基于应用识别的 P2P 蠕虫检测

夏春和 石昀平 李肖坚

(北京航空航天大学 网络技术北京市重点实验室, 北京 100083)

摘 要: 对等网中基于漏洞传播的 P2P 蠕虫是严重的安全威胁. 根据 P2P 蠕虫的传播特点,提出了一种 P2P 蠕虫检测方法 PWD (P2P Worm Detection). PWD 主要由基于应用识别技术的预处理和基于未知蠕虫检测技术的 P2P 蠕虫检测 2 部分组成,改进了干扰流量的识别和过滤规则,提出了 P2P 蠕虫检测规则,并引进博弈论的研究方法讨论了检测周期的选取问题. 仿真结果和局域网环境下的实验结果都表明,PWD 是检测 P2P 蠕虫和遏制其传播的有效方法.

关 键 词: P2P 蠕虫; 蠕虫检测; 应用识别

中图分类号: TP 393.08

文献标识码: A **文 章 编 号**: 1001-5965(2006)08-0998-05

P2P worm detection based on traffic classification and application identification

Xia Chunhe Shi Yunping Li Xiaojian

(Key Laboratory of Beijing Network Technology, Beijing University of Aeronautics and Astronautics, Beijing 100083, China)

Abstract: P2P worm exploits common vulnerabilities in Peer-to-Peer networks. It is a severe security threat. A P2P worm detection method was presented, which called P2P worm detection(PWD) based on the worm's propagation characteristics. PWD consists of a preprocess procedure which is based on application identification technology and a P2P worm detection procedure which is based on unknown worm detection technology. Improved heuristics was also advanced to identify and disturbing traffic was eliminated as well as heuristics to detect P2P worm. The selection of detection period was discussed by applying methodologies of game theory. Simulation result and LAN-scope experimental result both indicate that PWD is an effective method to detect and block P2P worm.

Key words: P2P worm; worm detection; application identification

目前互联网上使用的 P2P 软件缺乏统一规范,随意性、多样性强,无法保证这些软件和运行这些软件的系统在程序上和管理上不存在安全漏洞. P2P 蠕虫发掘系统应用漏洞,利用对等网的交互性质自主扩散[1],是广大互联网和 P2P 用户的重要威胁,有必要在该蠕虫的检测方面进行研究和探索.

本文基于应用识别技术和未知蠕虫的检测技术,提出了针对 P2P 蠕虫的检测方法 PWD(P2P Worm Detection). 仿真结果和真实环境下的检测实验结果均表明 PWD 是检测 P2P 蠕虫的有效方法,是防止 P2P 蠕虫大规模爆发的一道屏障.

1 相关研究

1.1 P2P 蠕虫及其传播特征

Staniford[2] 曾指出 P2P 系统具有非常适合蠕虫传播的特性. Jayanthkumar[3] 提出了发掘 P2P 系统应用漏洞、收集被感染机器的 P2P 邻居构造打击列表进行传播的蠕虫. 这种蠕虫比基于文件共享策略传播的蠕虫攻击性更强,其传播规则是

收稿日期: 2005-09-21

基金项目: 航空基金资助项目(03F51060); 北京市教委共建项目建设计划基金资助项目(SYS100060412); 国防基础科研资助项目

作者简介: 夏春和(1965-),男,江苏海安人,教授,xch@buaa.edu.cn.

Staniford 的攻击列表设想在对等网中的推广. Zhou L 等[1]指出利用覆盖网逻辑拓扑结构的蠕虫传播将省去扫描过程,令传播更隐蔽、攻击更精确高效,使目前绝大多数针对扫描型蠕虫的防御机制失效.

基于深入研究,本文建立了结构化对等网中 P2P 蠕虫的传播模型[4],该模型对其传播特征的准确刻画,为本文的检测处理周期方面提供了重要的参考价值.

1.2　未知蠕虫检测方法

未知蠕虫检测是基于自动提取特征的蠕虫检测方法,包括 Honeycomb[5], Autograph[6]和 EarlyBird[7]等. Honeycomb 在蜜罐基础上加入自动提取特征技术,但具有基于主机的性质,无法克服蜜罐的缺陷. Autograph 是一种脱线系统,其内容流行度计算依赖于对携带可疑扫描行为的流的识别和预过滤,该技术无法检测 Email 蠕虫、基于 UDP 的蠕虫和 P2P 蠕虫等非扫描蠕虫. EarlyBird 对所有内容基于流聚集进行指纹计算,缺少分流、分块等预处理,使其在 P2P 环境下的计算量猛增;EarlyBird 检测缺乏削弱或消除 P2P 流量干扰的机制,使得检测的有效性和可靠性难以保证.

