ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.6, June 2008, pp.1508−1518
DOI: 10.3724/SP.J.1001.2008.01508

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

拓扑相关蠕虫仿真分析*

王跃武+, 荆继武, 向 继, 刘 琦

(中国科学院 研究生院 信息安全国家重点实验室,北京 100049)

Topology Aware Worm Simulation and Analysis

WANG Yue-Wu+, JING Ji-Wu, XIANG Ji, LIU Qi

(State Key Laboratory of Information Security, Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)

+ Corresponding author: E-mail: ywwang@lois.cn

Wang YW, Jing JW, Xiang J, Liu Q. Topology aware worm simulation and analysis. *Journal of Software*, 2008,19(6):1508−1518. http://www.jos.org.cn/1000-9825/19/1508.htm

Abstract: This paper provides a complete packet-level topology aware worm simulation model based on worm targets discovery, worm code transmission and worm activation, and designs a directed Small World topology generation algorithm. Then, through selective abstraction, this paper implements a complement packet-level topology aware worm simulation system. Finally, with simulation experiments, the paper analyzes the impacts of topology structure and worm activation on worm propagation. The results of simulation experiments show that this simulation system can provide great support for topology aware worm research.

Key words: topology aware worm; simulation; model; logical topology; Small World model

摘 要: 从蠕虫传播的漏洞主机发现、代码传播以及代码启动 3 个方面提出了一个完整的数据包级拓扑相关蠕虫仿真模型,设计了基于有向图的Small World拓扑结构生成算法,并通过选择性抽象在NS2上实现了一个完整的数据包级拓扑相关蠕虫仿真系统.最后,通过该仿真系统实验分析了逻辑拓扑结构和蠕虫代码启动方式对蠕虫传播的影响.仿真实验结果表明,该仿真系统可以为拓扑相关蠕虫研究提供重要的支持.

关键词: 拓扑相关蠕虫;仿真;模型;逻辑拓扑;Small World 模型

中图法分类号: TP393 **文献标识码**: A

拓扑相关蠕虫(topology aware worm)是利用网络逻辑拓扑结构信息,寻找漏洞主机进行传播的一类蠕虫.因其传播受逻辑拓扑特性的影响较大,因此被称为拓扑相关蠕虫.根据具体实现形式的不同,拓扑相关蠕虫又可以分为多种形式,如利用电子邮件信息传播的 e-mail 蠕虫和利用即时消息(instant message,简称 IM)系统传播的 IM 蠕虫等.拓扑相关蠕虫具有如下特点,使其非常适合于蠕虫传播:① 拓扑相关蠕虫利用逻辑拓扑信息进行传播,提高了目标发现准确率,具有较高的传播效率;② 拓扑相关蠕虫无须进行大规模扫描,因而不会产生大量异常扫描数据包,相对于主动扫描蠕虫具有更高的隐蔽性;③ 拓扑相关蠕虫依赖广泛应用的网络服务形成的应用层逻辑网络进行传播,相对于主动扫描蠕虫具有更加广泛的传播范围;④ 应用层网络信息一般具有较强的私密性,如 e-mail 等,因此对其内容进行检测、防止蠕虫传播要受到更多的法律和政策限制.

* Supported by the National Natural Science Foundation of China under Grant No.60573015 (国家自然科学基金)
Received 2006-10-20; Accepted 2007-02-12

由于存在以上特点,拓扑相关蠕虫长期以来一直是网络安全的主要威胁之一,并且在近年来呈现快速增加的趋势.在大规模网络蠕虫出现初期,网络节点稀疏,主动扫描难以有效地发现目标主机.网络蠕虫大多采用这种拓扑相关的形式,如 Morris 蠕虫[1].其后,随着 e-mail 服务逐渐流行,e-mail 系统中不同的邮件地址构成了一个巨大的应用层逻辑网络,为拓扑相关蠕虫提供了一个有效的传播平台.这段时间内爆发了多种 e-mail 蠕虫,造成了巨大的破坏,如 Melissa[2],Love letter[3]]等.目前,IM 系统应用得到了迅猛发展,用户数量日益庞大,通信量不断增加,这为IM蠕虫产生和发展提供了良好的外部条件.相关研究显示,2005年后,IM蠕虫开始出现快速增长的趋势,极有可能成为下一代网络安全威胁的一种主要形式[4].

