计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2014.20131106
51(2):311-324,2014

基于社区的移动互联网混合蠕虫双向反馈遏制系统

杨海陆　张健沛　杨　静
(哈尔滨工程大学计算机科学与技术学院　哈尔滨　150001)
(yanghailu@hrbeu.edu.cn)

Community-Based Bidirectional Feedback System for Hybrid Worm Containment in Mobile Internet

Yang Hailu, Zhang Jianpei, and Yang Jing
(*College of Computer Science and Technology, Harbin Engineering University, Harbin* 150001)

Abstract　Aiming at the problem that the existing worm containment methods can't reply the mobile Internet worm attack which mixes long-range and short-range attack, this paper proposes a mobile Internet mixed worm bidirectional feedback and containment system based on community. The system consists of SIN (social information networks) containment unit and GIN (geographic information networks) feedback unit. The SIN containment unit is a type of online community quarantine strategy, which contains worms within the community by identifying the access nodes between communities and designing the corresponding worm label delivery algorithm. The GIN feedback unit collects the users' short range communication records, GPS location data and the historical security information committed by SIN to realize the trust-assessment. Through feeding back the results to SIN containment unit, the GIN limits the next communication decisions of community internal nodes, accordingly reduces the spreading speed of worms inside the community and realizes the bi-directional loop between the SIN containment unit and GIN feedback unit. Simulation experiments have proved that the method proposed by this paper has feasibility and effectiveness.

Key words　mobile Internet; hybrid worm containment; worm modeling; community quarantine strategy; trust assessment

摘　要　针对现有的蠕虫遏制方案无法应对移动互联网长短距混合蠕虫攻击这一问题,提出一种基于社区的移动互联网混合蠕虫双向反馈遏制系统.该系统分为社会信息网络(social information networks,SIN)遏制单元和地理信息网络(geographic information networks,GIN)反馈单元2个子系统,SIN遏制单元采用一种在线式社区隔离策略,通过识别社区间的门禁节点并设计相应的蠕虫标签投送算法,将蠕虫遏制在社区内部;GIN反馈单元收集用户的短程通信记录、GPS位置数据以及来自SIN遏制单元提交的历史安全信息,实现对节点的信任性评估,通过将结果反馈到SIN遏制单元,限制社区内部节点的下一步通信决定,从而降低蠕虫在社区内部的传播速度,实现了SIN遏制单元和GIN反馈单元的双向循环.最后通过仿真实验验证了所提方法的可行性和有效性.

收稿日期:2013-07-30;**修回日期**:2013-10-09
基金项目:国家自然科学基金项目(61073041,61073043,61202274,61370083);教育部高等学校博士学科点专项科研基金项目(20112304110011,20122304110012)
通信作者:张健沛(zhangjianpei@hrbeu.edu.cn)

关键词 移动互联网;混合蠕虫遏制;蠕虫建模;社区隔离策略;信任性评估

中图法分类号 TP309.5

移动互联网作为促进智慧城市发展的驱动力之一,以智能手机为载体,利用信息和通信技术使城市生活更加便捷化.然而随着智能终端的不断普及以及移动业务的不断拓展,移动互联网为人们带来诸多便利的同时,受恶意代码侵扰的频率也在日益增加.蠕虫病毒作为一种广为传播的恶意代码,由于其传播速度快、传播范围广以及爆发的突然性,构成了目前对移动互联网最大的威胁.

移动互联网蠕虫的传播方式分为短程传播和远程传播.其中短程蠕虫主要通过蓝牙接口等扩散到临近的设备上;而远程蠕虫主要以社交服务、多媒体信息服务(multimedia messaging service, MMS)等不受地理因素影响的复杂系统为传播载体进行传播.短程蠕虫针对无线传感设备的系统脆弱性进行攻击,该方向的蠕虫遏制工作主要集中在文献[1-7],主体思想是切断与脆弱节点之间的链接,从而避免蠕虫的进一步传播.远程蠕虫在功能上十分接近 IM 蠕虫[8],其在传播时首先扫描用户的通信列表,然后选择联系较为紧密的用户作为感染对象.针对远程蠕虫的遏制工作主要集中在文献[9-12],其核心思想是对网络中具有较高感染风险的节点进行隔离.

随着蠕虫种类的不断变化以及传播手段的不断改进,蠕虫的传播方式不再局限于上述传播途径.新型蠕虫采用混合式传播手段,在扫描临近设备的同时对远程服务中的通信列表实施攻击.由于混合蠕虫的协同式感染策略,其传播速度和遏制难度要远胜于单质蠕虫.目前尚无有效的遏制方案应对移动互联网蠕虫的这种混合传播行为.归纳而言,混合蠕虫遏制面临的挑战主要有 3 点:1)能够同时遏制蠕虫的远程传播和短程传播;2)能够对动态网络进行实时性监控;3)在遏制蠕虫传播的同时不会对网络造成过多的带宽负担.

为解决上述问题,本文设计了一种以远程遏制为主导、以短程遏制为反馈的移动互联网混合蠕虫遏制系统(hybrid worm containment system, HWCS).首先提出一种混合蠕虫传播模型,通过对模型中的相关参数进行分析,识别出蠕虫传播的脆弱环节;然后提出一种新的在线式重叠社区识别方法,并基于社区识别结果,设计了一种基于节点倾向性判别的蠕虫标签投送策略,从而可以将蠕虫遏制在社区内部;最后根据节点的历史安全信息、短远程通信记录提出一种新的节点信任性评估函数,对信任性较低的节点拒绝通信请求,进而遏制了蠕虫在社区内部的传播速度.

1 相关工作

相比于 Internet 蠕虫[13]的受关注程度,目前仅有少量文献面向移动互联网蠕虫建模与遏制,其中比较有代表性的工作有:

1) 在短程蠕虫遏制方面,Su 等人[1]通过对不同位置用户使用蓝牙设备的接触时长进行统计分析,证实了蠕虫可以经由蓝牙设备在极短的时间内感染到所有的临近设备中.Yan 等人[2]提出一种较为详细的传播模型分析蓝牙蠕虫的动态传播行为.Zyba 等人[3]提出一种纯分布式短距蠕虫遏制方案,该方案能够侦测到蠕虫,并且以一种泛洪式手段将蠕虫签名传播到网络中,但由于该方法会对网络的正常通信产生影响,因此局限性较强.Yang 等人[4]提出了一种软件多样性方法应对传感器蠕虫攻击,并将蠕虫防御定义为用有限的软件版本最小化网络脆弱链接的问题,通过设计一种基于角色的图染色方案,蠕虫的传播速度可以被很好地控制.Mickens 和 Noble[5]对 Kephart-Whilte 模型进行扩展,提出一种概率队列模型,对移动互联网中基于短程无线端口进行传播的蠕虫进行建模.Miklas 等人[6]将移动互联网中手持设备持有者之间的社会关系分为朋友和陌生人,以此提高系统的性能和请求命中率,通过拒绝与陌生人之间的通信来缓解蠕虫的传播.与之类似,Li 等人[7]设计了一种节点脆弱性评估方法,以限制脆弱节点间的通信作为蠕虫遏制手段.

2) 在远程蠕虫遏制方面,Fleizach 等人[9]首先证实了移动互联网与 Internet 蠕虫在传播性能上的不同,然后利用移动设备的各种信息服务对蠕虫的传播进行评估.Meng 等人[10]对短信网络(short message networks, SMNs)的传输可靠性及短信服务(short message service, SMS)上的蠕虫传播问题作出详尽的分析并提出相应的评估方法.Bose 和 Hu 等人[11]设计了一种 MMS/SMS 蠕虫遏制策略,首先生成移动互联网节点的易感染列表,然后对排名靠前的节点进行速度控制与隔离.Zhu 等人[12]通过对 MMS 网络进行聚类,限制蠕虫的进一步传播,

但该方法需要预先给出聚类数 k 并且未能反映节点真实的群集关系,因此遏制性能较低.

以上这些方法均假设蠕虫只通过 MMS 服务或蓝牙设备进行扩散,而忽视了蠕虫的混合传播行为对遏制效果产生的影响.本文的研究正是从这一角度出发,综合考虑 SIN 和 GIN 的纠缠特性,从而遏制移动互联网蠕虫的混合传播行为.

