序列.也就是说,不论 X 具有怎样的数值特征,在矩阵中都可以找到 $2n-1$ 个不同的元素.

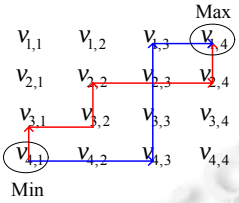

Fig.4 There are at least $2n-1$ different elements in the matrix D_v (two paths are shown in this figure)

图 4 D_v 矩阵中至少包含 $2n-1$ 个不同元素(图中给出了其中的 2 条路径)

2.4 差分信息熵及其特征

2.4.1 差分信息熵的定义

对于离散型的随机变量 X,我们用下面的方式定义其差分信息熵:

定义 4. 对于一个离散型随机变量 X,定义一个随机变量 $Y=X_1-X_2$,其中,X_1 和 X_2 相互独立并且和 X 具有完全相同的概率分布.此时,随机变量 X 的差分信息熵为随机变量 Y 的信息熵,记作 $DEN(X)$,即:

$$DEN(X)=EN(Y)=EN(X_1-X_2).$$

下面说明差分信息熵 $DEN(X)$的特征.根据上文中差分矩阵的性质(5),对于有 n 个可能取值的随机变量 X 做差分运算后,最多有 n^2-n+1 种不同的结果;最少有 $2n-1$ 种不同结果.而如果不考虑差值,仅仅考虑 X_1,X_2 不同的取值组合,则有 n^2 种情况.我们可以将这些不同的取值组合按照差分值 Y 分组,得到的分组数目记为 C,并将这 C 个分组分别为 $g_1,g_2,...,g_C$.图 5 是一个分组的样例.

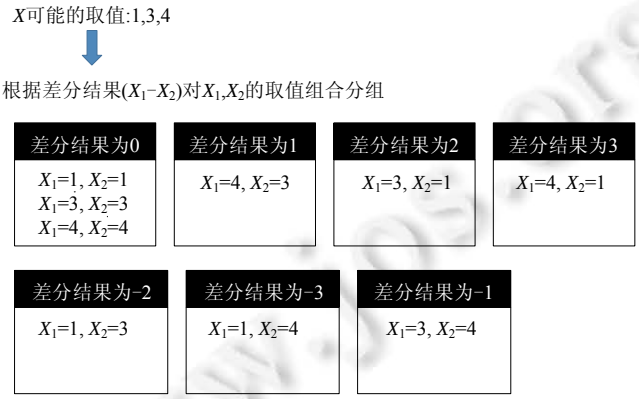

Fig.5 Example of classifying the combination of X_1 and X_2 according to the difference result (X_1-X_2)

图 5 根据差分结果对随机变量的取值组合分组的样例

显然,这里的分组数目 C 满足下面的关系式:

$$2n-1\leqslant C\leqslant n^2-n+1.$$

分组完成后,可以得到计算差分熵 $DEN(X)$的公式如下:

$$DEN(X)=-\sum_{k=1}^{C}SUM\{g_k\}\log_2 SUM\{g_k\},$$

其中,

$$SUM\{g_k\}=\sum_{i=1}^{n}\sum_{j=1}^{n}count(g_k,i,j),$$

$$count(g_k,i,j)=\begin{cases}p_{i,j}, & (X_1=v_i,X_2=v_j)\in g_k\\ 0, & (X_1=v_i,X_2=v_j)\notin g_k\end{cases}.$$

2.4.2 数值特性对差分熵的影响

本节说明数值特性对于差分熵的影响,并给出差分熵的值域.下文将分 3 种情况说明:首先考虑两种特殊情况,然后再说明更一般的情况.

情况 1. 取值序列 $v_1,v_2,...,v_n$ 是等差数列,假设公差为 d,此时,差分值矩阵变成了如图 6 所示的形式.

$$D_v=\begin{pmatrix}0 & -d & -2d & \cdots & -(n-1)d\\ d & 0 & -d & -2d & \cdots\\ 2d & d & 0 & -d & -2d\\ \cdots & 2d & d & 0 & -d\\ (n-1)d & \cdots & 2d & d & 0\end{pmatrix}$$

Fig.6 Difference value matrix in the case 1

图 6 情况 1 的差分值矩阵

如图 6 所示,此时位于同一条斜线上的元素的值都相等,矩阵中不同的值的个数恰好为最小值 $2n-1$,即 $C=2n-1$,此时对应的差分熵为

$$DEN_{\min}(X)=-2\sum_{k=1}^{n-1}\left[\left(\sum_{i=1}^{n-k}p_ip_{i+k}\right)\log_2\left(\sum_{i=1}^{n-k}p_ip_{i+k}\right)\right]-\left(\sum_{i=1}^{n}p_i^2\right)\log_2\left(\sum_{i=1}^{n}p_i^2\right).$$

关于这个值和熵值 $EN(X)$的大小关系,我们尚未找出完全的证明方式,这里仅仅给出一个不完全的分析.根据差分概率矩阵 D_p,我们可以定义一个离散随机变量 X',X'共有 n 个可能的取值,记作:$r_1,r_2,...,r_n$,而对应的概率,则由下面的规则给出:

$$P\{X'=r_i\}=\sum_{k=1}^{n-i+1}p_kp_{k+i-1}+\sum_{k=n-i+2}^{n}p_kp_{k+i-n-1}.$$

X'的分布律可以用图 7 来描述.

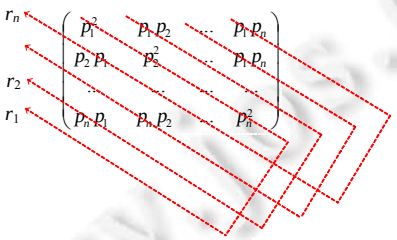

Fig.7 Distribution law of random variable X'

图 7 随机变量 X'的分布律

为了更直观地表达,我们列举各个变量值对应的概率如下:

$$r_1:p_1\cdot p_1+p_2\cdot p_2+...+p_n\cdot p_n$$

$$r_2:p_1\cdot p_2+p_2\cdot p_3+...+p_n\cdot p_1$$

$$r_3:p_1\cdot p_3+p_2\cdot p_4+...+p_n\cdot p_2$$

$$...$$

$$r_n:p_1\cdot p_n+p_2\cdot p_1+...+p_n\cdot p_{n-1}$$

可以看到,X'的每个概率值都是由 X 的概率值通过轮换相乘然后求和得到.这是一个平均化的运算过程,这

意味着 X'的概率分布比 X 跟接近于均匀分布(此处尚未找到完全的证明方式,但多次实验结果表明该结论是正确的).由于均匀分布的随机性最高,熵值最大,所以可以得到下面的结论:

$$EN(X')\geqslant EN(X).$$

下面通过拆分的方式,通过 X'构造 Y,将上文中每个 r_i 概率值拆分出来,如图 8 所示.

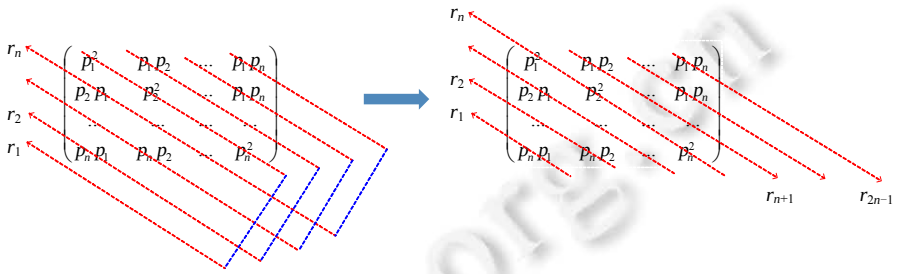

Fig.8 Getting Y from X' by splitting operation

图 8 拆分 X'得到 Y

可以看到,此时 Y 的熵值就是 $DEN_{\min}(X)$.根据定理 2,拆分使得熵值增大,因此:

$$DEN_{\min}(X)=EN(Y)>EN(X')>EN(X).$$

表 1 是通过程序模拟得到的实验数据,对于每一个不同的分布律,表中只列出各个取值的概率,随机变量具体可以取到哪些值并不影响实验结果,因此没有列出.实验数据充分表明,前文中的结论在绝大多数情况下是正确的.