近年来,蠕虫研究领域对拓扑相关蠕虫给予了较多的关注.卿斯汉等人对 IM 蠕虫进行了系统的研究[4],Wong 等人分析了 Sobig[5]等蠕虫爆发在局域网内引起的流量异常行为[6],Jiang 等人根据 e-mail 蠕虫传播中表现出来的用户行为信息,提出了蠕虫检测方法[7],Newman 等人应用数学解析方法分析了 e-mail 网络结构的统计特性,并分析了蠕虫传播情况[8].

为了系统地分析拓扑相关蠕虫传播的特性,提出有效的防御策略,需要有一个完善的蠕虫仿真模型.但是由于拓扑结构的复杂性和蠕虫行为的随机性,目前尚没有一个完善的拓扑相关蠕虫仿真系统.本文首先在拓扑相关蠕虫和蠕虫仿真技术研究的基础上,从蠕虫漏洞主机发现、蠕虫代码传播和蠕虫代码启动 3 个方面构建了一个完整的数据包级拓扑相关蠕虫仿真模型;其次,在 Watts 的 Small World 模型[9]基础上提出了蠕虫传播逻辑拓扑模型生成算法,并对算法参数和结构属性之间的关系进行了仿真分析,证明了该算法生成的拓扑结构能够满足蠕虫传播仿真的需要;最后,我们通过选择性抽象方法克服了数据包级蠕虫仿真的规模瓶颈,在通用网络仿真器 NS2 的基础上实现了一个完整的数据包级拓扑相关蠕虫仿真系统,并通过该系统进行仿真实验,分析了拓扑结构和用户行为对蠕虫传播的影响.

本文第 1 节介绍相关研究进展.第 2 节描述拓扑相关蠕虫仿真模型的构建.第 3 节介绍拓扑相关蠕虫仿真实现中的一些关键算法.第 4 节介绍仿真实验结果及其分析.第 5 节是总结和工作展望.

1 相关工作介绍

目前主要有 3 种蠕虫仿真技术.一种是将蠕虫传播过程用数学解析的方法进行描述,如一组微分方程或者一组递归公式,一般称这种形式的蠕虫仿真模型为解析模型,如 Two-Factor 模型[10]和 AAWP 模型[11].这些模型的一个重要前提是假定网络为一个全连通的拓扑结构,这显然与拓扑相关蠕虫要求不符,并且解析模型对蠕虫传播中的随机因素考虑不足.对此,Nicol 等人进行了详细的分析[12].虽然一些研究在拓扑理论的基础上采用数学解析的方法对蠕虫传播进行了分析,如文献[8,13]所述,但是这些模型过于简单,无法满足拓扑相关蠕虫传播系统分析的需求.

另一种是数据包级蠕虫仿真模型.在数据包级蠕虫仿真模型中,通过引入网络仿真技术,构建蠕虫传播网络仿真环境,根据蠕虫行为逻辑规则模拟蠕虫流量,如文献[14]所述.数据包级蠕虫仿真模型可以充分考虑物理网络因素对蠕虫传播的影响,因而能够更加真实地模拟蠕虫的传播过程.但是,数据包级蠕虫仿真存在规模限制瓶颈,拓扑相关蠕虫仿真对此尤其敏感,因为拓扑结构特性只有在一定规模下才能显现出来.针对规模限制瓶颈,相关研究人员分别提出了混合仿真方法,如文献[15]所述,以及并行分布式仿真方法,如文献[16]所述.混合仿真中部分采用了数学解析模型实现,并行分布式仿真则代价昂贵,性价比较低.此外,现有数据包级蠕虫仿真主要针对主动扫描蠕虫,缺少逻辑拓扑模型生成算法及实现,所以无法直接用于拓扑相关蠕虫仿真.