2 蠕虫遏制系统及工作原理

2.1 移动互联网蠕虫传播模型

理想的蠕虫传播模型应该能够充分地反映蠕虫的传播行为,并发现蠕虫传播链中存在的薄弱环节,从而可以有针对性地设计相应的蠕虫遏制策略,以此降低蠕虫带来的危害.受到文献[14-15]的启发,本文对流行病学理论中的 SIR(susceptible infected recovered)模型进行扩展,设计了移动互联网蠕虫的混合传播模型(hybrid SIR, HSIR).对于 SIR 模型而言,人群的感染性保持 3 种状态:易感染、被感染和免疫.其中被感染过的人群数 $J(t)$ 等于具有感染性的人群数 $I(t)$ 与免疫人群数 $R(t)$ 之和.如果用 β 表示感染率,γ 表示治愈率,N 和 $S(t)$ 分别表示人群总数和易感染人群数,则 SIR 模型的微分方程表达式为

$$\begin{cases} \mathrm{d}J(t)/\mathrm{d}t=\beta J(t)[N-J(t)]; \\ \mathrm{d}R(t)/\mathrm{d}t=\gamma I(t); \\ J(t)=N-S(t). \end{cases} \tag{1}$$

对于 HSIR 模型,移动设备之间的恶意通信被建模为感染者与易感染者之间的行为接触.移动设备的总量对应于人群总数 N,假设移动设备的分布较为均匀并且呈现一种稳定状态,则 GIN 上的节点密度可近似用 ρ 加以表示.本文认为节点之间所有的异构交互行为均来源于 SIN 和 GIN,因此可以推断出时刻 t 系统感染总数 $I(t)=I_{\mathrm{SIN}}(t)+I_{\mathrm{GIN}}(t)$,这里 $I_{\mathrm{SIN}}(t)$ 和 $I_{\mathrm{GIN}}(t)$ 分别为通过 SIN 和 GIN 得以感染的节点数.如果 $S(t)$ 为时刻 t 易感染节点的总量,则有:

$$I_{\mathrm{SIN}}(t)+I_{\mathrm{GIN}}(t)+S(t)=N, \tag{2}$$

同时满足:

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t}=\frac{\mathrm{d}I_{\mathrm{SIN}}(t)}{\mathrm{d}t}+\frac{\mathrm{d}I_{\mathrm{GIN}}(t)}{\mathrm{d}t}. \tag{3}$$

用 β_{SIN} 和 β_{GIN} 分别表示蠕虫在 SIN 和 GIN 上的感染率,η_{SIN} 和 η_{GIN} 表示 SIN 和 GIN 上节点的平均链接程度.需要说明的是,平均链接程度在 SIN 上代表的是网络的平均节点度,而在 GIN 上则用来度量移动设备物理覆盖半径 R 所形成感染圈内部的平均节点数.为了不失一般性,我们假设在扩散过程的初始阶段蠕虫首先基于 SIN 对移动设备进行感染,如基于 MMS、社交服务等;然后随着感染过程的不断深入,蠕虫在 GIN 上借由蓝牙等设备进行二次传播.这意味着 $I(0)=I_{\mathrm{SIN}}(0)$,并且 $I_{\mathrm{GIN}}(0)=0$.图 1 描述了这种混合的传播行为:

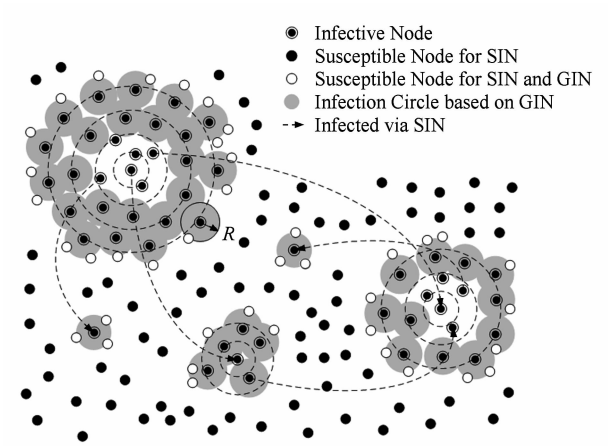


Fig. 1 The hybrid spreading behavior of mobile internet worm.

图 1 移动互联网蠕虫混合传播行为

2.1.1 SIN 上的蠕虫传播动力学分析

SIN 上的蠕虫传播动力学与 IM 蠕虫的传播特性较为接近.首先,SIN 蠕虫通过联系人列表进行传播,与主动探测蠕虫所采用的 Hit-list 传播机制类似,二者均拥有明确的攻击目标,并且传播速度很快;其次,SIN 蠕虫和 IM 蠕虫均不必大量发送探测数据包来实施对攻击目标的探测,导致对这类蠕虫的检测变得十分困难.在图 1 中,节点之间的社会交互特性使得 2 个距离较远的节点也可以极快的速度进行通信.如果用 β_{SIN} 表示节点接收蠕虫后成功激活的概率,则单一节点附近的易感染节点数近似等于 $\eta_{\mathrm{SIN}}\beta_{\mathrm{SIN}}$,可见蠕虫在经由 SIN 传播时的易感染节点数为

$$S'(t)=S(t)\frac{\eta_{\mathrm{SIN}}\beta_{\mathrm{SIN}}}{N}I(t). \tag{4}$$

基于式(4),SIN 节点的感染动力学微分方程为

$$\frac{\mathrm{d}I_{\mathrm{SIN}}(t)}{\mathrm{d}t}=\beta_{\mathrm{SIN}}^2\frac{\eta_{\mathrm{SIN}}S(t)}{N}I^2(t)-\gamma' I(t), \tag{5}$$

这里 γ' 为移动设备的修补率,通常经由对设备投送补丁得以实现.

2.1.2 GIN 上的蠕虫传播动力学分析

当蠕虫借由 GIN 进行扩散时,感染节点会扫描

感染半径 R 范围内的所有节点. 由于新时刻蠕虫的扩散行为不会对波纹内部节点产生二次感染(免疫或未痊愈),因此实际上每次能够对网络进行 GIN 感染的节点仅存在于波纹的边缘上. 假设在时刻 t 某波纹的感染半径增加至 $r(t)$,令 $F'(t)$ 和 $F(t)$ 分别表示波纹边缘感染节点数和波纹内部感染节点数,则边缘节点的感染动力学特性可用式(6)加以描述:

$$F'(t)=F(t)-\rho\pi(r(t)-R)^2, \tag{6}$$

这里 $r(t)-R$ 为蠕虫最后 1 次扩散前波纹的感染半径,$\rho\pi R^2$ 代表了节点的平均邻居数 η_{GIN}. 由于波纹中心节点的平均邻居数即为波纹的内部节点,因此有 $F(t)=\rho\pi r^2(t)$. 对式(6)进行化简,可得:

$$F'(t)=2R\sqrt{\rho\pi}\sqrt{F(t)}-\eta_{GIN}. \tag{7}$$

为了不失一般性,我们假设某波纹在时刻 a 受到 SIN 蠕虫感染,在持续 b 时间单位之后达到当前感染状态 $F(a+b)$,则感染波纹在时刻 $a+b$ 的感染增量为

$$F'(a,b)=\beta_{GIN}\frac{\eta_{GIN}\left(2R\sqrt{\rho\pi}\sqrt{F(a,b)}-\eta_{GIN}\right)^2}{2N}S(a+b). \tag{8}$$

化简后可得:

$$F'(a,b)=\beta_{GIN}\frac{\eta_{GIN}^2\left(c\sqrt{F(a,b)}-1\right)^2}{2N}S(a+b), \tag{9}$$

在式(9)中,$2N$ 表示每个边缘节点在分布平均的情况下会感染其一半的邻居节点(外部节点),β_{GIN} 为蠕虫经由蓝牙等物理设备成功转发并激活的概率,$c=2/R(\rho\pi)^{1/2}$ 为常参数,分数式代表感染波纹外部的易感染节点占所有易感染节点的比例. 在前文中 $I_{GIN}(0)=0$,这说明蠕虫的地域性感染实际上受到 I_{SIN} 的支配. 如图 1 中,每个扩散波纹的中心都存在 1 个经由 SIN 感染的起始节点,因此时刻 t 所有波纹新增感染节点的动力学方程为

$$\frac{\mathrm{d}I_{GIN}(t)}{\mathrm{d}t}=\int_0^t I'_{SIN}(\tau)F'(\tau,t-\tau)\mathrm{d}\tau, \tag{10}$$

这表明在时刻 τ 共有 $I'_{SIN}(\tau)\mathrm{d}\tau$ 个经由 SIN 感染的波纹存在,并且在时刻 t 每个波纹的平均感染节点增量为 $F'(\tau,t-\tau)$.

图 2 给出了蠕虫在 SIN,GIN 和混合模型上传播性能的仿真结果,其中节点密度 $\rho_{GIN}=0.75$,蠕虫激活概率 $\beta_{SIN}=\beta_{GIN}=0.1$,平均节点度 $\eta_{GIN}=\eta_{SIN}=6$. 如图 2 所示,相比于单独依托 SIN 或 GIN 的传播模式,由于混合模型的协同感染特性,蠕虫在混合网络上具有最快的感染速度. 可见,为了有效地遏制移动互联网蠕虫,所设计的遏制策略必须全面考虑 SIN 和 GIN 上的双重传播因素,独立的遏制方案无法显著地降低蠕虫的传播速度.

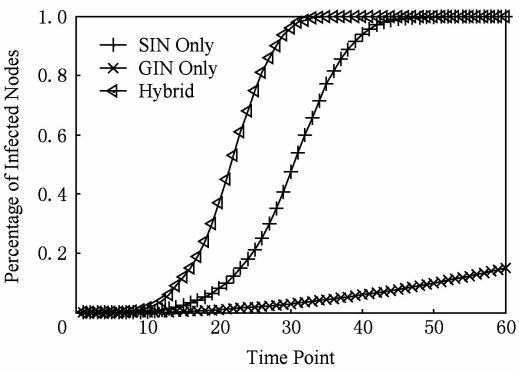
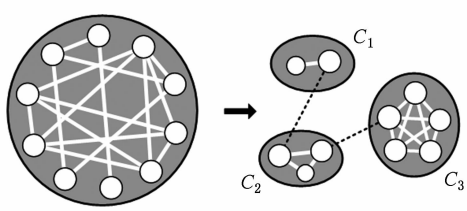


Fig. 2 Analytical results of propagation dynamics of spreading via SIN, GIN and hybrid model.