Table 1 Data from the experiment comparing values of entropy and minimum difference entropy

表 1 随机变量熵值和最小差分熵值的比较实验数据

分布律	熵值	最小差分熵值
{0.50,0.50}	1.000	1.500
{0.70,0.30}	0.881	1.401
{0.10,0.90}	0.468	0.860
{0.33,0.33,0.33}	1.584	2.199
{0.50,0.25,0.25}	1.500	2.186
{0.80,0.10,0.10}	0.921	1.603
{0.80,0.15,0.15}	0.884	1.520
{0.25,0.25,0.25,0.25}	2.000	2.655
{0.40,0.10,0.30,0.20}	1.846	2.663
{0.80,0.10,0.05,0.05}	1.021	1.789
{0.20,0.20,0.20,0.20,0.20}	2.322	2.999
{0.30,0.20,0.40,0.05,0.05}	1,946	2.640
{0.80,0.05,0.05,0.05,0.05}	1.121	1.983
{0.30,0.20,0.30,0.10,0.10}	2.171	2.870

情况 2. 取值序列 $v_1,v_2,...,v_n$ 呈不规则分布,使得任意两个不同的项作差值结果都不相同,此时,差分结果 Y 可能的取值个数为最大值 n^2-n+1,因此 $C=n^2-n+1$.此时,对应的差分熵值是:

$$DEN_{\max}(X)=-\sum_{i=1}^{n}\left[\sum_{j=1}^{i-1}(p_ip_j\log_2 p_ip_j)+\sum_{j=i+1}^{n}(p_ip_j\log_2 p_ip_j)\right]-\left(\sum_{i=1}^{n}p_i^2\right)\log_2\left(\sum_{i=1}^{n}p_i^2\right).$$

根据定理 1,可得:

$$-\left(\sum_{i=1}^{n}p_i^2\right)\log_2\left(\sum_{i=1}^{n}p_i^2\right)<-\sum_{i=1}^{n}p_i^2\log_2 p_i^2.$$

于是,

$$DEN_{\max}(X)<-\sum_{i=1}^{n}\left[\sum_{j=1}^{i-1}(p_ip_j\log_2 p_ip_j)+\sum_{j=i+1}^{n}(p_ip_j\log_2 p_ip_j)\right]-\sum_{i=1}^{n}p_i^2\log_2 p_i^2$$
$$=-\sum_{i=1}^{n}\sum_{j=1}^{n}(p_ip_j\log_2 p_ip_j)$$
$$=-\sum_{i=1}^{n}\sum_{j=1}^{n}[p_ip_j(\log_2 p_i+\log_2 p_j)]$$
$$=-\sum_{i=1}^{n}\sum_{j=1}^{n}(p_ip_j\log_2 p_i)-\sum_{i=1}^{n}\sum_{j=1}^{n}p_ip_j\log_2 p_j$$
$$=2EN(X).$$

因此我们可以知道,随机变量 X 的差分熵的最大值小于 2 倍的 X 熵值,即:

$$DEN_{\max}(X)<2EN(X).$$

情况 3. 对于更一般的情况,$2n-1<C<n^2-n+1$.考虑情况 2,Y 的取值个数是 n^2-n+1,此时矩阵 D_p 中,除去主对角线的元素,其余的每一个元素都对应 Y 的一个不同的取值的概率.而我们现在讨论的一般情况中,$C<n^2-n+1$,因此这种情况下,矩阵 D_p 中,除去主对角线的元素,一定存在 2 个或多个元素表示同一个差分值的概率的情况.所以,对于一般情况的差分变量 Y,可以通过情况 2 对应的 Y 通过合并操作得到.根据定理 2,合并操作会使熵值减小,因此,对于更一般的情况:

$$DEN(X)<DEN_{\max}(X).$$

然后我们考虑情况 1.情况 1 对应的差分概率矩阵中,位于同一条斜线的元素表示同一个差分值的概率,而这里讨论的一般情况,Y 的取值个数是大于情况 1 中 Y 的取值个数.说明这里对应的差分值矩阵中位于同一条斜线上的元素值不再相等.因此,一般情况的 Y 变量可以看做是由情况 1 中的 Y 变量通过拆分操作得到.根据定理 2,拆分操作会使熵值增大,我们可以得到:

$$DEN(X)>DEN_{\min}(X).$$

此时,可以得到下面的定理.

定理 3. 对于离散型随机变量 X,不论其数值特性如何,其差分熵 $DEN(X)$都满足下面的式子:

$$EN(X)<DEN(X)<2EN(X).$$

2.4.3 分布特性对差分熵的影响

本节讨论随机变量分布特性对于差分熵的影响.对于更一般的情况分析还有待研究.这里仅考虑特殊情况——二值随机变量,即 X 可能取的值只有两种,对应概率分别为 p 和 $1-p$.此时,可以写出 X 的熵值与差分熵值:

$$EN(X)=-p\log_2 p-(1-p)\log_2(1-p),$$
$$DEN(X)=-[p^2+(1-p)^2]\log_2[p^2+(1-p)^2]-2[p(1-p)]\log_2[p(1-p)].$$

此时,$EN(X)$和 $DEN(X)$均可以看做是概值 p 的函数,因此,我们可以采用分析函数的方法分析上面两个式子.求得导函数如下:

$$\frac{dEN}{dp}=\log_2\left(\frac{1}{p}-1\right),\frac{dDEN}{dp}=2(1-2p)\log_2\left[\frac{p^2+(1-p)^2}{p(1-p)}\right].$$

当 $p=1/2$ 时,两个导函数同时取 0 值,说明均匀分布时,熵值和差分熵值都达到了最大值;同时,当 $p<1/2$ 时,两个导函数均为正值;$p>1/2$ 时,两个导函数均为负数,这说明当随机变量的分布情况接近于均匀分布时,熵与差分熵最大.

熵值和差分熵值的原函数和导函数图像如图 9、图 10 所示.

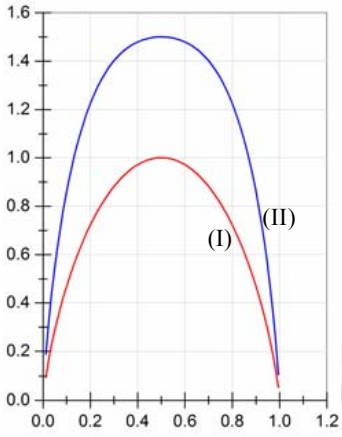

Fig.9 Image of entropy (line I) and difference entropy (line II) for binary random variable

图 9 二值随机变量的熵值(曲线 I)和差分熵值(曲线 II)的函数图像

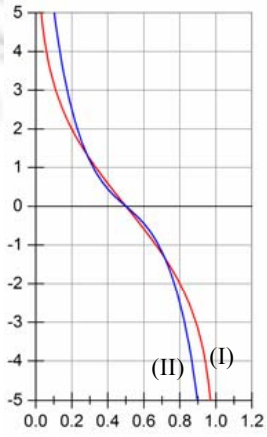

Fig.10 Image of derivative function of entropy (line I) and difference entropy (line II) of the binary random variable

图 10 二值随机变量熵值(曲线 I)和差分熵值(曲线 II)导函数图像

可以发现:两个导函数的图像具有 3 个交点,3 个交点的横坐标大约是 0.3,0.5,0.7(此处 0.3 和 0.7 为近似值),说明在[0.3,0.7]的范围内,熵值的变化速率比差分熵值要快;而[0,0.3]以及[0.7,1]的范围内,则差分熵值的变化速率更快.

接下来讨论熵值和差分熵值的差距,将它们的差分熵值和熵值做差如下:

$$DEN(X)-EN(X)=p\log_2 p+(1-p)\log_2(1-p)-[p^2+(1-p)^2]\log_2[p^2+(1-p)^2]-2[p(1-p)]\log_2[p(1-p)]$$

$$=(1-2p)[(1-p)\log_2(1-p)-p\log_2 p]-[p^2+(1-p)^2]\log_2[p^2+(1-p)^2].$$

做出以上差值函数图像如图 11 所示.

从图 11 中可以看到,图像的峰值大约处于 p=0.3 和 p=0.7 的位置,这两个位置也恰好也是熵和差分熵导函数的交点位置.从总的趋势上来看,中心部分的差值较大,两边则差值较小.这说明随机变量的分布情况越接近于均匀分布,差分熵和熵的差距就越大.

通过以上分析,我们可以得到下面的定理.

定理 4. 关于离散型随机变量 X,具有下面的结论.

(1) X 的分布越接近均匀分布,熵值和差分熵值就越大;

(2) X 的差分熵值永远大于 X 的熵值;同时,X 的随机性增大时,熵值和差分熵值的差距也在逐渐增大.在平均分布的附近会达到最大值;

(3) 如果 X 的分布接近均匀分布,那么在改变 X 的概率分布时,熵值比差分熵值对于分布情况变化更加敏感;如果 X 的分布是倾斜的,不同的取值对应的概率差距很大,那么在改变 X 的概率分布时,差分熵值比熵值对于分布情况的变化更加敏感.

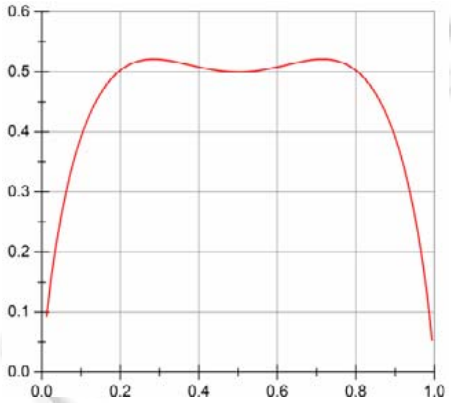

Fig.11 Image of the difference result entropy and difference entropy of binary random variable

图 11 二值随机变量熵值与信息熵值的差分函数图像

2.5 小 结

本节描述了差分熵的定义以及相关的特性.通过上文的分析可知,差分熵会同时受到数据的数值特性和分布特性的影响,并在某些情况下,差分熵值对于数据特性变化表现出了更高的敏感度.