Monte Carlo 仿真是介于数学解析模型和数据包级蠕虫仿真之间的一种仿真方法.它考虑了蠕虫传播中各个节点在不同时刻的状态变化及其概率模型,将蠕虫传播看作一个 Markova 过程,采用离散时间的方法推进仿真过程.Zou 等人在 Monte Carlo 仿真的基础上提出了一个 e-mail 蠕虫传播模型[17].该模型提出并实现了具有 Power-Law 结构特性的电子邮件网络模型.Monte Carlo 仿真相对于数学解析模型能够考虑更多的节点中间状态信息和随机因素的影响,但却不能像数据包级仿真那样考虑更多的网络传输特性,如带宽、延时等因素的影响.此外,离散时间仿真在模拟时间因素的随机性方面不如数据包级仿真采用的离散事件仿真,因为离散时间假

定感染必须在一个时间片内完成,对于跨时间片的感染行为则无法处理.

所以我们提出,在通用网络仿真器 NS2 的基础上,针对拓扑相关蠕虫传播特性进行选择性抽象,克服规模限制瓶颈,构建拓扑相关蠕虫仿真模型.

2 拓扑相关蠕虫仿真模型构建

根据上述分析,数据包级蠕虫仿真是根据蠕虫传播行为,通过节点和链路模型模拟蠕虫传播的流量,分析蠕虫传播的特性.所以,构建数据包级蠕虫仿真模型必须首先构建完善的蠕虫传播行为模型.Weaver 等人提出将蠕虫分成 4 个环节:漏洞主机发现、蠕虫代码传播、蠕虫代码启动和蠕虫扩展功能执行[18].4 个环节构成了一个完整的蠕虫行为框架.由于本文主要研究蠕虫传播特性仿真,所以我们在前 3 个环节的基础上进行蠕虫仿真模型构建.

2.1 拓扑相关蠕虫漏洞主机发现模型

拓扑相关蠕虫通过逻辑拓扑信息发现目标主机.当主机被感染时,蠕虫程序查找该主机上保存的逻辑拓扑信息,向这些逻辑拓扑信息指定的主机发送蠕虫数据包进行传播.忽略逻辑拓扑信息获取的具体方式,拓扑相关蠕虫的目标发现模型取决于逻辑拓扑模型.

逻辑拓扑模型可以用一个有向图 $G=\langle V,E\rangle$表示,任意一个顶点$\forall v\in V$表示一个主机,任意一条边$\forall e=\langle u,v\rangle\in E$, $u,v\in V$,表示主机 u 中保存有主机 v 的地址信息.$|V|$表示蠕虫传播环境中主机的总台数,$|E|$表示这种逻辑关系的总数.逻辑拓扑最终通过物理拓扑实现,逻辑拓扑与物理拓扑的对应关系如图 1 所示.