图 2 基于 SIN,GIN 和混合模型的传播动力学分析

2.2 蠕虫遏制系统

移动互联网蠕虫的传播行为十分接近用户的正常使用习惯(仅对联系人列表发送恶意程序),进而难以被快速发现,这就要求当蠕虫爆发时所设计的蠕虫遏制策略必须是全自动行为,因为蠕虫的传播速度过快以至于超出了人们的反应时间. 进一步对蠕虫的传播动力学方程进行分析可以发现,式(5)中决定 SIN 蠕虫传播速度的关键参数为激活率 β_{SIN}、平均节点度 η_{SIN} 以及设备修补率 γ'. 由于蠕虫经由联系人列表进行传播,因此当接收者发现文件来源是比较熟悉的人以后,会没有任何防备地打开文件从而将蠕虫激活,以至于在 SIN 中 β_{SIN} 近似等于 1. 对于修补率 γ',设备供应商在探测蠕虫以及编写补丁上要消耗一定的时间,难以立刻产生遏制效果. 可见,一种降低蠕虫传播能力的可行方案是降低网络的 η_{SIN} 值. 为此本文提出一种基于社区的 SIN 蠕虫遏制方案. 如图 3 所示,图 3(a)中网络的平均节点度为 $\eta_{SIN}=2.7$,在对网络进行社区识别后可得社区划分结果,如图 3(b)所示. 在图 3(b)中,如果各社区以节点



(a) Original network (b) After community detection

Fig. 3 Community structure in social information networks.

图 3 社会信息网络中的社区结构

的形式存在,则此时网络的平均节点度为 $\eta_{SIN}(C)=1.3$,约为原网络的 50%.可见,社区之间蠕虫的传播性能较低,进而蠕虫会被遏制在社区内部.

在式(9)和式(10)中,影响 GIN 蠕虫传播性能的关键参数是 β_{GIN} 和 η_{GIN}.与 SIN 这种非地域性网络模式相比,GIN 中 η_{GIN} 值代表了单位地域内移动设备的总量,由于人们在真实世界中的运动性通常很低,因此该值通常难以改变.为了能够降低 GIN 中的蠕虫激活率 β_{GIN},本文的做法是对 GIN 中节点的可信性进行评估,对信任度较低的节点采取拒绝通信.GIN 节点的信任程度主要基于 3 个参数:1)节点的地理位置信息.地理距离较远的节点实际上难以受到地域性传播的影响,对于这部分节点蠕虫只会经由 SIN 传播.2)节点的历史安全记录.该部分反映了用户的安全意识,长期受蠕虫侵扰的移动设备由于自身安全性较低,因此其请求的通信行为威胁性较大.3)移动设备之间的链接频率.经常使用蓝牙设备进行通信传输的设备在蠕虫爆发后具有潜在的传播威胁,由于设备之间长期的无障碍通信模式,蠕虫可以轻易地扩散到与感染设备配对的其他设备中.

根据上述分析,本文设计了一种面向移动互联网混合蠕虫的双向反馈遏制系统.如图 4 所示,系统首先对 SIN 中的节点进行在线式社区识别,由于蠕虫爆发后没有多余的时间重新挖掘 SIN 的社区结构,因此所设计的动态社区挖掘算法要能够实时地对前一时刻的社区结构进行更新,以满足网络的演变需要.一旦检测到蠕虫存在,系统首先生成蠕虫的签名,然后通过门禁节点检测单元,将签名投送至各门禁节点,以此遏制蠕虫在社区间的传播.这里的门禁节点分为 2 部分:首先是影响社区间通信的阀节点,例如图 3(b)中 3 个社区之间虚线链接的端点;其次是 GIN 节点可信性评估单元反馈的高危通信节点,这些节点可能分属于不同社区,如果不加以限制,蠕虫极可能通过 GIN 传播到不同的社区中,进而无法遏制蠕虫的传播.由于经由门禁节点的信息具有极快的传播效率,因此该单元同样作为供应商投送补丁的接口所在.节点可信性评估的输入参数来源于信息采集单元,这里移动设备的链接频率和地理位置信息可由供应商直接提供,历史安全记录源自 SIN 遏制单元的反馈,并分为正常通信和感染性通信 2 种,用参数 a 和 b 统计 2 种通信的总次数,并以此为输入参数提交到评估单元中.在接下来的章节中,本文将分别给出遏制系统内部单元的具体设计方案,其中包括 SIN 社区识别算法、动态社区校准算法、节点可信性评估方法、门禁节点选择方案以及蠕虫标签的投送策略.

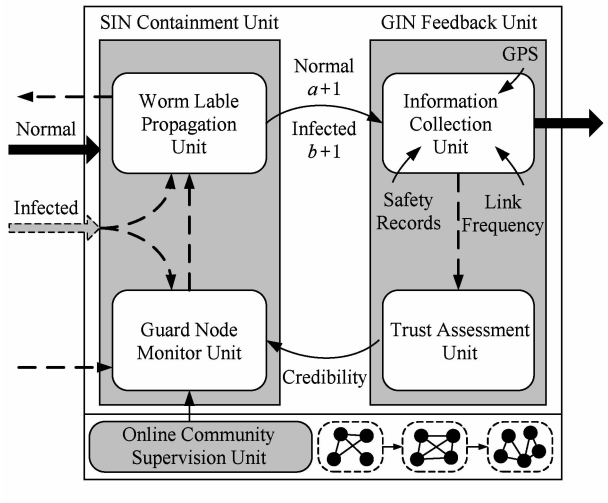


Fig. 4 Worm containment system overview.

图 4 蠕虫遏制系统总览

3 遏制系统组件设计

3.1 基于 SIN 的在线式社区监测

3.1.1 挖掘静态社区结构

为了能够实现遏制系统的在线式监控要求,本文提出一种基于历史模型校准的双阶段演化社区识别框架.在框架的第 1 阶段,首先对动态系统的初始状态进行静态社区识别,然后适应性地调整前一时刻的社区结构,以此获得当前时刻的社区划分结果.

首先给出静态加权网络社区挖掘算法.目前,尚无关于社区结构的形式化定义,但在无权网络中一个普遍被认可的观点是社区内部链接总数远大于社区间链接总数.借鉴该思想,本文提出一种节点的倾向性判别方法,并认为理想的社区结构应具有较高的内倾向性.

定义 1. 局部平均权重.设 η_u 为节点 u 的节点度,$\sigma(u)=\{i|(u,i)\in E\}$ 表示与节点 u 具有直接链接关系的节点集,则 u 的局部平均权重定义为

$$\delta(u)=\frac{1}{\eta_u}\sum_{v\in\sigma(u)}w(u,v). \tag{11}$$

对于任意节点 $i\in\sigma(u)$,如果 $w(u,i)>\delta(u)$,则称边是(u,i)是以 u 为核心构成的局部环境中的强指向链接,反之为弱指向链接.因此可定义指向性:

$$\theta_{u,v}=\begin{cases}1, & w(u,v)>\delta(u);\\ 0, & \text{else}.\end{cases} \tag{12}$$

链接的局部强弱程度也可以通过

$$w(u,i)>\delta'(u)=0.5\max_{i\in\sigma(u)}w(u,i) \tag{13}$$

进行判别,其代表的是以节点 u 为核心的边的权重相比于 u 周围最大边权的比重. 局部环境中链接的强弱可以反映人际交往中关系的倾向性,显然人们更愿意加入倾向性较高的社区. 基于这一原则,对于任意节点 u 和社区 C,u 对 C 的隶属度可定义为

$$S(u,C)=\frac{\sum_{v\in C}\theta_{u,v}}{\sum_{v\in V}\theta_{u,v}}, \tag{14}$$

其代表的是节点对社区 C 的指向链接占其所有倾向链接的比例. 当节点 u 所有的倾向链接均指向同一社区时,$S(u,C)=1$.

电导率一直被用作度量社区的内部链接程度,相比于社区质量函数 Q[16],电导率作为一种局部化判定函数灵活性更强,并且具有较低的计算开销. 本文对电导率进行扩展,判断节点对邻接社区的内指向程度,具体为

$$\phi(C)=\frac{cut(C)}{\upsilon(C)}, \tag{15}$$

其中,$cut(C)=\sum_{u\in C,v\notin C}\theta_{u,v}$ 度量社区内部节点的外指向性,$\upsilon(C)=\sum_{u\in C,v\in V}\theta_{u,v}$ 表示社区内部节点对所有邻居节点的指向性之和(包括对社区内部节点的指向性). 可见,电导率越低,社区的内外倾向性比值就越大,其轮廓也就越清晰.

图 3(b)中的社区电导率分别为 $\phi(C_1)=1/2=0.5$;$\phi(C_1)=2/5=0.4$;$\phi(C_3)=1/11=0.09$. 显然社区 C_3 表现出更强的结构特性. 寻找电导率最优的社区划分方式是 NP-难问题[17],为此本文设计了一种启发式算法来解决这一问题. 首先识别网络中具有互倾向性的链接,并以此作为社区核心. 然后对该核心进行扩张,合并社区周围对其隶属度最高的节点并判断的电导率变化,如果有 $\phi(C')<\phi(C)$,则扩张过程继续,否则 C 被定义为新社区. 该过程将持续进行直到网络中没有核心链接为止. 上述过程的伪代码如算法 1 所示.

算法 1. 静态 SIN 社区识别算法.

输入:加权网络 $G=(V,E,w)$;

输出:社区集合 $\mathcal{C}$.