3 基于差分信息熵的网络时序型隐蔽信道检测算法

本节提出基于差分信息熵的网络时序型隐蔽信道检测算法.首先需要说明的是:Cabuk 和 Gianvecchio 提出的算法均假设正常数据的随机性比隐蔽信道数据大[21,30],但是笔者认为,正常信道的特征取决于实际的网络环境,如果网络通畅,数据包的间隔的抖动可能会很小,此时包间隔数据的随机性也就会很小.因此,我们不可以对正常网络的数据特征进行任何的假设,而应该通过实验对正常数据进行特征提取,然后从待测数据中提取特征后,与正常数据的特征做比对,决定其中是否包含隐蔽信道.

基于以上的思想,设计基于差分信息熵的网络时序型隐蔽信道检测算法的思路如下.

(1) 收集网络中包时间间隔数据,构造时间间隔序列:$\{s_1,s_2,...,s_n\}$;

(2) 数据分类,根据一定的规则,将数据划分为 m 类,记作 $c_1,c_2,...,c_m$,并对每一个类赋予一个代表该类数据的数值:$v_1,v_2,...,v_m$.分类后,同一个类中的数据不加区别,统一用该类的代表数值来表示.分类后的序列表示为 $\{t_1,t_2,...,t_n\}$,其中,$s_i\in c_j\rightarrow t_i=v_j$;

(3) 将分类后的序列划分成大小为 w 的多个子窗口,如果采集到的数据量为 n,那么可以划分的窗口个数为 $\lfloor n/w\rfloor$,并经每个窗口中的序列记作:$\{t_{i,1},t_{i,2},\ldots,t_{i,w}\}$;

(4) 计算每个窗口的熵值 EN_i.统计窗口中每个数值出现的比率,计算熵值 $EN_i=\sum_{j=1}^{m}\frac{cnt_{i,j}}{w}\log_2\frac{cnt_{i,j}}{w}$,其中,$cnt_{i,j}$ 表示检测窗口 i 中属于 c_j 类的数据的个数;

(5) 计算每个窗口序列的差分序列,记作:$\{p_{i,1},p_{i,2},\ldots,p_{i,w-1}\}$,其中,$p_{i,j}=t_{i,j}-t_{i,j+1}$;

(6) 计算每个窗口的差分熵值,记作 DEN_i.根据上一节的分析,对原始数据划分 m 类后,差分运算后的数据最多有 m^2-m+1 类,最少则有 $2m-1$ 类.这里记差分运算后的数据类簇个数为 m',并将差分值的各个

类簇记作 $d_1,d_2,\ldots,d_{m'}$,然后计算差分熵值 $DEN_i=\sum_{j=1}^{m'}\frac{cnt'_{i,j}}{w-1}\log_2\frac{cnt'_{i,j}}{w-1}$,其中,$cnt'_{i,j}$ 表示检测窗口 i 中属于 d_j 类的差分值的个数;

(7) 将每个窗口的计算结果和正常信道的对应结果 EN_0 以及 DEN_0 比较,根据给定的偏差值 ε_E 和 ε_D,如果 $|EN_i-EN_0|\leqslant\varepsilon_E$ 并且 $|DEN_i-DEN_0|\leqslant\varepsilon_D$,说明当前检测窗口的数据中没有隐蔽信道;否则,可以认为当前检测窗口的数据可能包含隐蔽信道.

以上是算法的基本思路.

4 算法实现和参数设定

本节说明算法在实现过程涉及到的一些参数确定,我们将通过实验来确定最优的参数设定方案.通过前文对于算法原理的描述可知,待确定的参数值有:划分类簇的个数 m、每个类簇的代表值 v_i、窗口大小 w 以及正常信道的阈值 EN_0,DEN_0 和对应的偏差值 ε_E 和 ε_D.下文中,我们首先分析正常数据的特点,然后确定这些参数值.

4.1 正常信道数据分析

我们通过程序实时抓取网络数据包,统计包时间间隔的分布情况.实验结果表明:当前的实验室环境下,网络数据包的时间间隔主要分布在(75,1075]范围内.这里,为了便于数据分析,我们将数据包间隔的范围划分为 40 个长度为 25 的子区间,记作 $R_1,R_2,\ldots,R_n$,并将每个区间内数据的频率记作 p_i.而落在(75,1075]范围以外的数据被认为是离群点,不作为实验数据.统计落在各个子区间内的数据所占比例,如图 12 所示.

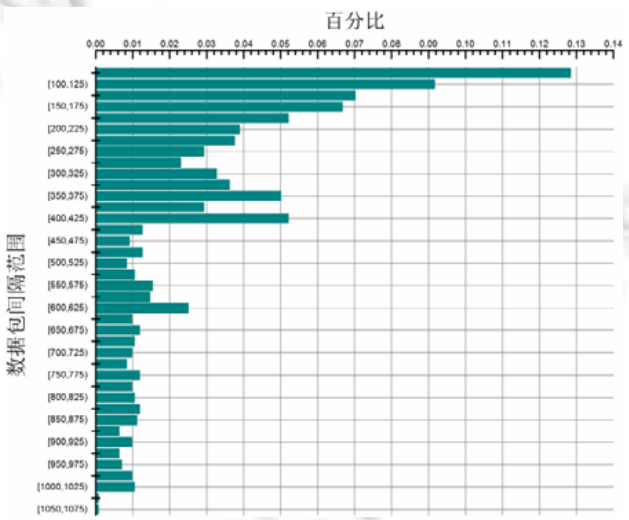

Fig.12 Bar-chart of the packet interval of legitimate channel

图 12 正常网络数据包时间间隔统计

接下来,用指数分布(exponential distribution)来描述正常数据包的分布情况,指数分布的密度函数如下:

$$f(x)=\lambda e^{-\lambda x}.$$

这里涉及到了参数 λ,可以采用一阶矩估计法对参数 λ 进行估计:

$$\frac{1}{\lambda}=\overline{v}_i=\sum_{i=1}^{40}p_iv_i.$$

代入我们的实验数据,可以得到:$\lambda\approx0.0029$.

于是,正常数据包的时间间隔分布的密度函数如下:

$$f(x)=0.0029\cdot e^{-0.0029x}.$$

4.2 数据离散化

至此,前文中所有的分析和说明都是基于离散型的随机变量,但是在实际网络环境中,数据包的时间间隔可以看做是一个连续型的随机变量.为了便于发现数据的特征,简化数据处理过程,我们首先需要对数据进行离散化,具体方法就是对数据进行分类,将连续型的随机变量按照取值范围划分类簇.数据离散化的过程对应于第 3 节算法执行步骤中的步骤(2).

数据离散化过程中,划分类簇的个数将会直接影响后续算法的处理效果.划分类簇过少,使得数据中的某些特征丢失;而划分类簇过多,则增加了算法处理的复杂度,影响检测效率.显然,我们知道这样两个事实:划分的类簇越多,就需要越多的测试数据才可以将数据中的特征表现出来,对于差分熵而言,最坏情况下,差分结果的类簇个数会随着原始数据的类簇个数的增长而呈现平方式的增长,如果数据量不够,则会使得计算得到的差分熵值不准确.

图 13 给出了对于有 5 个类簇均匀分布的情况下,熵值和差分熵值随着样本容量增大时的变化情况.可以看到:一开始,随着样本容量的增大,熵值和差分熵值也在显著增大;而当样本个数大于 150 时,熵值开始趋向于稳定;当样本个数大于 950 时,差分熵值也在开始趋向稳定.所以可以说:当样本个数大于 950 时,样本大小基本不会影响实验结果,此时的结果主要取决于数据本身的特性.这里,我们称可以包含全部数据特征的最小样本容量为有效样本容量(adequate sample size).

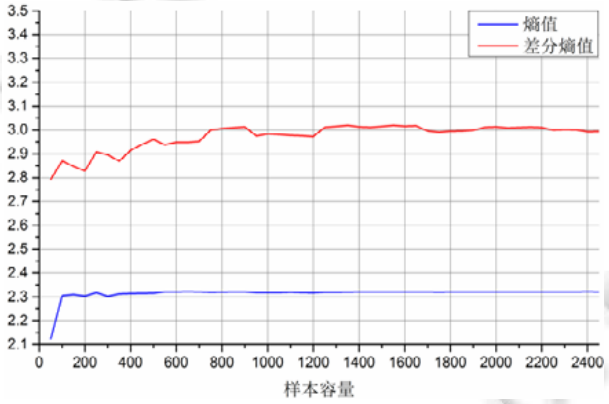

Fig.13 Influence of sample size on the entropy and difference entropy (5 bins)

图 13 样本个数对于熵值和差分熵值的影响(类簇个数:5)

表 2 记录了我们通过实验测试得到的对于不同的类簇个数下的有效样本容量的大小.

Table 2 Adequate sample size of different amount of bins

表 2 类簇个数与有效样本容量大小

类簇个数	有效样本容量
5	950
8	1 450
11	1 550
14	2 300
17	2 350
20	2 350
23	2 400
26	2 500
29	2 900

可以看到,划分的类簇个数越多,有效样本容量也就越大.兼顾检测效果和检测效率,我们决定将实验数据划分为 10 个类簇;同时,窗口大小设置为 1 500.即:m=10,w=1500.