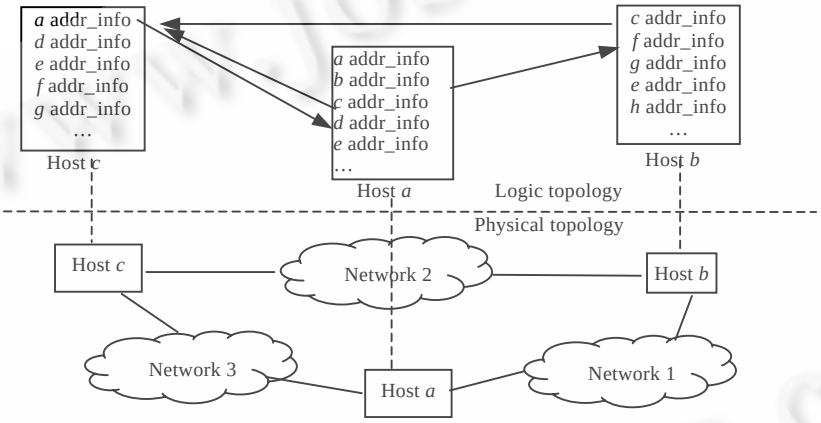


Fig.1 The relationship between logical topology and physical topology

图 1 逻辑拓扑和物理拓扑对应关系

由于逻辑拓扑的地址信息一般具有很大的私密性,如电子邮件地址信息、IM 账户信息等,所以很难通过直接测量确定这种逻辑拓扑结构模型.Zou 等人通过分析 yahoo e-mail 群规模分布,发现其存在 Power-Law 分布,所以采用 Power-Law 模型构建 e-mail 网络拓扑模型[17].而 Ebel 等人通过对邮件服务器 Log 文件进行分析,发现邮件网络节点的出度呈现 Power-Law 分布,同时网络结构呈现 Small World 特性[13].很明显,这些逻辑拓扑关系为人类社会联系模型的映射.而社会联系模型主要为 Small World[9]模型.Small World 结构的两个重要特性为较小的特征路径和较大的聚合系数(将在本文后面给出明确定义).Ebel 等人的 e-mail 网络模型和 Smith 等人的 IM 网络模型[19]都表现出了这两种特性.在社会关系模型向网络环境的转变过程中,因为网络通信的便捷性,可能会发生一些变化,如 yahoo e-mail 群和 QQ 群等通信形式的出现.为了方便获取信息,一个用户在加入用户群时一般总是选择规模较大的群加入,这就导致了群规模的 Power-Law 分布,所以拓扑结构中可能存在节点连接度的 Power-Law 分布.但是可以看出,逻辑拓扑结构的基本关系模型仍为社会关系模型,即 Small World 模型.此外,节点连接度的 Power-Law 分布对蠕虫传播的影响在文献[17]中已有详细描述,所以本文主要研究 Small World 结

构属性对蠕虫传播的影响.

Watts 等人提出了一个 Small World 模型构建方法.该方法可以用图 2 进行解释.可以看出,Small World 模型是介于规律模型和随机模型之间的一种状态.在左边严格规律模型中任选一点为起始点,从与其直接连接的节点中选择顺时针方向距离最近的一个,以概率 P 重新生成这两个节点之间的边,保持源节点不变,从拓扑中随机选择一个节点作为新生成边的另一个端点.依次对规则模型的各条边做同样操作,即可生成 Small World 模型.随着 P 值的不断变大,拓扑结构的随机性不断变大,直至成为完全随机模型.所以,可以通过改变 P 值适应不同蠕虫仿真的需求,提高系统的通用性.

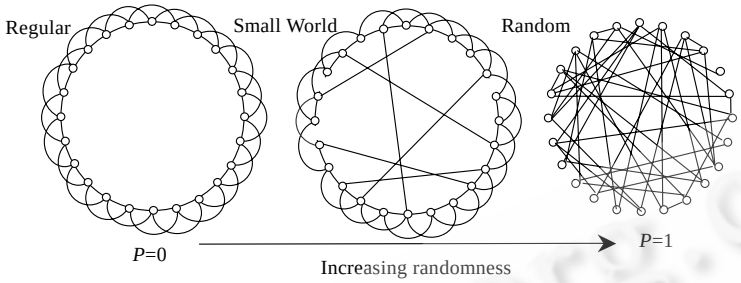


Fig.2 Watt's Small World model

图 2 Watts 的 Small World 模型

实际拓扑结构为有向图结构,如 A 有 B 的邮件地址,而 B 不一定有 A 的地址,而 Watts 的 Small World 模型为无向图结构,所以我们在具体实现过程中对其进行了相应处理,将在后面部分进行详细描述.我们在有向图中定义特征路径长度和聚合系数如下:① 特征路径长度 L 是指 Small World 模型中任意两个节点间最短路径长度的平均值;② 聚合系数 C,假定节点 v 有 k_v 个邻接点,那么有向图中这些节点之间最多可以拥有 $k_v\times(k_v-1)$ 条有向边,$C_v=k_v$ 个邻接点间存在的边数/$k_v\times(k_v-1)$.对于每个节点的 C_v 值取平均值即为 C.