1.3　应用识别技术

传统的应用识别技术依赖于对公认端口的识别[8],但不适应 P2P、游戏和流媒体等新型网络应用,于是出现了基于流的应用识别技术[9],利用流重组确定网络流量的应用类型.

P2P 应用识别是 P2P 蠕虫检测涉及的重要技术. 使用 P2P crawler 采集 P2P 信号流量[10]的方法扩展性差且代价昂贵,分析通过 ISP 主干流信息的方法[11]无法反映覆盖网的结构特征. Karagiannis T[12]提出的 P2P 识别技术采用无负载识别的预处理机制,识别出大量已知网络行为并滤去背景流量,结合端口匹配、负载特征码检查和传输层动态综合分析识别 P2P.

2　P2P 蠕虫检测方法 PWD

2.1　PWD 处理流程

在给出 P2P 蠕虫及相关干扰流量的识别法则的基础上,提出 P2P 蠕虫的实时检测和动态提取特征方法 PWD(见图 1).

PWD 对流量数据的处理分为 4 步:①将原始流量通过一般流量过滤器滤除一般背景流量(例如 DNS,NetBios,Arp 等). ②利用干扰流量检测法则依次滤除 Web/HTTP,Game 和 P2P 流量,同时生成 P2P 节点可能集. ③对剩余流量进行 P2P 蠕虫特征匹配检查,检测 P2P 蠕虫特征库中已存在的蠕虫. ④采用 P2P 蠕虫检测法则检查剩余流量,检测未知 P2P 蠕虫进而更新 P2P 蠕虫特征库.

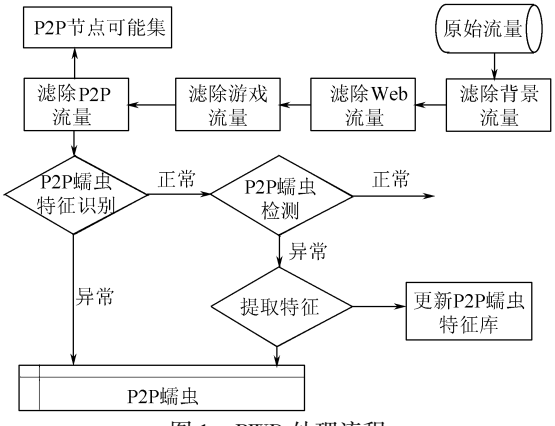


图 1　PWD 处理流程

2.2　应用识别与干扰流量滤除

P2P 蠕虫检测需要滤除 Web/HTTP、游戏和 P2P 3 种应用流量,这是因为越来越多的 P2P 应用为躲避防火墙的过滤开始复用 Web/HTTP 端口;其次,游戏和 P2P 应用流量与恶意代码传播流量具有一定的共同点:含有流向多个不同地址和端口且载有相同字节数目或数据包数目的流,容易干扰 P2P 蠕虫检测. Karagiannis 提出了识别上述应用的技术,但对背景流量的处理过于繁复,准确度不高. 由于背景流量(DNS,ICMP,多播,Netbios,ARP 等)特征明显,相应的处理工具也有很多,为简化处理过程,基于 Ethereal 予以滤除.

在识别 Web/HTTP 应用流量时,采用 Karagiannis 的检测法则[12]:若客户端流量均流向 $\{W, 80\}$ 对,且端口总数与 IP 地址总数比率较高,则认为这些流量基于 Web 应用.

在识别游戏应用时,Karagiannis 的法则[12]主要基于如下观点:①游戏流量使用小的、周期性的 UDP 包. ②游戏的主要流量是客户端和服务器间周期性的实时动作和调整信息,服务器向客户端周期性广播全局状态信息. ③流量具有一定周期性,流会话长度以分钟为数量级. 然而通过对进出校园网的多种网络游戏流量的观察分析,发现游戏应用流量并非均采用 UDP 包进行传输,并且由于网络游戏种类繁多,不同游戏的应用层数据大小及广播周期的波动范围均难以掌握. 因此,采用服务器向客户端发送的全局状态调整信息广播作为检查法则:若检测周期内发现由某主机发送给多个主机的内容相似度高的数据包,客户端、服务器双方均使用大于 1 024 的唯一端口,双方使用

唯一协议类型,且客户端端口互异则判定为游戏流量.