① set $\mathcal{C}=\varnothing$;

② 根据式(12)初始化核心链接集合 E_{core};

③ while $E_{\text{core}}\neq\varnothing$ do

④ 随机选取$(u,v)\in E_{\text{core}}$,$C=\{u,v\}$;

⑤ while $\sigma(C)\neq\varnothing$ do

⑥ $C'=C\cup\{\arg\max_{x\in\sigma(C)}S(x,C)\}$;

⑦ if $\phi(C')<\phi(C)$ then

⑧ $\mathcal{C}=\mathcal{C}\cup C$;

⑨ else

⑩ break;

⑪ end if

⑫ end while

⑬ $E_{\text{core}}=E_{\text{core}}\backslash(u,v)$;

⑭ end while

⑮ return $\mathcal{C}$.

算法 1 结束后,输入网络被描绘成由若干重叠社区组成的分块网络. 某些社区可能具有相同的节点集或者社区之间共享有高度重叠的子结构,因此即使对这些社区进行合并也不会破坏社区的内聚性,为了实现这一目的,下面给出同源社区的概念.

定义 2. 同源社区. 对于任意社区 $C_a,C_b\in\mathcal{C}$,令 $O_{ab}=C_a\cap C_b$ 表示社区间的重叠部分,$\bar{O}_x$ 表示 O_{ab} 相对于社区 C_x 的补集,如果满足:

$$0.5\,|\bar{O}_{a\vee b}|\leqslant\sum_{u\in C_a\vee C_b\backslash O_{ab}}(\max_{v\in O_{ab}}\theta_{u,v}), \tag{16}$$

则称 C_a 与 C_b 为同源社区.

同源社区具有相同的核心指向性. 式(16)度量的是社区合并后的分裂倾向性,如果社区非重叠部分节点倾向于重叠部分节点,则合并后的社区同样具有较强的内聚性,进而合并操作不会破坏原社区的稳定性. 反之,如果重叠部分节点具有较高的外倾向性,则社区面临分裂的风险.

算法 1 主要的计算复杂性体现在算法行③~⑭,由于集合 E_{core} 的最大规模为 $O(m)$ 且 $\sigma(C)$ 的大小通常为 η_{SIN},因此算法 1 的时间复杂度为 $O(\eta m)$. 根据定理 1,可知合并同源社区的复杂度同样为 $O(\eta m)$,因此静态社区识别的总体复杂度为 $O(\eta m)$.

定理 1. 在具有 $|\mathcal{C}|$ 个社区的复杂网络中,合并同源社区的时间复杂度为 $O(\eta|\mathcal{C}|)=O(\eta m)$.

证明. 对于任意 $C_a\in\mathcal{C}$,由于社区具有重叠性,因此有 $|\mathcal{C}|<\rho(C)\cdot|\mathcal{C}|=\sum_{C_a\in\mathcal{C}}|C_a|$,$\rho(C)$ 为社区的平均节点数. 此外 $\sum_{C_a\in\mathcal{C}}|C_i|=n+\sum_{a<b}|C_a\cap C_b|$. 并且由于算法在扩张过程中始终选择社区的邻接节点进行合并,因此 $|C_a\cap C_b|<\eta_{\text{SIN}}$. 显然重叠部分数量小于 n,则 $\sum_{a<b}|C_a\cap C_b|<\eta_{\text{SIN}}n$. 综上,$|\mathcal{C}|<(\eta_{\text{SIN}}+1)n$. 由于社会网络的平均节点度通常为 $\eta_{\text{SIN}}=2m/n$,因此 $O(\eta|\mathcal{C}|)=O\left(\eta\left(\frac{2m}{n}+1\right)n\right)=O(\eta m)$. 证毕.

3.1.2 动态社区结构校准

通常来讲,动态网络在各时间片段所发生的演化事件可以用节点的添加和删除、链接的添加和删除加以描述.通过更细致的观察,我们发现某些事件可以进一步分解为一系列其他事件.例如,当1个新用户加入网络时,该用户通常会与网络中的其他用户建立新联系,如果将该用户看作孤立节点,并将其邻接边看作网络中添加的新链接,则节点的添加即可用添加新链接的集合形式加以描述.这意味着高效的链接变化校准算法足以处理动态网络中所有的拓扑变化.

当1条新链接(u,v)加入网络,该链接可能出现以下2种情况:1)位于某社区内部;2)链接于不同社区之间.对于情况1),新链接的到来可能影响到端节点$\{u,v\}$对社区内其他节点的指向性.如果$w(u,v)<\min\{2\delta(u),2\delta(v)\}$或者$w(u,v)>\max\{2\delta(u),2\delta(v)\}$,则社区结构不会发生变化.原因是前者不会影响$u$和$v$对周围节点的指向性判定,而后者使得新链接$(u,v)$必然成为核心链接,从而保证了节点$u$和节点$v$仍然收敛于原社区.如果有$w(u,v)>2\delta(u)$或$w(u,v)>2\delta(v)$,则$u$或$v$对社区内其他节点的指向性就会发生变化,从而影响到社区的电导率.但在该情况中,由于u和v对社区外部节点的指向性同样被削弱,因此相比较而言u和v仍然会维持在原有社区中.因此对于情况1),本文选择不对现有社区作任何调整.

相比于情况1)而言,情况2)的处理较为复杂.首先,新链接可能具有较大的边权从而成为核心链接.此时需要对该链接进行扩张,以此生成新社区.如果新链接没能成为核心链接,则需要判断端点对邻接社区的归属性.需要注意的是,当节点发生社区转移行为以后,节点的邻接边会变为跨社区链接,因此对其邻居节点同样需要判断社区归属性.根据定理2,我们可以以一种局部化思想来判定节点是否需要加入新社区,以此避免算法的重复执行.总体上,添加新链接的社区校准过程如算法2所示.

定理2. 令$L(u,C)=\sum_{v\in C}\theta_{u,v}+\theta_{v,u}$表示节点$u$与社区$C$的交互次数.则对于任意节点$u\in\sigma(C)$,如果满足$L(u,C)>(1-\phi(C))\sum_{v\in V}\theta_{u,v}$,则$u$将加入社区$C$.

证明. 首先分析u加入社区后$cut(C)$和$\upsilon(C)$的变化情况.令k_u^{out}表示节点u的外指向性,则u加入社区C后,新社区C'的整体交互程度变化为

$$\Delta_\upsilon=k_u^{\text{out}}S(u,C)+k_u^{\text{out}}(1-S(u,C)),\quad(17)$$

方程的前后项分别度量节点u对社区的内部和外部指向性,由于社区内部节点对u的内指向程度已在$\upsilon(C)$中有所体现,因此无需重新计算.对式(17)进行扩展,可得:

$$\upsilon'(C)=\upsilon(C)+k_u^{\text{out}}.\quad(18)$$

$cut(C)$的计算需要减去原有的外指向链接并加上新增加的外指向链接,因此有:

$$\Delta_{cut}=k_u^{\text{out}}(1-S(u,C))-(L(u,C)-k_u^{\text{out}}S(u,C)),\quad(19)$$

可见:

$$cut'(C)=cut(C)+k_u^{\text{out}}-L(u,C).\quad(20)$$

如果u满足社区合并条件,显然有$\phi(C)>\phi(C')$,进而可得

$$cut(C)/\upsilon(C)>cut'(C)/\upsilon'(C).\quad(21)$$

将式(18)和式(20)代入式(21),可得$L(u,C)>(1-\phi(C))k_u^{\text{out}}$.由于$k_u^{\text{out}}=\sum_{v\in V}\theta_{u,v}$,则$L(u,C)>(1-\phi(C))\sum_{v\in V}\theta_{u,v}$. 证毕.

算法2. 链接添加校准算法.

输入:$E=E\cup(u,v)$,社区集合$\mathcal{C}^{(t-1)}$;

输出:社区集合$\mathcal{C}^{(t)}$.

① if $C_u\neq C_v$ and $\theta_{u,v}+\theta_{v,u}\neq 2$ then
② 根据定理2重新决定u和v的社区归属;
③ if $u\vee v$改变所属社区 then
④ $\sigma(u)=\{x|(u,x)\in E\}$;
⑤ for $a\in\sigma(u)$ do
⑥ 根据定理2判定a的社区归属;
⑦ if a改变社区归属性 then
⑧ 将a转移到新社区;
⑨ end if
⑩ end for
⑪ end if
⑫ else if $C_u\neq C_v$ and $\theta_{u,v}+\theta_{v,u}=2$ then
⑬ 扩张新社区$C=\{u,v\}$直到$\phi(C)<\phi(C')$;
⑭ else
⑮ $\mathcal{C}^{(t)}\leftarrow\mathcal{C}^{(t-1)}$;
⑯ end if
⑰ update $\mathcal{C}^{(t)}$.

如果链接(u,v)从网络中移除,则可能:1)(u,v)的端点存在度为1的节点;2)(u,v)位于两社区之间;3)(u,v)位于某社区内部.对于情况1),将u或v从现有社区移除即可完成校准工作.对于情况2),定理3表明跨社区链接的删除不会对社区结构产生

任何影响,因此保持现有社区不变.在情况 3)中,由于社区在构建过程中始终保证节点对社区内部的强指向性,因此大多数情况下,删除某几条链接不会影响节点对社区的归属性.但某些情况下删除内部链接可能会引发社区结构的破裂,原因是某些由同源社区合并而来的新社区结构不是十分紧密,因此当内部链接被删除以后,合并前的社区失去了对重叠部分的指向性.对于这一问题,本文的做法是对这两部分节点分别执行局部化的核心扩张过程,以此获得内聚性最强的局部社区.根据上述分析,我们给出算法 3 来处理删除新链接后社区结构的变化情况.