2.2 拓扑相关蠕虫代码传播模型

蠕虫在发现漏洞主机后需要复制自身代码,并将其传播到目标主机.蠕虫代码传播过程具有多种具体实现形式,根据参考文献[18],蠕虫在发现漏洞主机后一般存在3种方式将蠕虫代码传播到漏洞主机,分别为:① 自传播方式,将蠕虫代码作为漏洞主机发现时产生的蠕虫感染数据包的一部分,随着这些感染数据包一起发送到漏洞主机,多数拓扑相关蠕虫都采用该传播方式,如 Serflog 蠕虫[20]的若干变种为利用 MSN 直接向联系人传输经过加壳伪装的蠕虫副本文件,多数邮件病毒也都是将蠕虫作为附件进行传输;② 第二通道传播方式,即蠕虫感染数据包不包含蠕虫代码,当其传播到漏洞主机后开启另一个传播通道,用于蠕虫代码传播,如 Aplore 蠕虫[21]在感染主机的 8180 端口开启 Web 服务,然后向所有的联系人发送指向该感染主机 Web 服务的 URL 信息,诱使联系人从该 URL 下载伪装成 Web 浏览器插件的蠕虫副本;③ 嵌入式传播方式,将蠕虫代码嵌入到正常的网络数据包中,或者替换正常的网络数据包,以正常网络流量的形式传播蠕虫代码,因而具有较高的隐蔽性,如 AimVen[22]蠕虫等,通过修改用户和联系人之间传输的可执行文件传送蠕虫代码.

数据包级蠕虫仿真可以方便地实现 3 种传输方式的模拟.受篇幅限制,本文主要分析了拓扑相关蠕虫采用最多的自传播方式对蠕虫传播的影响.

2.3 拓扑相关蠕虫启动模型

蠕虫代码传送到漏洞主机后,需要激活才能感染其他主机并执行特定的功能,蠕虫可能采用多种激活方式,如 Non-Reinfection 和 Reinfection 启动方式.一些蠕虫,如 Melissa 蠕虫,代码被成功启动一次后,再收到蠕虫代码,将永远不会被再次成功启动,被称为 Non-Reinfection 启动方式.另一些蠕虫,当被感染主机再次收到蠕虫代码后,可以被再次感染并成功启动,被称为 Reinfection.

蠕虫启动方式根据实现原理不同,又可以分为两类:① 依赖计算机系统行为或漏洞的自动启动方式;② 用

户参与的人工启动方式.其中,人工启动方式具有较强的随机性,对蠕虫传播有较大影响,并且是拓扑相关蠕虫采用较多的一种启动方式.所以,本文重点对人工启动方式进行仿真分析,人工启动方式和自动启动方式都可以与 Non-Reinfection 和 Reinfection 启动方式结合.

人工启动方式通常采用一些 Social Engineer 技术提高蠕虫感染概率,因此,其启动过程可以分成两个阶段:① 收取 Social Engineer 信息,如打开邮件阅读蠕虫电子邮件内容、阅读 IM 信息等;② 收取 Social Engineer 信息后采取相应动作,打开或者丢弃蠕虫代码.如因安全意识较差而打开蠕虫邮件附件,被感染.

Social Engineer 信息收取过程复杂,与个人生活习惯和工作性质关系密切.仿真过程中,我们假定蠕虫在一个大型组织内部传播,这样,每个用户的信息获取过程具有相对较高的相似性,简化了仿真模型.在仿真中参考电话问题中的通话时间分布,采用指数分布模型模拟收取 Social Engineer 信息时间 T.对每个节点有:$P(T=t)=\lambda e^{-\lambda t}, t>0$;$P(T=t)=0, t\le 0$;$P(T=t)$表示该节点收取 Social Engineer 信息所用时间 T 等于 t 的概率.通过调整λ值,可以改变 Social Engineer 信息收取时间模型的均值大小.