Gianvecchio 在文献[30]提出了一种数据离散化的方式,根据数据包时间间隔的概率分布划分类簇,使得数

据落入每一个类簇中的概率相等.作者认为,这样划分类簇的好处是使得程序可以在常数时间内决定采集到的每一个数据项属于哪个类簇.如果将数据划分 m 类,并用 $F(x)$表示时间间隔数据的分布函数,那么落入每个类簇的概率为 $1/m$;同时,对于某个特定的数值 x,可以通过$\lfloor F(x)/m \rfloor$来计算数据落入的类簇.

这里,我们按照正常网络中的数据分布情况来对数据进行等概率划分类簇,类簇个数为 10,因此落入每个类簇的概率是 10%.我们需要找到正常数据对应指数分布的 10%,20%,...,90%的分位点,来确定每个类簇的边界.这样处理的好处是不管正常信道中的包间隔数据的分布情况是怎样的,数据离散化处理后,正常的数据分布情况一定是均匀分布,可以统一处理正常数据的分布特性.这里确定 10 个类簇的范围见表 3.

Table 3 Binning strategy for data discretization

表 3 数据离散化划分类簇方案

类簇编号	取值区间	代表数值
1	(0,36)	18
2	[36,76)	55
3	[76,123)	101
4	[123,176)	148
5	[176,239)	207
6	[239,316)	276
7	[316,415)	362
8	[415,555)	481
9	[555,794)	660
10	[794,+∞)	1 139

数据离散化后,落入每个区间的数据均由该区间的代表数值来表示.代表数值通过计算每个区间的平均值得到,对于类簇 c_i,给定其区间的左右边界 a_i,b_i,可以利用下面的式子计算代表数值 v_i:

$$v_i = \int_{a_i}^{b_i} 0.0029x \cdot \mathrm{e}^{0.0029x} \mathrm{d}x.$$

4.3 正常信道数据特征提取

根据上文中的参数设定,我们可以将算法编程实现,提取正常信道的数据特征.这里,我们通过程序实时抓包得到了 24 610 个 IP 数据包时间间隔数据,共可以划分成 16 个大小为 1 500 的检测窗口.对于每个窗口的数据离散化后,计算熵值和差分熵值,得到的结果见表 4.

Table 4 Testing result of data from legitimate channel

表 4 正常信道数据的检测结果

窗口编号	熵值	差分熵值
1	0.720	0.923
2	0.766	0.970
3	1.432	1.770
4	1.480	1.857
5	1.322	1.601
6	1.364	1.582
7	1.531	1.837
8	1.636	2.015
9	1.355	1.711
10	1.578	1.918
11	1.472	1.949
12	1.581	2.107
13	1.484	1.881
14	1.636	2.037
15	1.576	1.986
16	1.582	2.062

求得各个检测窗口的熵值和差分熵值数据后,可以通过计算这些数据的平均值提取正常信道数据的特征,确定算法中的参数 EN_0 和 DEN_0 的值,得到的结果为:EN_0=1.407,DEN_0=1.762.

最后确定ε_E 和ε_D 的值.这里,根据 3 倍标准差的准则,确定它们的值如下:ε_E=0.831,ε_D=1.062.

以上是算法实现过程中的一些参数设定.

5 实验测试及结果分析

本节通过实验检验算法的性能和效果.我们通过程序模拟构建包含隐蔽信道的异常数据,然后用我们提出的算法进行检测,统计检测隐蔽信道的准确率.

5.1 对照实验参数设定

对于对照实验参数的设定,我们将采用 4 种检测算法作为对照组,分别是 Cabuk 提出的ε相似度算法和随机性检测算法、Gianvecchio 提出的 CCE 算法以及熵评估算法.这里需要说明的是,熵评估算法是指直接考察待测数据的熵值大小来判定待测数据中是否包含隐蔽信道(即:计算待测数据的熵值低于指定阈值,则认为待测数据包含隐蔽信道).Gianvecchio 在其实验中将熵评估算法作为 CCE 算法实验的对照实验,进而说明 CCE 算法的先进性.在这里,熵评估算法也是我们提出的差分熵算法的一部分,因此,这里也效仿 Gianvecchio 的做法,将熵评估算法作为实验的对照组,验证差分信息熵算法的效果.通过对照实验,评估差分熵在检测隐蔽信道过程中的作用.利用本文提出的差分熵检测算法与以上 4 种检测算法对同样的数据进行检测,通过对比检测结果,评估本文提出的算法的性能和效果.以上 4 种算法的实现原理前文中已有说明,这里主要说明算法中相关参数的设定.

- 随机性检测

随机性检测算法通过考察待测数据标准差变化情况,来判定待测数据中是否包含隐蔽信道.Cabuk 在提出该算法时,在 2 000 的检测窗口下,设定子窗口大小为 250 和 100 进行了实验.而 Gianvecchio 在文献[30]中介绍 CCE 算法时,也将随机性检测算法作为对照组,在其实验中,作者同样选择 2 000 大小的检测窗口,以 100 为子窗口进行了随机性检测算法的对照实验.根据作者给出的实验结果数据,在当前参数设定下,随机性检测算法可以对正常数据和隐蔽信道数据达到很好的区分.基于以上因素,我们选择 100 为子窗口大小进行随机性检测算法的对照实验.此时,对于正常数据的 16 个检测窗口,每个检测窗口数据均可以通过算法得到 15 个不同的标准差值,通过计算任意两个标准差值的相似度值,进而得到 105 个数据,计算这些数据的标准差,就是该算法的输出结果.通过计算 16 个检测窗口结果的平均值,就可以得到我们实验中用到的对照阈值.我们通过计算,得到该阈值大小为 0.07,检测结果低于该阈值的数据均被认为包含隐蔽信道.

- ε相似度

Cabuk 在文献[21]中对于ε相似度算法的测试中,将参数ε设定为 0.005,0.008,0.01,0.02,0.03 和≥0.1 共 6 种不同的值,并将算法检测的阈值设定为$\mu+\sigma$,$\mu+1.5\sigma$,$\mu+2\sigma$,Max 共 4 种不同的值,其中μ,σ,Max 分别表示ε相似度算法检测合法数据所得结果的均值、标准差以及最大值.Cabuk 对于上面每一种ε和阈值的组合进行了测试.根据作者给出的实验结果,当检测阈值设定为$\mu+\sigma$和$\mu+1.5\sigma$时,对于每一种ε参数的设定,检测效果都比较好.此时,对于隐蔽信道的检测可以达到 10%以下的漏检率,对于正常信道的检测则可以达到 30%以下的误检率.我们的实验设定参考 Cabuk 的实验过程,将ε参数设定为 0.02,阈值则设定为$\mu+1.5\sigma$.通过对正常数据的测试,我们ε相似度算法的对照实验中的测试阈值设定为 0.973.测试结果高于此阈值时,被认为可能包含隐蔽信道.

- CCE

对于 CCE 算法,Gianvecchio 提出了一种基于树形结构的程序实现方式,使得程序时间复杂度可以保持在$O(n\cdot m\cdot\log(Q))$水平,同时,空间复杂度为 $O(n\cdot m)$[30],其中,n,m,Q 分别表示测试的数据总量、考察的最长序列长度以及数据离散化过程中划分类簇的个数.我们在前文中已经对于数据离散化作出说明,这里我们同样设定 Q 值为 10;同时,为了兼顾检测效果和检测效率,我们设定考察的最长序列为 10.通过对正常数据的测试,我们可以得到测试结果阈值为 0.439.检测结果低于该阈值的数据,均被认为可能存在隐蔽信道.

- 熵评估算法

熵评估算法直接考察待测数据的信息熵值来判定待测数据是否包含隐蔽信道.前文中已经说明,引入熵评估算法作为当前实验的对照组的目的,是对差分熵在隐蔽信道检测中所起到的作用进行更深入地评估,熵评估算法过程是本文提出的差分熵算法过程的子过程.这里的阈值前文中已经给出,大小为 1.407,低于该阈值的数据均认为其中包含隐蔽信道.另外,熵评估算法在检测过程中同样需要数据离散化的步骤,离散化的过程与差分

熵算法使用的离散化过程是相同的,具体步骤前文中已有说明,这里不再赘述.

可见,我们当前实验的网络环境中的数据包时间间隔的随机性非常小.这一点,和 Cabuk 以及 Gianvecchio 提出的“正常数据随机性比隐蔽信道数据高”的假设是不一致的.正如前文所述,正常信道数据的特征由网络环境决定,我们不可以在没有进行实际测试的情况下,对正常信道数据的特征给出任何的假设.

5.2 实验测试数据的收集

实验中,我们主要对 IPCTC,TRCTC,JitterBug 这 3 种类型的时序型隐蔽信道的数据进行了检测,这 3 种类型的隐蔽信道的实现原理前文已有说明.测试数据通过编程模拟得到.