定理 3. 如果 C_u 和 C_v 是网络中的 2 个社区结构,则删除 C_u 和 C_v 之间的链接(u,v)不会对 C_u 和 C_v 产生任何影响.

证明. 根据算法 1,由于链接(u,v)位于两社区之间,因此(u,v)不会是核心链接,进而有 $\theta_{u,v}=\theta_{v,u}=0$;如果 $w(u,v)$很小以至于(u,v)没有任何指向性,则删除(u,v)不会影响电导率 $\phi(C_u)$和 $\phi(C_u)$;当(u,v)具有单指向性时,$\theta_{u,v}$或 $\theta_{v,u}$等于 1,由于 $cut(C)<\phi(C)$,因此假如删除 1 条外指向链接,则根据式(15)可得 $\phi(C'_{u\vee v})<\phi(C_{u\vee v})$,此时社区轮廓更加清晰,进而 C_u 和 C_v 保持不变. 证毕.

算法 3. 链接删除校准算法.

输入:$E=E\backslash(u,v)$,社区集合 $\mathcal{C}^{(t-1)}$;

输出:社区集合 $\mathcal{C}^{(t)}$.

① if $\eta_u=\eta_v=1$ then

② $\quad\mathcal{C}^{(t)}=(\mathcal{C}^{(t-1)}\backslash\{u,v\})\cup\{u\}\cup\{v\}$;

③ else if $\eta_{u\vee v}=1$ then

④ $\quad\mathcal{C}^{(t)}=(\mathcal{C}^{(t-1)}\backslash C_u)\cup\{u\}\cup\{C_u\backslash u\}$;

⑤ else if $C_u\neq C_v$ then

⑥ $\quad\mathcal{C}^{(t)}=\mathcal{C}^{(t-1)}$;

⑦ else if $C_u=C_v$ then

⑧ $\quad$if $S(u,C)=\max\limits_{i\in\sigma(u)}S(i,C)$ then

⑨ $\quad\quad\mathcal{C}^{(t)}=\mathcal{C}^{(t-1)}$;

⑩ $\quad$else if $S(u,C)\neq\max\limits_{i\in\sigma(u)}S(i,C)$ then

⑪ $\quad\quad$根据定理 2 判定 u 的社区归属;

⑫ $\quad$else

⑬ $\quad\quad V(C)\leftarrow C_{u,v}$;

⑭ $\quad\quad S_1\leftarrow$距 u 最近的核心扩张所得社区;

⑮ $\quad\quad V(C)\leftarrow V(C)\backslash S_1$;

⑯ $\quad\quad S_2\leftarrow$距 v 最近的核心扩张所得社区;

⑰ $\quad\quad V(C)\leftarrow V(C)\backslash S_2$;

⑱ $\quad\quad$使用定理 2 判定 $x\in V(C)$的社区归属;

⑲ $\quad$end if

⑳ end if

㉑ update $\mathcal{C}^{(t)}$.

由于算法的局部化校准策略,算法 2 和算法 3 的时间复杂度均为 $O(\eta\varepsilon)$.ε 为网络演化时发生的事件数,在本文中为新时间片段链接的增删总量.考虑最极端的情况,即演化过后网络会将所有链接全部替换,此时 $O(\eta\varepsilon)=O(\eta m)$.极端情况的存在缺乏合理性,在真实网络中算法可以在很短的时间内完成对社区的校准工作,从而实现对社区的实时性监控.

3.1.3 基于社区的门禁节点选择与标签投送

以社区结构为基础,本节设计相应的门禁节点选择算法和蠕虫标签投送策略.蠕虫标签投送的目的在于防止蠕虫扩散到其他社区,并且如果投送转发的效率较高,蠕虫标签可以赶在蠕虫之前对节点进行免疫,从而降低蠕虫的感染率.门禁节点是指决定社区与外界通信的那部分节点,根据文献[18]的推论,对随机选择节点的邻居节点的免疫策略可以近似覆盖整体网络,因此在本文中门禁节点被定义为社区重叠部分节点的邻居集合.这部分节点只需有限的跳数即可跨越不同的社区,因此具有很高的转发效率.

当节点 u 生成或收到蠕虫标签并打算转发时,u 会优先选择比自身转发效率更高的节点进行投送.本文基于识别出的门禁节点重新设计了式(15),将节点对社区的倾向程度定义为其对该社区内门禁节点的指向性之和,具体为 $BS(u,C)=\sum\limits_{v\in V_G,v\in C}\theta_{u,v}$.此外将节点对各社区的倾向属性集合$\{\omega_1,\cdots,\omega_x,\cdots,\omega_y\}$作为附加信息随标签一同传播,以此判定邻接节点是否超出现有节点的转发效率.如果新节点对某社区的倾向程度大于原节点对该社区的倾向程度,则认为新节点的转发效率更高,此时将标签投送至该节点,并由该节点继续标签的转发工作.如果新节点的转发效率较低,标签仍然会投送至该节点,但节点不会继续投送标签,从而降低了网络的带宽资源开销.算法 4 描述了上述投送策略.

算法 4. 基于社区的蠕虫标签投送算法.

输入:社区集合 $\mathcal{C}$;

输出:门禁节点集合 V_G 及标签投送结果.

① set $V_G=\varnothing$;

② for $C_a,C_b\in\mathcal{C}$ do

③ $\quad$if $C_a\cap C_b\neq\varnothing$ then

④ $\quad\quad V_G\leftarrow\{v\mid v\in\sigma(u),u\in C_a\cap C_b\}$;

⑤ end if
⑥ end for
⑦ for 节点 i 缓冲区内的蠕虫标签 do
⑧ 更新投送判别集合$\{\omega_1,\cdots,\omega_x,\cdots,\omega_y\}$;
⑨ if 存在社区 C_x,使得 $BS(j,C_x)>\omega_x$ then
⑩ if 该标签对于 j 而言为新标签 then
⑪ 复制节点 i 的蠕虫标签并投送至 j;
⑫ end if
⑫ end if
⑭ end for

3.2 基于 GIN 的节点信任性评估

GIN 反馈单元收集来自 SIN 的历史安全信息并提供对节点的信任性评估.在移动互联网中,节点间的信任性被定义为节点对临近蠕虫的抵抗能力,一个具有良好使用习惯和安全记录的用户通常对恶意程序的抵抗力较强.信任性评估的目的有 3 个:1)对社区之间的脆弱链接进行拒绝通信,进一步控制社区间蠕虫的传播;2)切断社区内部易感染节点对之间的联系,降低蠕虫在社区内部的传播速度;3)限制不属于门禁节点但地理位置过于接近的节点对之间的通信,防止 SIN 和 GIN 间的协同感染.

首先给出节点之间的安全性分析,本部分的目的在于根据历史安全记录预测节点未来的安全性.节点间每轮通信中是否包含恶意行为可以由 SIN 监控单元进行记录,在本文中我们使用参数 a 记录节点之间的正常通信,用参数 b 记录节点之间的恶意通信,进而节点的安全性可用式(22)加以描述:

$$\alpha=\frac{a}{a+b}. \tag{22}$$

式(22)度量的是节点间的正常通信占全部通信的百分比.当网络未存在监控行为时,a 和 b 的初始值均设置为 1.这种度量方式比较直观但存在一定的局限性,例如现有 2 种安全记录方式:一种是在总数为 2 的通信中有 1 次被检测为恶意通信;另一种是在总数为 100 的通信中有 50 次被检测为恶意通信.这 2 种记录体现出的安全性均为 $\alpha=0.5$,但显然后一种由于具有更多的证据确定性更强.而前一种则无法确定后续的安全情况.为此本文采用一种贝叶斯推理方式,构建参数 β 来度量安全性的不确定程度,具体为

$$\beta=\frac{\min\{a,b\}}{\max^2\{a,b\}}. \tag{23}$$

可以看出:β 值越大,节点安全性评估的可靠性越低,这表明由于证据的不足,节点安全的可预测性较低.当 a 值和 b 值同时很大或其中一方占据主导地位时,β 值变小,这表明我们有足够的证据来验证节点的安全程度.通过将节点的安全性与度量的不确定性相结合,根据历史安全记录实际获得的节点安全性为

$$e_{u,v}=\alpha(1-\beta). \tag{24}$$

在对前例中的 2 种历史记录进行安全性计算后可得 $e_1=0$ 和 $e_2=0.49$,显然后者的安全性更高.