在识别 P2P 应用时,Karagiannis 检测法则[12]包括:①利用已知的 P2P 特征信息. ②在一定时间内,将未使用已知 P2P 应用端口、且同时使用 TCP 和 UDP 的源-目的 IP 对认为是 P2P. ③在一定时间内,检查所有的源{srcIP, srcport}和目的{dstIP, dstport}对,认为互异的连接 IP 数等于互异的连接端口数的对是 P2P. 通过对进出校园网 P2P 流量的观察和分析,发现由于 P2P 应用的种类繁多,很多应用流量对端口和协议的使用很不规则,而 Karagiannis 对端口和协议的组合使用情况以及对源端口的不规则使用未给出细化的检测法则.

基于观察与分析,为 Karagiannis 的法则②增加“通讯一方使用端口唯一”这一条件,新法则可识别 Maze 等流量;另外,修改法则③为:检测周期内,检查所有的源{srcIP, srcport}和目的{dstIP, dstport}对,若互异的连接数与互异的端口数相等,使用协议相同,目的端口唯一,且源-目的端口均大于1024,则认为源-目的对是 P2P,新法则可识别 BT 等应用流量.

2.3 P2P 蠕虫检测与取征

P2P 蠕虫不需要扫描,仅利用对等网向多个 P2P 节点的特定端口传送蠕虫体即可实现高效传播. 基于这些特性可得到相应的检测法则:短时间内向多台主机发送内容相似的数据包,且使用协议、目的端口基本一致,目的主机多数运行 P2P 应用,认为这样的源主机在试图进行 P2P 蠕虫传播. 另一方面,在检测出 P2P 蠕虫传播节点的同时需要获取蠕虫指纹,加入蠕虫特征库,在之后的流量监测中可以直接通过特征匹配检测已知蠕虫的传播,提高检测效率.

基于流的预处理滤除了干扰流量,再进行指纹计算,弥补了 EearlyBird 的缺陷,在降低计算量的同时提高了检测可靠性. P2P 可能节点列表也为后续检测提供了方便.

2.4 检测周期的博弈分析

PWD 的每个检测周期由流量收集和数据处理2部分组成. 过大的流量易增大处理负荷;过小的流量易降低处理精度. 下面通过博弈分析确定检测周期 T.

不妨设蠕虫的传播方程为 $f(t)$(具体表达式见参考文献[4]),设攻击方在$[0,T)$周期内的某时刻 t 释放蠕虫,设 t_f 为流量收集完成时刻. 由于 PWD 的检测和抑制阶段至多相隔一个周期,所以蠕虫的有效传播时间至多为 $2T$,由 $f(t)$ 的单调递增特性知道,蠕虫感染的节点数不超过 $f(2T)$.

蠕虫的危害程度由感染规模决定,假设造成危害的感染节点个数阈值为 N,由 $f(t)$ 的单调性,不妨设 $N=f(t_0)$,也就是蠕虫在有效传播时间 t_0 内感染的节点个数为 N. 若 $f(2T)<N$,蠕虫不会造成大规模危害,所以假定 $0<N<f(2T)$ 总成立.

设一个周期内攻击方的平均开销为 C_a,防御方的检测开销为 C_d,设攻击发生概率为 p. 若攻击未发生,攻击方获得效用为零,防御方效用为 V_d^+. 若攻击发生且成功,攻击方获得效用 V_a^+,防御方效用为 V_d^-;若攻击发生但失败,攻击方获得效用 V_a^-,防御方效用为 V_d. 其中,$V_a^+>0>V_a^-$,且 $V_d^+>V_d>0>V_d^-$.

攻防效用由感染规模、网络性质、攻击发生时间等多个因素决定,是攻防双方不易控制或作用机制非常复杂的函数. 为简化,视其为常量.