- IPCTC

Cabuk 在其 IPCTC 隐蔽信道的实现中,模拟了 IPCTC 隐蔽信道的 3 种形式:单一时间间隔、多种时间间隔、含噪声.单一时间间隔的 IPCTC 隐蔽信道仅采用了 1 个固定的时间间隔值,如要发送比特 1,则在该时间间隔内发送一个数据包,否则,发送端保持沉默;多种时间间隔的 IPCTC 则采用了多个不同的时间间隔的值,发送端每隔 t 个数据包就调整一次发送的时间间隔,这里的时间间隔之间的切换方式可以是轮换也可以是随机选择;第 3 类隐蔽信道则是在信道中引入了噪声.在我们的实验中,共实现了 4 种不同的 IPCTC 隐蔽信道,其中:单一时间间隔的类型实现了两种,时间间隔分别为 20ms 和 80ms;多个时间间隔的类型实现了 2 种,时间间隔采用{20ms, 40ms,60ms}这 3 种,时间间隔的切换方式有轮换切换和随机切换两种方式,切换的频率为 50,即:每隔 50 个数据包,就切换一次时间间隔.对于每一种 IPCTC 信道,我们通过程序生成了 200 000 条数据.具体情况见表 5.

Table 5 Testing data of IPCTC for experiment

表 5 IPCTC 隐蔽信道实验数据

数据编号	采用的时间间隔	时间间隔切换方式
IPCTC-1	{20}	–
IPCTC-2	{80}	–
IPCTC-3	{20,40,60}	轮换
IPCTC-4	{20,40,60}	随机

- TRCTC

TRCTC 隐蔽信道是 IPCTC 的改进,其发送数据的时间间隔不再是随意确定,而是通过对合法数据采样得到.用于发送比特 0 的时间间隔的集合记作 S_0,用于发送比特 1 的时间间隔的集合记作 S_1.集合 S_0 和 S_1 互不相交,二者的总和就是合法信道中所有出现过的时间间隔值.这里主要研究了基于不同的比例划分下指定 S_0 和 S_1 集合构成的 TRCTC 隐蔽信道.根据 S_0 在全集中所占的比例,我们构建了 3 条不同的 TRCTC 隐蔽信道,同样对于每一条信道,通过程序生成 200 000 条数据.具体情况见表 6.

Table 6 Testing data of TRCTC for experiment

表 6 TRCTC 隐蔽信道实验数据

数据编号	S_0 集合中的时间间隔	S_1 集合中的时间间隔	正常信道数据落入 S_0 集合的频率(%)
TRCTC-1	{50,74,99,120,155,190,220,300}	{325,388,500,1000}	30
TRCTC-2	{53,72,100,117,150,220,293,500,520}	{550,570,645,750,810,1000}	40
TRCTC-3	{50,155,190,220,300,325,388,500,520,673,720}	{795,873,1000}	50

表中,S_0 和 S_1 集合中的时间间隔数据通过这样的方式得到:首先统计正常信道中出现的时间间隔值以及每个值出现的频率,然后根据规定的比例,选择适当的数值构成集合 S_0,剩余的数值构成 S_1.

- JitterBug

JitterBug 隐蔽信道的数据通过修改正常信道的数据得到.其中,时间参数ω的选择直接决定了信道的效果.本文提出差分信息熵检测算法的一个目的就是弥补 CCE 算法在检测 JitterBug 隐蔽信道时的不足.JitterBug 隐蔽信道的检测是我们实验研究的主要内容.下文中,我们取ω值为 10,20,30,...,1000 共 100 个不同的值,构建了 100 条不同的 JitterBug 隐蔽信道.同样地,对于每一条隐蔽信道,我们会通过程序生成 200 000 条数据.

与差分熵算法的参数设定相同,实验中作为对照实验的 4 种算法在检测时也将检测窗口确定为 1 500,因此,

200 000 条数据就可以构成 133 个检测窗口.

5.3 实验结果及分析

5.3.1 IPCTC 隐蔽信道检测实验结果

表 7~表 11 分别显示了ε-相似度算法、CCE 算法、随机性检测算法、熵评估算法以及差分熵算法对于 IPCTC 隐蔽信道的检测效果.表中给出了算法检测 133 个检测窗口数据所得结果的平均值和标准差以及每个算法判断规则下的检测率.表 7 给出了各个窗口中小于ε参数的相似度值比例的均值和标准差;表 8 给出了 CCE 算法检测得到的每个窗口的熵率值的均值和标准差;表 9 给出了每个窗口的随机性检测结果的均值和标准差;表 10 给出了各个窗口熵值的均值和标准差;最后,表 11 给出了差分熵检测算法检测得到的熵值和差分熵值的均值和标准差.

Table 7 Testing result of ε-similarity algorithm in IPCTC data

表 7 IPCTC 隐蔽信道检测实验ε相似度算法检测结果

隐蔽信道数据	均值	标准差	检测率(%)
IPCTC-1	0.960	0.001 8	1.50
IPCTC-2	0.965	0.001 8	3.75
IPCTC-3	0.959	0.001 7	3.00
IPCTC-4	0.959	0.001 7	4.51

Table 8 Testing result of CCE algorithm in IPCTC data

表 8 IPCTC 隐蔽信道检测实验 CCE 算法检测结果

隐蔽信道数据	均值	标准差	检测率(%)
IPCTC-1	1.365	0.015	7.51
IPCTC-2	2.665	0.011	9.02
IPCTC-3	2.058	0.017	11.2
IPCTC-4	1.999	0.065	8.27

Table 9 Testing result of randomness testing algorithm in IPCTC data

表 9 IPCTC 隐蔽信道检测实验随机性检测算法检测结果

隐蔽信道数据	均值	标准差	检测率(%)
IPCTC-1	0.110	0.025	1.45
IPCTC-2	0.109	0.029	2.93
IPCTC-3	0.280	0.099	2.55
IPCTC-4	0.372	0.137	3.08

Table 10 Testing result of entropy testing algorithm in IPCTC data

表 10 IPCTC 隐蔽信道检测实验熵评估算法检测结果

隐蔽信道数据	均值	标准差	检测率(%)
IPCTC-1	1.433	0.025	10.27
IPCTC-2	2.786	0.021	12.85
IPCTC-3	2.287	0.051	9.46
IPCTC-4	2.272	0.081	13.05

Table 11 Testing result of difference entropy algorithm in IPCTC data

表 11 IPCTC 隐蔽信道检测实验差分熵算法检测结果

隐蔽信道数据	熵均值	熵标准差	差分熵均值	差分熵标准差	检测率(%)
IPCTC-1	1.433	0.025	2.456	0.049	5.26
IPCTC-2	2.786	0.021	5.121	0.052	100
IPCTC-3	2.287	0.051	3.968	0.115	100
IPCTC-4	2.272	0.081	3.942	0.176	100

可以看到,差分熵算法的检测效果明显好于ε相似度算法、随机性检测算法以及 CCE 算法.主要原因在于其他两种算法都是假设正常数据的随机性比隐蔽信道数据高,因此两种算法的策略都是检测结果低于阈值时判

定待测数据中包含隐蔽信道.我们通过实验证明事实并非如此,在较为流畅的网络环境下,信道中的数据流动相对稳定,此时信道中的数据包传输速率也比较均匀,数据包的时间间隔也基本上为恒定值,随机性较小.本文提出的算法不再对正常信道的特征做出假设,将算法处理的结果作为信道的一种特征,通过待测数据和正常数据特征的差异程度来判定待测数据是否异常,而并非通过量化的大小关系来确定是否包含隐蔽信道,因此检测的准确率较高.

另外,从表中数据还可以看到:尽管熵评估算法是本文提出的差分熵算法的子过程,但是熵评估算法的检测效果和差分熵算法有很大的差别.我们通过实验发现:仅仅通过评估信息熵值,对于正常数据和隐蔽信道数据已经可以达到很好的区分能力.然而,Gianvecchio 在文献[30]中描述的熵评估算法依然采用比较阈值的方式来决定检测结果,基于作者“低于阈值的数据包含隐蔽信道”的思路,导致熵评估算法的检测准确率很低.实验中,我们还将熵评估算法进行了改进,将判定数据是否包含隐蔽信道的标准修改为$|EN_i-EN_0|\leqslant\varepsilon_E$,即采用差分熵算法的部分判定标准,此时,熵评估算法对于 4 组 IPCTC 隐蔽信道的检测率分贝为 5.03%,95.72%,92.30%以及 93.43%.该结果同样低于差分熵算法,主要原因是信息熵对于待检测数据的分布特性的评估比较充分,而对于待测数据的数值特性的评估能力比较欠缺.则也说明了采用差分信息熵在隐蔽信道检测过程中作为评估标准的必要性.

从实验数据可以看出,当选择单一的时间间隔,并且时间间隔较小时,产生的隐蔽信道的数据特征会和正常信道的数据特征很相似,检测的难度就会加大.对于 IPCTC-1 这组数据,3 种检测算法的效果都不太理想.当增加 IPCTC 信道中的传输时间间隔,或者引入多个可选的时间间隔时,数据的随机性会显著提高.Cabuk 认为,引入多个时间间隔可使 IPCTC 隐蔽信道的隐蔽性增强[21],这个结论的依据是假定正常数据随机性较高.通过上文的实验结果可以看出:3 个算法对于 IPCTC-3 和 IPCTC-4 这两组数据的检测准确率反而比前面两组数据更高一些.

5.3.2 TRCTC 隐蔽信道检测实验结果

表 12~表 16 总结了 TRCTC 隐蔽信道检测实验的结果.