收取 Social Engineer 信息后,每个节点蠕虫代码被成功启动的概率为 $P_{active}\sim N(\mu_p, \sigma_p^2)$;当 $P_{active}>1$ 时,设定 $P_{active}=1$;当 $P_{active}<0$ 时,设定 $P_{active}=0$;μ_p 为实际过程中蠕虫代码启动成功概率的平均值.根据正态分布特性,蠕虫代码启动成功概率可能的取值范围为$(\mu_p-3\sigma_p, \mu_p+3\sigma_p)$.改变$\mu_p$和$\sigma_p$,可以反映 Social Engineer 技术和安全意识教育对蠕虫传播的影响.如果 Social Engineer 技术非常有效,则蠕虫代码被成功启动的概率增加,μ_p 取值增加.如果安全意识培训较差,那么,虽然 Social Engineer 技术造成的总体感染水平较低,但是组织中每个人启动蠕虫代码的概率差异增加,相应的σ_p 也增加.

通过对节点模型发送蠕虫数据包控制,可以在数据包级蠕虫仿真中方便地实现 Non-Reinfection 和 Reinfection 启动方式模拟.

3 拓扑相关蠕虫仿真实现

我们在通用网络仿真器 NS2 上进行选择性抽象,简化节点和链路模型,克服了数据包级蠕虫仿真的规模限制瓶颈,并且实现了具有 Small World 拓扑结构的逻辑拓扑模型,从而构建了一个完整的数据包级拓扑相关蠕虫仿真系统.下面我们将对系统实现的关键技术:选择性抽象技术和 Small World 逻辑拓扑模型实现进行分析.

3.1 基于选择抽象的仿真网络环境构建

通用网络仿真软件如 NS2,为了适应各种不同仿真的需要,都有各种详细的网络协议仿真模块.所以,系统资源消耗较大,仿真规模受限.拓扑相关蠕虫一般在较大范围内传播,并且逻辑拓扑结构特性对拓扑相关蠕虫的影响往往需要在仿真规模达到一定程度后才能正确反映.为此,我们采用选择性抽象的方法,重新构造网络仿真模块,在保证仿真结果准确的前提下,简化不必要的仿真细节,减少资源消耗,提高网络仿真规模.NS2 的模块化结构为选择性抽象实现提供了方便.

不同于主动扫描蠕虫,拓扑相关蠕虫能够通过逻辑拓扑信息发现目标主机,所以不会产生大量的蠕虫扫描数据包,造成网络拥塞.如在电子邮件系统中,蠕虫可能造成总体邮件数量突然增加,但对于一台主机仍表现为正常网络流量.对于即时消息等系统,网络流量特性变化更小,所以在网络流量仿真方面,我们主要实现带宽、延迟、数据包转发机制、网络路由等正常网络流量下的网络因素仿真,而不进行网络拥塞等异常情况下的网络因素仿真,从而简化节点模型,提高网络仿真效率.为了验证选择性抽象简化对拓扑相关蠕虫仿真的影响,本文在小规模条件下,采用不同的拓扑相关蠕虫模型,分别在详细网络仿真模型和选择性抽象网络仿真模型下进行蠕虫仿真,其结果如图 3 所示.可以看出,两种网络仿真模型下的仿真结果基本保持一致,并且改变拓扑相关蠕虫模型也不影响这种一致性,证明我们的选择性抽象是合理的.

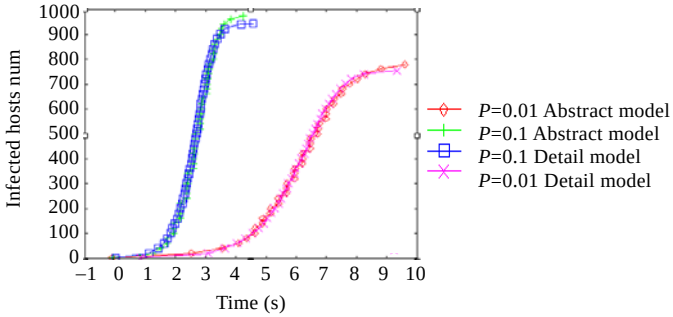


Fig.3 Simulation results with different network simulation models

图 3 不同网络模型下的仿真结果

3.2 Small World逻辑拓扑模型实现

我们在第 2.1 节中描述的 Watts 的基于无向图的 Small World 构建方法的基础上,进行基于有向图的 Small World 逻辑拓扑结构生成算法设计,具体过程如下:

(1) 将所有主机节点按照物理地址关系(也可以根据需要参照其他关系)生成逻辑编号,逻辑编号形成一个连续递增的序列,序列首尾连接形成一个环,元素个数为$|V|$,预计有向边数为 $Esum$.

(2) 从环上任选一个编号为 u 的节点(一般为计算方便,选择逻辑编号序列的起始节点)开始.确定 i=1,选择编号为 v_k=(u+i)mod($|V|$)的节点.Small World 模型随机概率为 P,以概率(1–P)生成边$\langle u,v_k\rangle$,同时,以概率 P_o 生成该边的对称反向边$\langle v_k,u\rangle$,保存这两条边信息.

(3) 若步骤(2)中的规则边生成没有成功,进行本步操作.在节点序列中随机选择一个节点 v_r,生成边$\langle u,v_r\rangle$,判断 v_r 和 u 是否相同,若相同,则表示边$\langle u,v_r\rangle$为回环,丢弃;同时查看边$\langle u,v_r\rangle$是否存在,若存在,则丢弃;否则,保存边$\langle u,v_r\rangle$信息,同时以概率 P_o 生成该边的对称反向边$\langle u,v_r\rangle$.

(4) 检查生成的边数是否等于 $Esum$,如果小于,则转向(5);如果等于,则停止生成过程,完成逻辑拓扑生成.

(5) 按顺时针方向,在节点序列中选择相邻的下一个节点重复步骤(2)、步骤(3)的操作,直到序列中所有节点都被处理,完成一轮操作.

(6) i=i+1,重复步骤(2)~步骤(5)的操作.

改变 P,P_o,$Esum$ 可以改变 Small World 模型的结构属性.逻辑拓扑结构信息生成后发送到相应的结点模型中保存,用于蠕虫数据包发送.

3.3 Small World模型特征路径和聚合系数计算分析

如前分析,Small World 模型的两个重要结构属性为:特征路径长度 L 和聚合系数 C.我们首先给出了 L 和 C 的算法,随后对模型生成参数和模型的这两个结构属性之间的关系进行分析.

特征路径长度 L 的计算:假定路径长度用源节点和目标节点之间经过的边数表示.参考 Dijkatra 最短路径算法,进行 L 计算的算法设计如下:

(1) 将第 3.2 节中生成的模型信息以 $Esum$×2 数组的形式保存,每行表示一条边,第 1 列表示该边起点信息,第 2 列表示该边终点信息;

(2) 从节点序列中任意一点 u 开始,遍历数组找出该节点的相邻接点集合 V_n 并保存,则 u 到这些节点的最短路径长度为 1,到节点 u 的最短长度为 2 的节点必然为 V_n 中所有节点的全部邻接点;

(3) 依次找出 V_n 中所有节点的全部邻接点.若新找到的邻接点 v_n 和节点 u 的最短路径已经找到,则丢弃 v_n;若没有找到,则记录节点 u 到 v_n 的最短距离为 2,保存该节点为新增待处理节点,顺序处理所有邻接点,找到所有与节点 u 最短距离为 2 的节点,同样,与节点 u 最短距离为 3 的节点必然为新增待处理节点的邻接点;

(4) 根据步骤(2)、步骤(3)所描述的算法,按递增顺序生成与节点 u 的最短距离为步骤(3)、步骤(4)等等的

信息,直到新增待处理节点数为 0,完成节点 u 与逻辑拓扑模型中其余节点之间最短距离的计算.与节点 u 没有最短距离信息的节点为不可达节点;

(5) 计算其余节点与 u 的所有最短距离的平均值;

(6) 依次对拓扑中的每个节点,重复步骤(2)~步骤(5);

(7) 将得到的平均值再求平均值,得到 L.