令 λ 为攻击成功概率,于是攻防双方的效用方程为

$$E(U_d)=(1-p)V_d^+ + p[\lambda V_d^- + (1-\lambda)V_d] - C_d \quad (1)$$

$$E(U_a)=p[\lambda V_a^+ + (1-\lambda)V_a^-] - C_a \quad (2)$$

由于 t_0 与 T,t_f 之间的关系影响到 λ 的取值,需要分情况进行讨论. 当 $T<t_0<2T$ 时,若攻击发起时刻 $t<t_f$,由于 $f(T-t)<f(T)<f(t_0)=N$,攻击失败;若攻击发起时刻 $t\in[t_f,T)$,当 $t_f\leqslant t<2T-t_0$ 时,有 $f(2T-t)>f(t_0)=N$,攻击成功;当 $2T-t_0<t<T$ 时,有 $f(2T-t)<f(t_0)=N$,攻击失败. 综上,只有攻击发起时刻 t 落入$[t_f,2T-t_0)$区间攻击才能够成功,于是 $\lambda=1-(t_0+t_f)/2T$. 相应地,攻击失败概率为 $(t_0+t_f)/2T$. 同理,当 $T-t_f<t_0\leqslant T$ 时,$\lambda=[(T-t_0)+(T-t_f)]/T$,攻击失败概率为$[t_f-(T-t_0)]/T$. 当 $0<t_0\leqslant T-t_f$ 时,攻击必定成功,即 $\lambda\equiv 1$.

由上可见,效用方程中的 λ 由攻防双方共同控制. λ 中的 T 和 t_f 由防御方控制,反应了检测技术水平. 由于 T 和 t_f 相互制约,可视为一个变量,为简化起见设 $t_f=\alpha T$,其中 $0<\alpha<1$. λ 中的 t_0 由攻击方控制,反应了蠕虫的攻击水平. 另外,由于攻击方决定是否发起攻击,所以攻击概率 p 由攻击方掌握.

为求得纳什均衡解,针对不同情况,分别对攻防双方的可控因素求偏导数. 当 $T<t_0<2T$ 时,得到

$$\frac{\partial E(U_d)}{\partial T}=p\left(V_d^- - V_d\right)\frac{t_0}{2T^2} \quad (3)$$

$$\frac{\partial E(U_a)}{\partial t_0}=p\left(-V_a^+ + V_a^-\right)\frac{1}{2T} \quad (4)$$

$$\frac{\partial E(U_a)}{\partial p} = \left[(1-\lambda)V_a^+ + \lambda V_a^-\right] \quad (5)$$

式(3)恒小于零,说明在攻击方取定 t_0 的基础上,防御方选取的检测周期 T 越小,获得的效用就越大. 为获得效用极值,防御方将选取边界值,即最小的检测周期 T.

式(4)恒小于零,说明在防御方取定 T 的基础上,攻击方蠕虫达到一定破坏程度的有效传播时间越短,获得的效用就越大. 为获得效用极值,攻击方将选取边界值即最短的有效传播时间.

对于式(5),若 $\lambda > V_a^-/(V_a^- - V_a^+)$ 则攻击效用随攻击概率的增大而增大,于是攻击概率必为 1;反之,若 $\lambda \leqslant V_a^-/(V_a^- - V_a^+)$,则攻击效用始终为非正值,攻击概率必为零.

当 $T - t_f < t_0 \leqslant T$ 时的分析结果与前一种情形相似. 当 $0 < t_0 \leqslant T - t_f$ 时,效用方程组退化为

$$E(U_d) = (1-p)V_d^+ + pV_d^- - C_d \quad (6)$$

$$E(U_a) = pV_a^+ - C_a \quad (7)$$

此时,由式(7)可知,攻击者将选择 $p=1$ 以求效用最大化.

t_0 与 T,t_f 之间的 3 种关系仅限于讨论和分析,攻防展开时真正成立哪一种关系,双方只能根据推测进行策略选择. 值得注意的是,不论哪种情况,防御方总是倾向于使用尽量小的 T,攻击方总是倾向于使用尽量小的 t_0,而攻击概率的选择则充满随机性. 这也可理解为:攻击方选用的 t_0 越小,蠕虫达到一定破坏程度所需的时间越短,攻击效果可能越好;防御方的检测周期越小,可能在较短时间内较准确地发现蠕虫的存在并尽快加以遏制. 基于该结果在检测实验中将选取尽可能小的检测周期.

3　仿真实验与真实环境下的实验

3.1　仿真实验

在 NS-2 环境下,仿真建立了由 50 个节点组成的网络,生成节点间 Web/HTTP, Game, P2P 3 种应用和 P2P 蠕虫的流量,执行 PWD 检测算法的监控节点运行图 1 的处理流程.

限于篇幅,现介绍基于 Chord[13] 的仿真实验. 覆盖网由 16 个节点组成. 进行感染用的蠕虫体系特定内容的字符串,传播时由节点向自身的逻辑拓扑邻居传播,当邻居收到蠕虫包以后,通过改变自身颜色表示被感染,经过一定延时后向其逻辑拓扑邻居传播蠕虫,直到所有节点均被感染. 传播时向邻居发包的源端口随机,目的端口固定.