Table 12 Testing result of ε-similarity algorithm in TRCTC data

表 12 TRCTC 隐蔽信道检测实验ε相似度算法检测结果

隐蔽信道数据	均值	标准差	检测准确率(%)
TRCTC-1	0.926	0.000 0	0.00
TRCTC-2	0.906	0.000 0	0.00
TRCTC-3	0.913	0.000 0	0.00

Table 13 Testing result of CCE algorithm in TRCTC data

表 13 TRCTC 隐蔽信道检测实验 CCE 算法检测结果

隐蔽信道数据	均值	标准差	检测准确率(%)
TRCTC-1	2.790	0.005 5	3.75
TRCTC-2	2.702	0.011 8	4.51
TRCTC-3	2.252	0.009 6	7.52

Table 14 Testing result of randomness testing algorithm in TRCTC data

表 14 TRCTC 隐蔽信道检测实验随机性检测算法检测结果

隐蔽信道数据	均值	标准差	检测准确率(%)
TRCTC-1	0.072	0.003 7	10.05
TRCTC-2	0.075	0.010 9	9.52
TRCTC-3	0.079	0.009 4	12.27

Table 15 Testing result of entropy testing algorithm in TRCTC data

表 15 TRCTC 隐蔽信道检测实验熵评估算法检测结果

隐蔽信道数据	均值	标准差	检测准确率(%)
TRCTC-1	2.871	0.016 8	2.37
TRCTC-2	2.785	0.019 8	1.58
TRCTC-3	2.362	0.038 0	2.09

Table 16 Testing result of difference entropy algorithm in TRCTC data

表 16 TRCTC 隐蔽信道检测实验差分熵算法检测结果

隐蔽信道数据	熵均值	熵标准差	差分熵均值	差分熵标准差	检测率(%)
TRCTC-1	2.871	0.016 8	5.350	0.046 0	100
TRCTC-2	2.785	0.019 8	5.167	0.046 8	100
TRCTC-3	2.362	0.038 0	4.436	0.072 0	100

TRCTC 采用的数据包时间间隔来自正常数据的采样结果,因此在数值特征上,TRCTC 数据和正常信道数据有类似之处.但是 TRCTC 的主要问题在于没有考虑到每个时间间隔数值在正常信道中的分布情况.实验中,我们根据集合 S_0 和 S_1 中的数据在正常信道中的比例构建了 3 组 TRCTC 隐蔽信道数据,从实验结果中可以看出:构建集合 S_0 和 S_1 使得正常信道中的数据落入两个集合的概率相等时,数据的特征和正常信道的数据最相似,所以相比另外两组数据,5 个算法对于 TRCTC-3 这组数据的检测结果都更接近于正常信道.但是其结果依然和正常数据有较大的差异,原因在于 S_0 和 S_1 的二类划分仅仅是一种粗粒度的划分,同在 S_0 集合中的数据在正常数据中出现的概率也可能是不同的,但 TRCTC 没有在构建信道时没有考虑到这一点,因此在统计特性上依然会表现出和正常数据较大的差异.

同样可以很明显地看出,差分熵检测算法在 TRCTC 隐蔽信道的检测上有非常好的效果.对于 ε 相似度算法和 CCE 算法,TRCTC 隐蔽信道的数据也表现出了和正常数据不同的特征,检测结果表明,TRCTC 数据的随机性要比正常数据高.这和两个算法中“隐蔽信道数据的随机性低于正常信道数据”的思想是相悖的,因此检测的效果不理想.

观察表 14 还可以发现,随机性检测算法对于 3 种 TRCTC 隐蔽信道的检测结果和正常信道非常接近.这说明随机性检测算法对正常信道和 TRCTC 隐蔽信道的区分能力很低,而检测结果却好于该算法在 IPCTC 隐蔽信道的实验效果,我们通过实验发现:就区分能力而言,随机性检测算法对于 IPCTC 隐蔽信道的识别能力更高,然而,基于正常信道随机性高的假设使得该算法在 IPCTC 数据中表现较差.这也再一次说明了在没有进行实验验证的情况下,对于正常信道数据情况进行假设的做法是不可取的.

对于熵评估算法,这里可以得到和 IPCTC 实验类似的结论.由于我们的实验环境中正常信道的随机性很低,我们的实验结果中,正常信道的熵值低于隐蔽信道数据,因此熵评估算法的检测率很低.通过我们改进后的熵评估算法得到的检测率为 92.43%,95.25%,97.32,同样低于差分信息熵算法.

5.3.3 JitterBug 隐蔽信道检测实验结果

IPCTC 将要传输的隐秘信息转换为指定的时间区间内是否发送数据包的事件,IPCTC 产生的数据包间隔完全没有参照正常数据,因此得到数据的可能会呈现出和正常数据完全不同的特征;TRCTC 数据中的数值通过正常数据采样得到,因此 TRCTC 的数据和正常数据具有相似的数值特性,但是 TRCTC 在构建信道过程中忽视了通过采样得到的每个数值的出现概率,这将导致 TRCTC 的数据呈现出不同于正常数据的统计特性.所以,IPCTC 和 TRCTC 的抗检测能力都比较差.

JitterBug 隐蔽信道通过对正常数据包的时间间隔做出细微的修改来编码隐秘信息.选择合适的参数 ω 的值,就可以尽可能小地减少构建隐蔽信道过程中对于正常数据统计特性的影响.因此,JitterBug 隐蔽信道具有比 IPCTC 和 TRCTC 更强的抗检测性.但是由于 JitterBug 的数据包间隔均为 $\omega/2$ 的倍数,因此 JitterBug 产生的包间隔数据具有较为明显数值特性,因此就可以利用这样的特征来检测 JitterBug 隐蔽信道.

对 JitterBug 隐蔽信道的检测是我们重点要研究的内容.实验中,我们为参数 ω 赋予不同的值,得到了 100 组不同的 JitterBug 隐蔽信道数据,然后比较 3 个算法对这 100 组数据的检测效果.

ε 相似度算法的检测结果如图 14 所示.

从图中可以看出,随着参数 ω 值的增大,小于 ε 参数值的相似度值比例也在逐渐增加;并且在 ω 处于[10,340]范围内时,ω 的变化会显著影响 ε 相似度算法的检测结果.产生该现象的主要原因是:当 ω 的取值较小时,得到的 JitterBug 数据的特征主要依赖于构建该隐蔽信道的原数据的特征;随着 ω 值的增大,JitterBug 的数据特征将同时受到 ω 值和原数据特征的影响;而当 ω 值继续增大时,原数据特征对于产生的 JitterBug 数据的影响在逐渐减弱,

因为更大的ω值意味着原始数据中更多的不同的值在构建 JitterBug 过程中映射到了同一个值,因此,数据的随机性在下降,数据之间的相似度也就逐渐在下降,于是就有更多的相似度值小于参数ε.

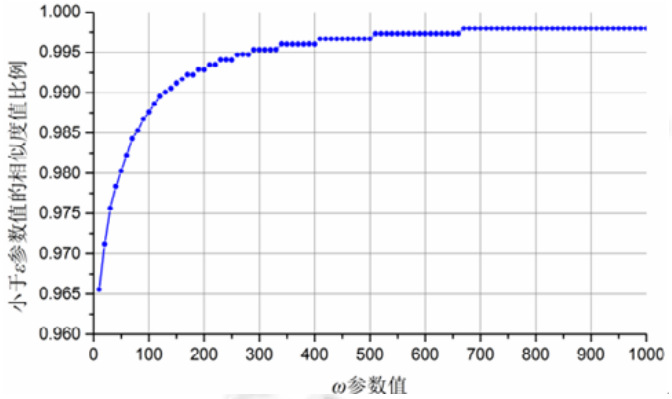

Fig.14 Results of ε-similarity algorithm in JitterBug

图 14 JitterBug 隐蔽信道检测实验ε相似度算法检测结果

基于正常信道检测得到的阈值为 0.973、高于该阈值的检测结果被认为包含隐蔽信道的思路,我们可以看到:当ω值大于 50 时,ε相似度算法可有效地检测出 JitterBug 隐蔽信道;ε值小于 50 时,算法对于 JitterBug 隐蔽信道的识别能力很低.然而基于常理推断,ω值很大时,隐蔽信道数据的异常特征更加明显.因此,过大的ω构建的隐蔽信道实用性并不强,我们更关注算法对于ω值较小时生成的 JitterBug 隐信道的检测效果.

随机性检测算法对于 JitterBug 隐蔽信道的检测结果如图 15 所示.