除了需要度量节点间的安全性以外,节点间的交互频率也是推理中的证据之一,这意味着即使节点间的安全性较高,但如果交互频率较低仍然无法说明节点间实际的可信性.本文在链接的频繁性提取上使用了大小为 ws 的滑动窗口.如图 5 所示,各滑动窗口描述了来自 SIN 和 GIN 的所有节点的交互记录.其中深色部分表明节点之间在此时间片段具有交互行为,白色部分表示该阶段没有进行通信.链接的频繁性实际度量的是节点间交互的周期性,为此本文提出一种循环链接的概念(如图 5 中窗口 2 的虚线框部分),认为如果未通信部分前后均具有通信行为,则该间隔可看作是常规通信,令 n_{normal} 表示滑动窗口中的正常通信次数,n_{cycle} 表示窗口中的循环次数,则链接频繁性可被度量为

$$P_{u,v}=\frac{n_{\text{normal}}+n_{\text{cycle}}}{ws}. \tag{25}$$

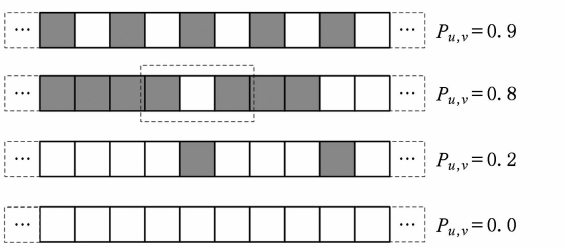


Fig. 5 Frequency measurement of communications.
图 5 链接频繁性度量

对图 5 所示的 4 条时间窗口分别计算链接频繁性,可得 $P_1=0.9$,$P_2=0.8$,$P_3=0.2$,$P_4=0.0$.这表明窗口 1 和窗口 2 具有较强的周期性,而窗口 3 和窗口 4 中节点的通信频率较低.本节拟构建的节点信任性评估主要基于节点间的安全性和交互频率.直观上讲,所构建的信任性评估函数应具有以下 2 种特性:

性质 1. 节点间的安全性越高,交互程度越频繁,节点间通信的可信性越高;但安全性较高的同时交互程度较低,则缺乏足够的证据说明节点间的通信安全性.

性质 2. 节点间的安全性越低,交互程度越频繁,

节点间通信的可信性越低;但安全性较低的同时交互程度较低,则缺乏足够的证据说明节点间面临通信风险.

根据上述性质,本文将节点间的信任性评价函数定义为一种带惩罚的安全性评估方式,具体为

$$T_{u,v}=e_{u,v}-\frac{(e_{u,v}-0.5)}{2}(1-P_{u,v})^2, \quad (26)$$

其中分子部分用来判定需要对安全性进行增益还是惩罚,平方项为根据交互频率定义的增益或惩罚程度.下面给出若干实例来说明式(26)的有效性.

表1中信任性最高的安全记录方式为记录5,其次是记录7,这表明我们有充分的证据表明该记录的可信性.记录2,3由于缺乏明确的证据来表明通信的安全性,因此信任性略低.此外随着交互频率的不断增加,安全记录大于0.5的可信值会不断上升,而安全记录小于0.5的可信值会不断降低,符合性质1和性质2的要求.

Table 1 Trust Analysis with Different Input Parameter

表1 不同输入参数下的节点信任性

Record	a/b	α	β	e	$T(0.1)$	$T(0.5)$	$T(0.9)$
1	1/1	0.50	1.00	0.000	0.202	0.062	0.002
2	5/5	0.50	0.20	0.400	0.441	0.413	0.401
3	10/10	0.50	0.10	0.450	0.471	0.456	0.451
4	1/10	0.09	0.01	0.089	0.256	0.141	0.091
5	10/1	0.90	0.01	0.891	0.733	0.842	0.889
6	5/10	0.33	0.05	0.313	0.389	0.336	0.314
7	10/5	0.67	0.05	0.636	0.581	0.619	0.635

根据所定义的节点信任性评价函数,我们给出算法5对通信威胁较高的节点做隔离处理.

算法5. 基于节点信任性评估的社区隔离算法.

输入:$G=(V,E,w)$,信任函数 $T_{u,v}$;

输出:社区隔离结果.

① if u,v 位于同一社区或 u,v 属于门禁节点集或 $R_{u,v}<30$ then

② if u 接收到新蠕虫标签且 $T_{u,v}<0.5$ then

③ 节点 u 开始拒绝接受 v 所在社区节点的通信请求(蠕虫标签除外);

④ 节点 u 向所在社区的其他节点发送隔离通知,当 v 所在社区对这部分节点发送通信请求时,判定该通信的安全性;

⑤ end if

⑥ end if

算法5中行①的节点距离判定取自GPS地理数据,根据文献[19]所述,短距离通信的最大有效半径为30 m.通过对低信任节点进行通信限制,蠕虫在SIN和GIN上的传播可以同时得到有效的遏制.算法5主要的时间开销为算法行②$T_{u,v}$中频繁度 $P_{u,v}$ 的计算上.在 $P_{u,v}$ 中,循环链接可用复杂度为 $O(n)$ 的KMP算法得以识别,因此算法5最终的时间复杂度为 $O(n)$.

4 实验结果与分析

在本节中,我们通过对真实数据集上的仿真实验验证混合蠕虫遏制系统HWCS的效率和有效性.实验的运行环境为Inter Pentium Ⅳ 3.0 GHz处理器,2 GB内存,Windows XP操作系统,实现算法采用C++与Matlab混合编程.

4.1 实验准备

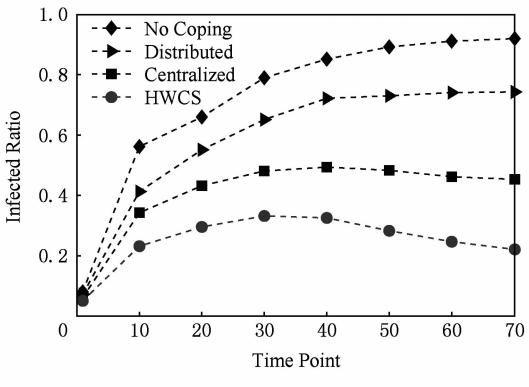
本文选用MIT Reality与Haggle Project作为实验的仿真数据(源自CRAWDAD).所选数据集以智能手机为载体进行数据采集,并且提供用户的通话记录(包括通话时长、通话次数等)以及蓝牙通信记录(包括蓝牙交互的发起时间、结束时间、节点IDs).我们根据用户的通话对象以及通话次数构建了相应的加权社会信息网络SIN,根据用户的地理位置以及蓝牙交互记录生成了相应的地理信息网络GIN,并选择30%左右的总数据作为历史记录,以生成用户在GIN上的安全信息.蠕虫的传播模型采用第2节提出的混合蠕虫传播模型,生成方式采用泊松过程,其在SIN上的传播基于用户的通信列表(本文中为通话次数排名前10的联系人).在GIN上,蠕虫会扫描所有的临近设备,然后优先扩散到距离较近的设备上.

4.2 蠕虫遏制系统性能分析

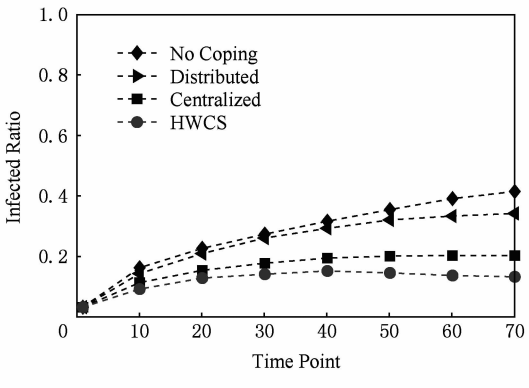
实验1. 验证蠕虫遏制系统的有效性,实验指标为感染节点所占的百分比.显然感染节点数越少遏制性能越优.选取2个分别面向SIN、GIN的蠕虫遏制方案进行性能对比,SIN蠕虫遏制方案选取文献[7]的集中式蠕虫遏制方法(记为Centralized方案),GIN蠕虫遏制选取Zyba等人在文献[3]中提出的纯分布式蠕虫遏制方案(记为Distributed方案).所有相关参数均采用与原文相同的配置.仿真结果如图6所示.

图6中菱形曲线代表的是不采取任何遏制方案时蠕虫最终感染的节点数.MIT Reality数据集具有较低的节点密度分布,从而变相地降低了蠕虫扩

散的可能性,因此图 6(b)中蠕虫的传播效率明显弱于图 6(a). 本文提出的 HWCS 在 2 个数据集上均成功地遏制了蠕虫的进一步传播,并且具有最优的遏制性能. 相比较而言,当面临混合蠕虫传播时,SIN 蠕虫遏制性能(方块形曲线)要优于 GIN 蠕虫遏制性能(右三角形曲线),这主要是由于蠕虫在 SIN 上的传播速度明显快于在 GIN 上的传播速度(如图 2 所示),并且由于泛洪式蠕虫阻断方案会占据大量网络资源,因此 Distributed 方案效率较低. SIN 上的集中式遏制方案由于整合了网络不同分块中的节点信息,因此遏制更有效率. 但由于其没有考虑不同分块区域中节点的物理性感染,因此蠕虫仍然可以进一步扩散到网络中的其余节点上,进而在性能上逊于 HWCS.



(a) Haggle dataset



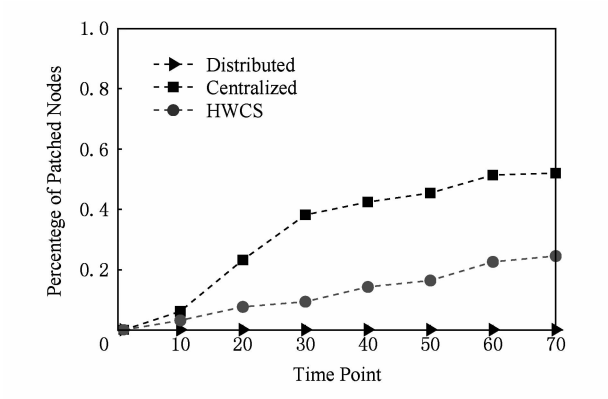
(b) MIT reality dataset

Fig. 6 Performance on worm containment system.