图 2 展示了仿真实验结果. 初始时刻,圆形空心节点为正常节点,感染源(方形节点)开始传播 P2P 蠕虫. 第一轮感染发起后(此时已感染节点增为 5 个,即图中的方形节点),PWD 检测出蠕虫传播的数据包,随即生成蠕虫特征,后续的特征匹配检测拦截了网络中蠕虫包的进一步传输,图中 6 个圆形实心节点代表了被拦截蠕虫包的目的端,即被保护了的节点. 由于这些圆形实心节点未被感染,当然也未给它们的逻辑拓扑邻居发送感染包,于是那些节点仍保持未感染状态(圆形空心). PWD 的存在,使得蠕虫无法继续扩散,16 个 P2P 节点仅感染了 5 个,感染率为 31.25%.

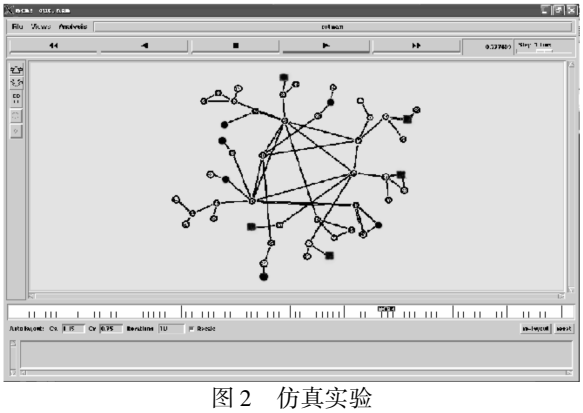


图 2　仿真实验

将 Chord 节点增加为 32 个,仿真结果与上例类似,PWD 在蠕虫发起第一轮感染后开始遏制其传播,仿真结束时 32 个节点感染了 9 个,感染率为 28.1%. 多组实验表明基于 Chord 的 PWD 检测将感染率控制在 $\frac{n+1}{2^n}$.

3.2　真实环境下的检测实验

选取 7 d 的不同时段在校园网网关处进行 PWD 检测实验,流量截取时间长度为 10 s,每组实验编号为 1 ~ 7. 表 1 列出了实验截取的流量信息,包括包数目、流数目和预处理后的应用信息等.

表 1　数据采集信息

编号	日期(2005)	星期	起始时间	包总数	滤后流数	Web 流数	游戏流数	P2P 节点数
1	5/22	日	14:20	15 687	829	31	1	61
2	5/24	二	16:20	21 998	234	13	0	22
3	5/26	四	15:20	48 990	395	12	2	84
4	5/27	五	10:29	19 925	907	29	1	124
5	5/28	六	11:47	17 877	451	21	2	71
6	5/29	日	10:40	28 731	409	10	4	65
7	5/30	一	10:27	85 914	289	17	2	48

在 P2P 蠕虫的传播实验中,采用特定字符串代表蠕虫体,通过向 P2P 节点的特定端口发送“蠕虫”包“感染”对方主机. 为检验 PWD 的有效性,准备了多组“蠕虫体”,并间隔一段时间反复发送. PWD 的主要工作是捕获这些包提取特征,

并对后续的包基于特征库进行特征匹配. 根据检测周期大小将蠕虫的持续攻击划分为不同轮次,首轮攻击与 PWD 的流量分析、检测和取征过程相对应,在后几轮攻击过程中 PWD 开始启用特征匹配拦截蠕虫.

图 3 中,PWD 检测出的节点比感染节点多一个节点,通过比较感染节点列表发现这个节点恰恰是感染源,由于感染源的传播包在首轮攻击中尚未被拦截,其感染对象被 PWD 列为感染节点. 在次轮攻击过程中,由于基于特征的包拦截,蠕虫的传播包基本被过滤掉,所以感染节点数和检测节点数基本不变(图 4),PWD 保护了首轮感染节点欲传播的大量节点. 从图 3 和图 4 的实验结果可以看出,PWD 能够在首轮攻击中准确捕获蠕虫感染源与蠕虫特征,并在后续攻击轮次中根据特征截获和过滤蠕虫包,控制蠕虫的传播.