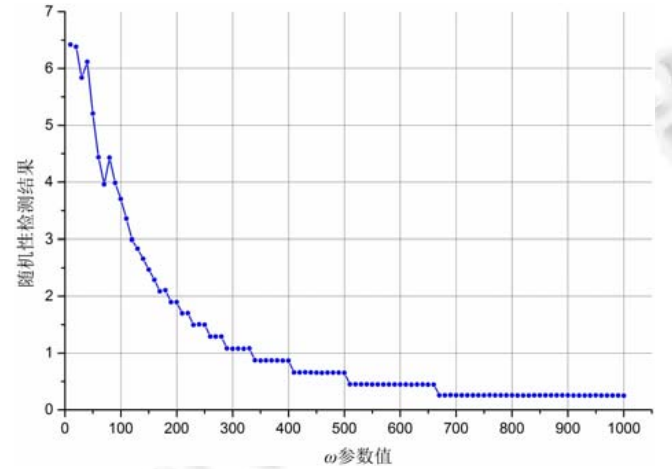

Fig.15 Results of randomness testing algorithm in JitterBug

图 15 JitterBug 隐蔽信道检测实验随机性检测结果

从实验结果中可以看出:随着ω值的增大,随机性检测算法的检测结果在逐渐减小.然而通过观察实验结果,当ω取值为 1 000 时,实验结果取到最小值,然而其值仍然大于检测阈值 0.07.这充分说明随机性检测算法对于 JitterBug 隐蔽信道的检测能力不足.事实上,当ω值很大时,随机性检测算法对于正常数据和 JitterBug 隐蔽信道数据的识别率很高.而在实际应用中,过大的ω值会减慢 JitterBug 隐蔽信道的传输速率,同时也更有可能暴露其异常数据的特征,因此,过大的ω参数设定下生成的 JitterBug 隐蔽信道并不实用;较小的ω参数值生成的 JitterBug 隐蔽信道同时具备高传输速率和高隐蔽性的特征,随机性检测算法对于这样的 JitterBug 数据的识别能力是可

观的.然而,由于算法定义小于阈值的数据为隐蔽信道数据,这就为算法识别隐蔽信道数据造成了误导,这里再一次证明了正常数据随机性高的假设是不合理的.

图 16、图 17 分别描述了 CCE 算法和差分熵算法对于 JitterBug 隐蔽信道的检测结果.

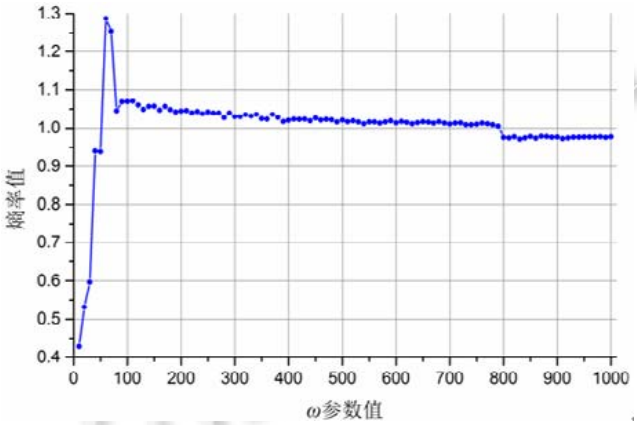

Fig.16 Results of CCE algorithm in JitterBug

图 16 JitterBug 隐蔽信道检测实验 CCE 算法检测结果

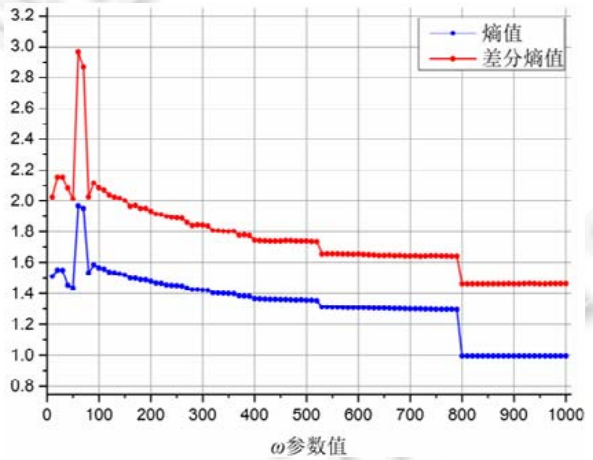

Fig.17 Results of difference entropy algorithm in JitterBug

图 17 JitterBug 隐蔽信道检测实验差分熵算法检测结果

可以看到:对于 CCE 算法得到的熵率值以及差分熵算法得到的熵值和差分熵值,三者图像的形状很相似.产生此结果的原因可能是 JitterBug 的数据是通过修改正常数据得到的,因此其分布特征会依赖于正常数据的分布特征.而实验中我们发现,正常数据的随机性较低,根据前文的分析,数据的随机性增大时,差分熵和熵值的差别也在逐渐增大,因此数据的随机性较小时,差分熵值和熵值就会有较为相似的结果.

我们还可以从图像中看出,当参数ω大于 100 时,CCE 算法和差分熵算法得到的结果都在随着ω的增加而逐渐下降;但相比差分熵值,当ω参数的值增大时,CCE 算法得到的熵率并未出现很明显的下降趋势.这说明差分熵值对于数据特征的变化更为敏感,这也意味着差分熵算法的灵敏度更高一些.

关于熵评估算法检测 JitterBug 隐蔽信道的实验情况,其结果在图 16 中已经有所呈现,图中的曲线 II 即为不同ω参数设定下的熵值.通过图像可以看到:类似于前面随机性检测算法的实验结果,虽然随着ω参数的增大,检测结果在逐渐减小,当ω参数大于 300 后,熵值结果将低于我们的阈值 1.40,此时熵评估算法对于 JitterBug 隐蔽

信道的检测效果比较好.然而,前文中已经说明,过大的ω值构建的 JitterBug 隐蔽信道的实用性并不强.因此,对于这些数据检测效果好并没有太大的意义.因此,Gianvecchio 提出的熵评估算法对于 JitterBug 隐蔽信道的检测能力比较弱.而改变熵评估算法的评定标准为$|EN_i-EN_0|\leqslant\varepsilon_E$后,所有的实验结果均会落在$EN_0\pm\varepsilon_E$的范围内.根据算法的定义,这意味着所有的数据均会被判定为正常数据,这说明改进判定规则后的熵评估算法依然无法有效地检测 JitterBug 隐蔽信道.

最后,根据每个算法判定是否包含隐蔽信道的准则,统计每个算法的检测准确率,如图 18 所示.

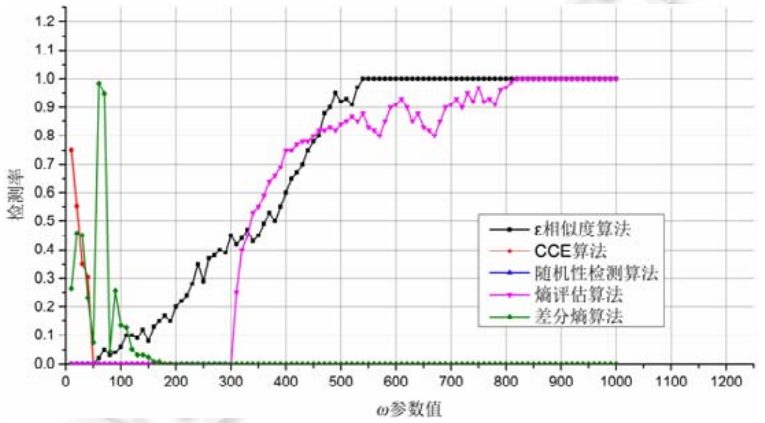

Fig.18 Detecting rate of JitterBug covert channel

图 18 JitterBug 隐蔽信道检测率

从图中可以看到:ω值大于 50 时,ε相似度算法检测效果较好;ω值在 60~190 的范围内,差分熵检测算法表现出了较好的检测效果;ω值在 0~60 范围内,CCE 算法的检测效果较好;而当ω大于 300 后,熵评估算法效果很好.

当ω值大于 200 时,CCE 算法和差分熵算法的检测率为 0.产生该现象的主要原因是:当ω值远远大于正常数据中的数值时,得到的 JitterBug 数据中的数值有很大的概率会落在原始数据的范围之外,此时再进行数据离散化时,就有很大的概率将所有的数据划分到同一个类簇中,于是得到的数据将会是常数序列.就这一点而言,数据离散化的过程会使得原始数据的主要特征丢失,因而导致检测算法失效.这也说明算法的的实现过程中还有一些问题,有待改进.

以上实验证明,差分熵检测算法在检测 IPCTC,TRCTC 以及 JitterBug 隐蔽信道方面表现出了比ε相似度算法、随机性检测算法以及 CCE 算法更好的效果.差分熵检测技术同时考察了待测数据的数值特性和分布特性,检测指标更加严格,能更深入地挖掘数据中存在的特征.对于 JitterBug 隐蔽信道,差分熵检测技术可以找到其数据中不同于正常数据的数值特性,从而达到较好的检测效果.相比之下,ε相似度算法和随机性检测算法考察了数据的数值特性,而数据的分布特性在算法对数据进行排序时和计算标准查过程中已经造成了部分丢失;CCE 算法和熵评估算法则主要考察了数据的分布特性,忽视了数据的数值特性.因此,这 4 种算法在检测 JitterBug 中效果不太理想.而熵评估算法那和ε相似度算法虽然在ω取较大值是表现出很好的效果,但是对于 JitterBug 隐蔽信道的检测,我们更关注ω取较小值是的情况.

6 总结与展望

本文基于前人的研究成果,提出了一种基于差分信息熵的网络时序型隐蔽信道检测技术.文中首先给出了差分信息熵的概念;然后,通过理论分析总结了一些差分信息熵所具备的性质;然后,提出基于差分信息熵的网络时序型隐蔽信道检测算法,并通过实验证明算法在 IPCTC,TRCTC 和 JitterBug 隐蔽信道的检测上具有较好的效果.本文所涉及的研究工作还存在一些问题,这也是今后研究工作的主要方向.