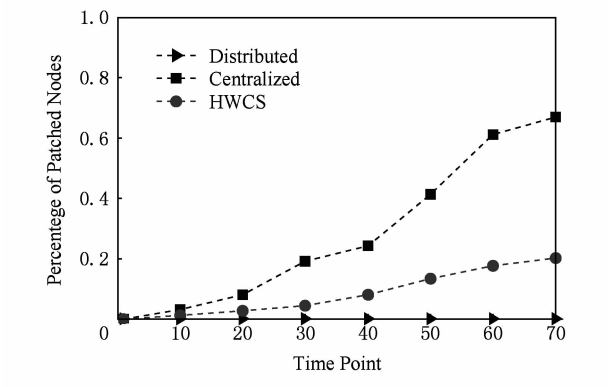
图 6 蠕虫遏制系统性能分析

实验 2. 遏制系统代价分析. 首先判定遏制系统性能与网络资源占用上的权衡. 理想的遏制系统能够在遏制蠕虫传播的同时保证原网络的正常功能. 在实验 2 中,遏制蠕虫所需投送标签的节点数为考察指标,实验结果在图 7 中给出.

Distributed 遏制方案由于只单纯地阻止蠕虫



(a) Haggle dataset



(b) MIT reality dataset

Fig. 7 Analysis of total number of patching node.

图 7 投送节点数量分析

的物理传播,因此无需任何标签投送,但其同时也失去了与网络中其他节点交换安全信息的机会,这导致其在实验 1 中具有最差的遏制性能. Centralized 方案在实验 1 中表现出不错的遏制性能,但其对分块中所有节点同时分发标签的遏制方式对网络造成了大量的负担. 通过对图 7 进行分析,相比于 HWCS, Centralized 方案约需要额外的 1~3 倍标签投送数,而 HWCS 虽然在前 30 时间片段没能体现出明显的优势,但随着时间的累积,节点收集了大量的安全信息,并通过 GIN 反馈单元决定节点之间的通信频率,从而使得即使在社区内部蠕虫仍然具有较低的传播效率. 此外由于选择社区中的门禁节点作为投送对象,HWCS 不但在性能上胜过 Centralized 方案,并且所需投送的节点更少.

接下来验证 HWCS 社区监控单元的时间开销. 选取较为新颖动态社区识别算法 A^3CS[20] 以及 Blondel[21] 等人提出的静态社区挖掘算法作为比对算法,数据集为 MIT Reality. 在图 8 中,HWCS 的平均社区识别时间不超过 0.5 s,并具有较高的稳定性. A^3CS 同样具有较低的时间开销,但稳定性较差.

Blondel 算法无法处理动态网络,进而需要在不同的时间片段重复执行. 因此相比于增量式挖掘的 HWCS 和 A^3CS,效率较低. 图 8 表明,HWCS 的这种在线式监控是有效的,并且具有较高的效率.

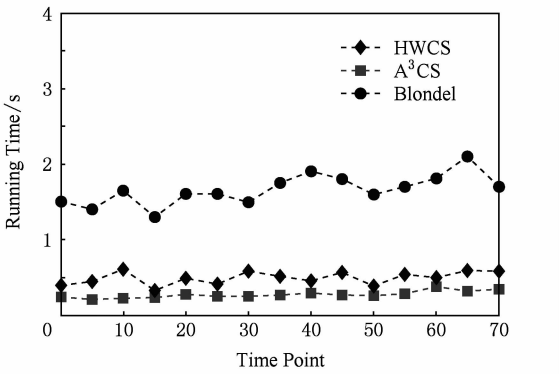


Fig. 8 Performance of community mining algorithms.

图 8 社区识别效率分析

实验 3. 遏制系统效率分析. 由于移动互联网蠕虫具有较高的隐蔽性,因此当蠕虫开始传播时监测系统未必能够在第一时间发现蠕虫并给予警报. 本节实验将充分考虑蠕虫检测系统的反映时间,对网络中的起始感染节点数 μ 进行调整并设计了如下的仿真实验:1)验证不同投送节点数下蠕虫最终的感染率;2)分析在固定投送节点数下蠕虫传播速度的变化.

在仿真实验 1 中,我们选择了前文中表现出较好遏制性能的 Centralized 方法,并选取了 2 种图挖掘方法作为比对算法,包括 Zhu 等人[12]的 Clustering 方法、Blondel 等人[21]的社区挖掘方法. 前者主要针对 MMS 蠕虫而设计,后者被证明具有良好的社区识别性能. 起始感染节点分别为 5%,10%和 20%,具体的实验结果如图 9 所示.

一般来讲网络的感染程度越高所需投送的补丁数越多. HWCS 遏制策略在所有情况下都具有最低的感染率,并且在感染程度 $\mu=20\%$ 时的遏制性能与感染程度 $\mu=10\%$ 时相差不大. 特别地,当 $\mu=5\%$ 时,HWCS 只需要投送 30%的网络节点即可将蠕虫感染率控制在 0.2 以下,是 Blondel 方法效率的 1.6 倍,是 Clustering 方法的 2.8 倍. 当 $\mu=10\%$ 时 HWCS 平均超过 Blondel 方法 6%,超过 Clustering 方法 13%. 当 $\mu=20\%$ 时这一优势扩大到 8%和 17%. 根据 HWCS 的社区挖掘策略,社区重叠部分的邻居节点只需有限次转发即可横跨不同的社区之间,并具有较高的内指向性,这些条件保证了被投送补丁的节点可以将补丁分布到社区的大部分节点中,因此

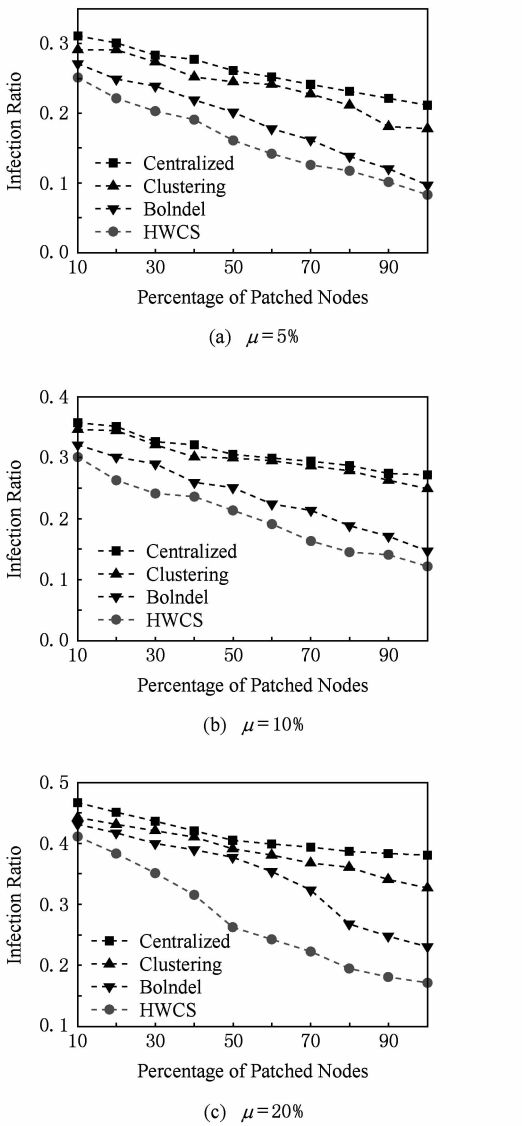


Fig. 9 Performance of node patching strategy.

图 9 节点投送策略效率

HWCS 具有较高的效率,同时这也说明了本文提出的社区挖掘算法所识别的社区具有较高现实意义.

对于仿真实验 2,当投送的节点数固定时,能够以最短的时间将蠕虫控制在一个可接受的传播速度,则说明遏制策略效率更高. 本实验选取网络中 30%的节点作为投送节点,并以随机化投送方式作为基准参考. 由于随机投送完全无视了节点的任何信息,因此随着感染程度的不断增加该方法的遏制性能没有任何根本性的变化.

如图 10 所示,HWCS 可以在最短的时间内遏制蠕虫的进一步传播. 当网络感染程度 $\mu=5\%$ 时,HWCS 只需 10 时间单位即可将蠕虫的传播速度控制在 0.05 左右,而同一时间的随机投送方案早已使蠕虫的传播速度超出 0.3. Centralized 方案可在 35

时间单位将蠕虫传播速度控制在 0.1 左右,同样具有较好的性能.当网络感染程度 $\mu=10\%$时,HWCS 和 Centralized 方案的性能差距开始变大,HWCS 在 30 时间单位左右时可将蠕虫传播速度控制在 0.21,而 Centralized 方案在 50 时间单位时才控制住蠕虫的进一步传播,此时蠕虫传播速度已达 0.4 以上,这表明网络中会有更多的节点被感染.这一差距在 $\mu=20\%$时更加明显,Centralized 方案需要 70 时间单位才能将蠕虫的传播速度控制在 0.7,而 HWCS 在 $\mu=20\%$时的遏制效果与 Centralized 方案在 $\mu=10\%$时较为接近.然而 Centralized 方案仍然要远胜于随机投送方案,因为后者在 80 时间单位时才能控制蠕虫的进一步传播,而此时蠕虫的传播速度已达 0.9 以上.上述实验表明,HWCS 的投送策略具有较高的效率,如果能够尽早地探测到蠕虫,使用 HWCS 对蠕虫进行遏制可以使网络受到极小的影响.