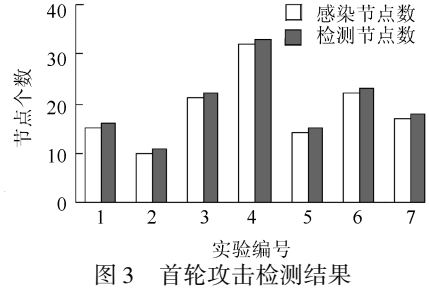


图 3 首轮攻击检测结果

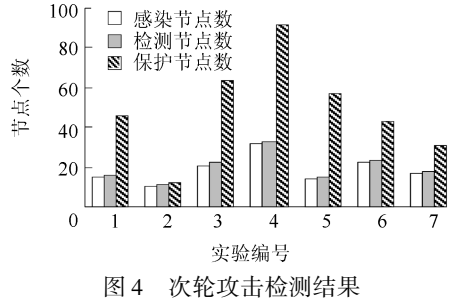


图 4 次轮攻击检测结果

与仿真环境相比,真实环境下的实验规模更大,网络流量种类更丰富,实验结果更有力地显示了 PWD 检测的有效性.

4 结束语

P2P 蠕虫是对等网的重要威胁. 本文基于对 P2P 蠕虫的深入研究和对该蠕虫传播特点的掌握,提出通过基于流的预处理、过滤干扰流量,利用法则检测未知蠕虫,再进行指纹计算及特征匹配的检测方法 PWD,并进行了仿真实验和校园网范围的传播实验. 2 种实验结果均表明,PWD 是检测和控制 P2P 蠕虫的有效方法.

参考文献 (References)

[1] Zhou L, Zhang L, McSherry F, et al. A first look at Peer-to-Peer worms: threats and defenses[C]//Proceedings of the Peer-to-Peer Systems 4th International Workshop. Ithaca: [s. n.], 2005:24-25

[2] Staniford S, Paxson V, Weaver N. How to own the internet in your spare time[C]//Proceedings of the 11th VSENZX Security Symposium. San Francisco: [s. n.], 2002:149-167

[3] Kannan J. Implications of Peer-to-Peer networks on worm attacks and defenses [EB/OL]. California: CS294-4 Project, 2003 [2003-09-01]. http://www. cs. berkeley. edu/ ~ kubitron/courses/cs294-4-F03/projects/karthik _ jayanth. pdf

[4] Xia Chunhe, Shi Yunping, Li Xiaojian. Research on propagation models of P2P worm in structured Peer-to-Peer networks [J]. Chinese Journal of Computer, 2006, 6:952-959

[5] Kreibich C, Crowcroft J. Honeycomb in creating intrusion detection signatures using honeypots[C]//Proceedings of the USENIX/ACM Workshop on Hot Topics in Networking. Cambridge: ACM Press, 2003

[6] Kim K, Karp B. Autograph: toward automated distributed worm signature detection[C]//Proceedings of the USENIX Security Symposium. California: Usenix Association, 2004:271-286

[7] Sumeet S, Cristian E, George V, et al. Automated worm fingerprinting[C]//Proc of the 6th USENIX Symposium on Operating Systems Design and Implementation (OSDI). [S. l.]: Usenix Association, 2004:45-60

[8] Fraleigh C, Moon S, Lyles B, et al. Packet-level traffic measurements from the sprint IP backbone[J]. IEEE Network, 2003, 17(6): 6-16

[9] Choi T, Kim C, Yoon S, et al. Content-aware internet application traffic measurement and analysis[C]// IEEE/IFIP Network Operations & Management Symposium. New York: IEEE Communications Society, 2004

[10] Krishnamurthy B, Wang J, Xie Y. Early measurements of a cluster-based architecture for P2P systems[C]//Proceedings of ACM Sigcomm Internet Measurement Workshop. New York: ACM Press, 2001

[11] Sen S, Wang J. Analyzing Peer-to-Peer traffic across large networks[J]. ACM/IEEE Transactions on Networking, 2004, 12(2):219-232

[12] Karagiannis T, Broido A, Faloutsos M. Transport layer identification of p2p traffic[C]//ACM SIGCOMM/USENIX Internet Measurement Conference. Italy: ACM SIGCOMM, 2004

[13] Stoica I, Morris R, Karger D, et al. Chord: a scalable Peer-to-Peer lookup service for Internet applications[C]//Proceedings of the ACM SIGCOMM 2001 Conference. California: ACM SIGCOMM, 2001: 149-160

(责任编辑:张 嵘)