(1) 对于差分信息熵的研究,目前主要基于离散型随机变量.网络数据包的时间间隔数据从某种程度上可

以看作是连续型的随机变量,文中涉及到的数据离散化处理将连续型随机变量转化为离散型,但从实验结果也可以看到,该处理步骤会使得原始数据中的部分数值特性和分布特性丢失,使得检测结果产生误差;

(2) 文中对于差分信息熵的部分理论仅仅给出了不完全的分析,更严密、更准确的分析结果还有待研究;

(3) 实验测试了新算法对于 IPCTC,TRCTC 和 JitterBug 这 3 种隐蔽信道的检测效果,而对于其他类型的隐蔽信道的效果还需要更多的研究工作.

References:

[1] Lampson BW. A note on the confinement problem. Communications of the ACM, 1973,16(10):613−615.

[2] Tsai CR, Gligor VD, Chandersekaran CS. A formal method for the identification of covert storage channels in source code. In: Proc. of the IEEE Symp. on Security & Privacy. DBLP, 1987. 74−87.

[3] Wang YJ, Wu JZ, Zeng HT, Ding LP, Liao XF. Covert channel research. Ruan Jian Xue Bao/Journal of Software, 2010,21(9): 2262−2288 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3880.htm [doi: 10.3724/SP.J.1001.2010.03880]

[4] Zander S, Armitage G, Branch P. A survery of covert channels and countermeasures in computer network protocols. IEEE Communications Surveys & Tutorials, 2007,9(3):44−57.

[5] Handel TG, Sandford Ii MT. Hiding data in the OSI network model. In: Proc. of the Information Hiding, 1st Int'l Workshop. Cambridge: DBLP, 1996. 23−38.

[6] Kundur D, Ahsan K. Practical internet steganography: Data hiding in IP. 2003. In: Proc. of the Texas Workshop on Security of Information Systems, 2003.

[7] Danezis G. Covert communications despite traffic data retention. In: Proc. of the Int'l Workshop on Security Protocols, 2008. 198−214.

[8] Ahsan K, Kundur D. Practical data hiding in TCP/IP. In: Proc. of the Workshop on Multimedia Security at ACM Multimedia 2002. Juan-les-Pins (on the French Riviera), 2002.

[9] Jones E, Moigne OL, Robert JM. IP traceback solutions based on time to live covert channel. In: Proc. of the IEEE Int'l Conf. on Networks, Vol.2. IEEE Xplore, 2004. 451−457.

[10] Qu H, Su P, Feng D. A typical noisy covert channel in the IP protocol. In: Proc. of the 2004 Int'l Carnahan Conf. on Security Technology. IEEE, 2004. 189−192.

[11] Hintz D. Covert channels in TCP and IP headers. Presentation at Defcon, 2002.

[12] Rowland CH. Covert channels in the TCP/IP protocol suite. First Monday, 1997,2(2):32−48.

[13] Rutkowska J. The implementation of passive covert channels in the linux kernel. In: Proc. of the Chaos Communication Congress. 2004.

[14] Feamster N, Balazinska M, Harfst G, *et al*. Infranet: Circumventing Web censorship and surveillance. In: Proc. of the Usenix Security Symp. 2002. 247−262.

[15] Bauer M. New covert channels in HTTP: Adding unwitting Web browsers to anonymity sets. In: Proc. of the Workshop on Privacy in the Electronic Society. 2003. 72−78.

[16] Bowyer L. Firewall bypass via protocol steganography. 2002. http://www.networkpenetration.com/protocolsteg.html

[17] Anonymous. DNS covert channels and bouncing techniques. 2005.

[18] Handel TG, Sandford Ii MT. Hiding data in the OSI network model. In: Proc. of the 1st Int'l Workshop on Information Hiding. Cambridge: DBLP, 1996. 23−38.

[19] Wolf M. Covert channels in LAN protocols. In: Proc. of the Workshop on Local Area Network Security (LANSEC'89), European Institute for System Security. DBLP, 1989. 91−101.

[20] Doğu TM, Ephremides A. Covert information transmission through the use of standard collision resolution algorithms. In: Proc. of the 3rd Int'l Workshop on Information Hiding (IH'99). Dresden: DBLP, 1999. 419−433.

[21] Cabuk S, Brodley CE, Shields C. IP covert timing channels: Design and detection. In: Proc. of the ACM Conf. on Computer and Communications Security (CCS 2004). Washington: DBLP, 2004. 178−187.

[22] Cabuk S. Network covert channels: Design, analysis, detection, and elimination [Ph.D. Thesis]. Purdue University, 2006.

[23] Shah G, Molina A, Blaze M. Keyboards and covert channels. In: Proc. of the Conf. on Usenix Security Symp. USENIX Association, 2006.

[24] Gianvecchio S, Wang H, Wijesekera D, *et al*. Model-Based covert timing channels: Automated modeling and evasion. In: Proc. of the Int'l Symp. on Recent Advances in Intrusion Detection (RAID 2008). Cambridge: DBLP, 2008. 211−230.

[25] Gianvecchio S, Wang H, Wijesekera D, *et al*. Model-Based covert timing channels: Automated modeling and evasion. In: Proc. of the Int'l Symp. on Recent Advances in Intrusion Detection (RAID 2008). Cambridge: DBLP, 2008. 211−230.

[26] Wu J, Ding L, Wu Y, *et al*. C2 Detector: A covert channel detection framework in cloud computing. Security & Communication Networks, 2014,7(3):544−557.

[27] Wei SQ, Yang W, Shen Y. A covert communication method based on reliable packet sorting. Journal of Chinese Computer Systems, 2016,37(1):124−128 (in Chinese with English abstract).

[28] Luo X, Chan EWW, Chang RKC. Cloak: A ten-fold way for reliable covert communications. Computer Security—Esorics, 2007, 4734:283−298.

[29] Shannon C. A mathematical thoery of communication. Bell System Technical Journal, 1948.

[30] Gianvecchio S, Wang H. Detecting covert timing channels: An entropy-based approach. In: Proc. of the ACM Conf. on Computer & Communications Security. 2007. 307−316.

[31] Xu XD, Wang CA, Zhu SR. Covert channel detection in ICMP payload based on information entropy SVM. Journal of Computer Applications, 2009,29(7):1796−1798 (in Chinese with English abstract).

[32] Sohn T, Seo JT, Moon J. A study on the covert channel detection of TCP/IP header using support vector machine. In: Proc. of the Int'l Conf. on Information and Communications Security (ICICS 2003). Huhehaote: DBLP, 2003. 313−324.

[33] Wu CW, Sun R, Luo M. Detection of telnet covert channel based on SVM. Information Security and Communications Privacy, 2012,(9):97−98 (in Chinese with English abstract).

[34] Tumoian E, Anikeev M. Detecting NUSHU covert channels using neural networks. 2005. http://www.ouah.org/neural_networks_vs_NUSHU.pdf

[35] Tumoian E, Anikeev M. Network based detection of passive covert channels in TCP/IP. In: Proc. of the IEEE Conf. on Local Computer Networks. Anniversary: IEEE Xplore, 2005. 802−809.

[36] Tang ZG, Li HZ, Zhong MQ, Zhang J. Heuristic detection model of covert channel based on quantum neural network. Journal of Computer Research and Development, 2012,29(8):3033−3035 (in Chinese with English abstract).

[37] Qiu GH. A detection method for cloud platform covert channels based on hybrid indicators of entropy and standard deviation. Computer Applications and Software, 2013,(9):200−204 (in Chinese with English abstract).

附中文参考文献:

[3] 王永吉,吴敬征,曾海涛,丁丽萍,廖晓峰.隐蔽信道研究.软件学报,2010,21(9):2262−2288. http://www.jos.org.cn/1000-9825/3880.htm [doi: 10.3724/SP.J.1001.2010.03880]

[27] 魏三强,杨威,沈瑶.一种基于可靠包排序的隐秘通信方法.小型微型计算机系统,2016,37(1):124−128.

[31] 徐晓东,王传安,朱士瑞.基于信息熵 SVM 的 ICMP 负载隐蔽通道检测.计算机应用,2009,29(7):1796−1798.

[33] 吴传伟,孙瑞,罗敏.基于 SVM 的 Telnet 隐蔽信道检测.信息安全与通信保密,2012,(9):97−98.

[36] 唐彰国,李焕洲,钟明全,张健.基于量子神经网络的启发式网络隐蔽信道检测模型.计算机应用研究,2012,29(8):3033−3035.

[37] 邱桂华.一种基于熵率和标准差混合指标的云平台隐蔽信道检测方法.计算机应用与软件,2013,(9):200−204.

张宇飞(1992−),男,山西吕梁人,工程师,主要研究领域为信息安全,隐信道检测.

沈瑶(1989−),男,博士,高级工程师,主要研究领域为信息安全,隐私保护.

杨威(1978−),男,博士,副教授,主要研究领域为信息安全,大数据与云计算,量子信息处理.

肖奕汉(1992−),男,工程师,主要研究领域为信息安全,大数据,云计算.

黄刘生(1957−),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为网络(无线传感网,物联网,车联网),信息安全(信息隐藏,安全多方计算,量子信息安全),大数据与云计算(大数据分析挖掘,大数据隐私保护,云安全).