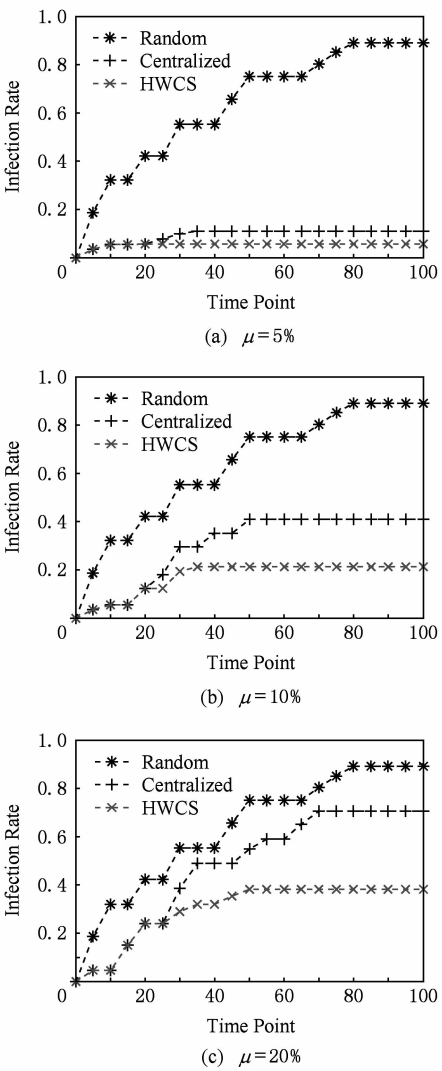


Fig. 10 Infection rate under fixed patching number.

图 10 固定投送节点下的感染速度

实验 4. GIN 反馈单元有效性验证.当 GIN 反馈单元关闭时,社区监测算法占据主导地位,因此本节选取了静态和动态 2 种社区挖掘算法来验证本文方案的有效性,分别为 Blondel 算法和 A^{3}CS 算法.在图 11 中,HWCS 在脱离 GIN 反馈单元之后性能并不十分突出(仅略胜于同样能处理演化社区的 A^{3}CS 算法),这主要是由于蠕虫在社区内部的传播速度同样很快,如果不加以限制蠕虫会以极快的速度传播到所有社区.Blondel 算法由于不断在新时间片段重复执行,进而占用了大量的反应时间,因此在性能上不如 A^{3}CS 算法.当 GIN 反馈单元开始工作后,在前 20 时间单位 HWCS 仍然没有明显的性能优势,这是由于在网络演化的初始阶段 GIN 节点信任性评估单元没能充分收集到节点的安全信息.在 30 时间单位以后,HWCS 通过限制可信程度较低的节点间的通信,降低了蠕虫在社区内部的传播机率,进而有效的降低了网络的整体感染率.

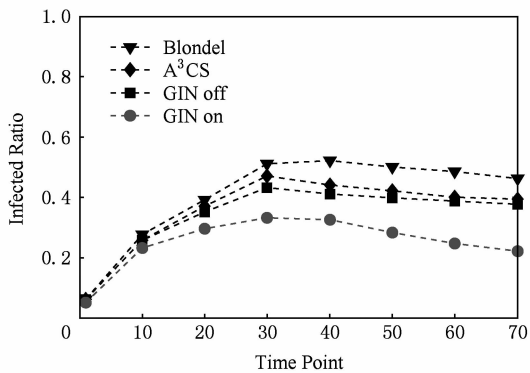


Fig. 11 Performance of GIN feedback component.

图 11 GIN 反馈单元效率分析

5 结 论

本文提出一种基于社区的移动互联网混合蠕虫遏制系统 HWCS.不同于现有的蠕虫遏制方法,HWCS 可以处理蠕虫短程传播和远程传播的并发行为.首先设计了基于节点倾向性判别的在线式社区检测算法,通过对社区间的门禁节点进行蠕虫标签投送,将蠕虫控制在社区内部.由于算法的局部化校准策略,社区结构可以随网络的演化进行实时性调整.然后提出一种节点信任性评估函数,通过收集用户的历史安全信息、蓝牙使用记录以及 GPS 位置数据,控制社区内部节点的通信决定,以此降低蠕虫在社区内部的传播效率.在真实数据集上的仿真结果表明,HWCS 不仅能够有效地遏制蠕虫的传播,同时不会对网络造成过多的通信负担.

参 考 文 献

[1] Su J, Chan K K W, Miklas A G, et al. A preliminary investigation of worm infections in a bluetooth environment [C] //Proc of the 4th ACM Workshop on Recurring Malcode (WORM). New York: ACM, 2006: 9-16

[2] Yan G, Eidenbenz S. Modeling propagation dynamics of bluetooth worms [J]. IEEE Trans on Mobile Computing, 2009, 8(3): 353-368

[3] Zyba G, Voelker G M, Liljenstam M, et al. Defending mobile phones from proximity malware [C] //Proc of the 28th IEEE Int Conf on Computer Communications (INFOCOM). Piscataway, NJ: IEEE, 2009: 1503-1511

[4] Yang Y, Zhu S, Cao G. Improving sensor network immunity under worm attacks: A software diversity approach [C] // Proc of the 9th ACM Int Symp on Mobile Ad Hoc Networking and Computing (MobiHoc). New York: ACM, 2008: 149-158

[5] Mickens J W, Noble B D. Modeling epidemic spreading in mobile environments [C] //Proc of the 4th ACM Workshop on Wireless Security. New York: ACM, 2005: 77-86

[6] Miklas A G, Gollu K K, Chan K K W, et al. Exploiting social interactions in mobile systems [C] //Proc of 9th Int Conf on Ubiquitous Computing (UbiComp). Berlin: Springer, 2007: 409-428

[7] Li F, Yang Y, Wu J. CPMC: An efficient proximity malware coping scheme in smartphone-based mobile networks [C] //Proc of the 29th IEEE Int Conf on Computer Communications (INFOCOM). Piscataway, NJ: IEEE, 2010: 1-9

[8] Qing Sihan, Wang Chao, He Jianbo, et al. Research and development of instant messaging worms [J]. Journal of Software, 2006, 17(10): 2118-2130 (in Chinese)
(卿斯汉, 王超, 何建波, 等. 即时通信蠕虫研究与发展[J]. 软件学报, 2006, 17(10): 2118-2130)

[9] Fleizach C, Liljenstam M, Johansson P, et al. Can you infect me now?: Malware propagation in mobile phone networks [C] //Proc of the 5th ACM Workshop on Recurring Malcode (WORM). New York: ACM, 2007: 61-68

[10] Meng X, Zerfos P, Samanta V, et al. Analysis of the reliability of a nationwide short message service [C] //Proc of the 26th IEEE Int Conf on Computer Communications (INFOCOM). Piscataway, NJ: IEEE, 2007: 1811-1819

[11] Bose A, Hu X, Shin K G, et al. Behavioral detection of malware on mobile handsets [C] //Proc of the 6th Int Conf on Mobile Systems, Applications, and Services (MobiSys). New York: ACM, 2008: 225-238

[12] Zhu Z, Cao G, Zhu S, et al. A social network based patching scheme for worm containment in cellular networks [C] //Proc of the 28th IEEE Int Conf on Computer Communications (INFOCOM). Piscataway, NJ: IEEE, 2009: 1476-1484

[13] Zou C C, Towsley D, Gong W. Modeling and simulation study of the propagation and defense of internet e-mail worms [J]. IEEE Trans on Dependable and Secure Computing, 2007, 4(2): 105-118

[14] Pradip D, Yonghe L, Sajal K D. An epidemic theoretic framework for vulnerability analysis of broadcast protocols in wireless sensor networks [J]. IEEE Trans on Mobile Computing, 2009, 8(3): 413-425

[15] Cheng S M, Ao W C, Chen P Y, et al. On modeling malware propagation in generalized social networks [J]. IEEE Communications Letters, 2011, 15(1): 25-27

[16] Liu Dayou, Jin Di, He Dongxiao, et al. Community mining in complex networks [J]. Journal of Computer Research and Development, 2013, 50(10): 2140-2154 (in Chinese)
(刘大有, 金弟, 何东晓, 等. 复杂网络社区挖掘综述[J]. 计算机研究与发展, 2013, 50(10): 2140-2154)

[17] Leskovec J, Lang K J, Mahoney M. Empirical comparison of algorithms for network community detection [C] //Proc of the 19th Int Conf on World Wide Web (WWW). New York: ACM, 2010: 631-640

[18] Christakis N A, Fowler J H. Social network sensors for early detection of contagious outbreaks [J]. PloS One, 2010, 5 (9): e12948

[19] Wang P, González M C, Hidalgo C A, et al. Understanding the spreading patterns of mobile phone viruses [J]. Science, 2009, 324(5930): 1071-1076

[20] Dinh T N, Nguyen N P, Thai M T. An adaptive approximation algorithm for community detection in dynamic scale-free networks [C] //Proc of the 32nd IEEE Int Conf on Computer Communications (INFOCOM). Piscataway, NJ: IEEE, 2013: 55-59

[21] Blondel V D, Guillaume J L, Lambiotte R. Fast unfolding of communities in large networks [J]. Journal of Statistical Mechanics: Theory and Experiment, 2008(10): P10008

Yang Hailu, born in 1985. PhD candidate in Harbin Engineering University. Student member of China Computer Federation. His main research interests include network security, social computing and machine learning.

Zhang Jianpei, born in 1956. Professor and PhD supervisor in Harbin Engineering University. Senior member of China Computer Federation. His main research interests include network security, social computing and data mining.

Yang Jing, born in 1962. Professor and PhD supervisor in Harbin Engineering University. Senior member of China Computer Federation. Her main research interests include network security, data mining, and privacy preserving (yangjing@hrbeu.edu.cn).