聚合系数 C 计算:按照第 2.1 节中聚合系数的定义,聚合系数算法定义如下:

(1) 从节点序列中的任意一点 u 开始,假定该节点所有邻接点之间存在的实际边数为 E_{actual}=0;

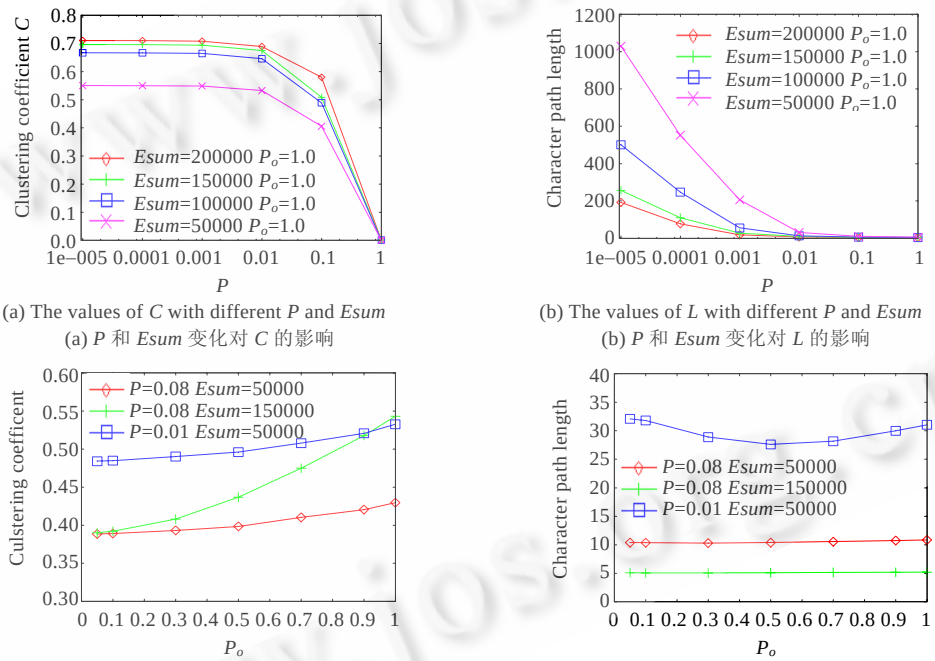
(2) 遍历 $Esum$×2 数组,找到节点 u 的所有 k_v 个邻接点,它们之间可能存在的理论边数 $E_{thero}=k_v\times(k_v-1)$;

(3) 依次处理与这些邻接点相连的所有边,若其目标节点为节点 u 的邻接点,则 $E_{actual}=E_{actual}+1$,计算出节点 u 的邻接点之间实际存在的边数 E_{actual};

(4) 计算 $C_v=E_{actual}/E_{thero}$;

(5) 依次对节点序列中的每个节点重复步骤(1)~步骤(4),计算聚合系数 $C=\Sigma C_v/|V|$.

图 4 显示了模型参数与 L 和 C 值的关系.从图 4(a)、图 4(b)可以看出,随着 P 值的增加,L 值迅速下降,而 C 值则开始缓慢下降,之后快速下降.所以,当 P 在 0~1 之间取值时,拓扑结构有较高的聚合系数 C 和较低的特征路径长度 L,与 e-mail 网络的测试结果相似.同时,随着 $Esum$ 的增加,C 增加,而 L 却显著下降.很明显,当 $Esum$ 足够多且网络为全连通结构时,有 C=1.0,L=1.当 $Esum$ 不变时,L,C 主要受 P 的影响.从图 4(c)、图 4(d)可以看出,P_o 变化对 L,C 的影响较小.



(a) The values of C with different P and $Esum$
(a) P 和 $Esum$ 变化对 C 的影响

(b) The values of L with different P and $Esum$
(b) P 和 $Esum$ 变化对 L 的影响

(c) The values of C with different P_o
(c) P_o 变化对 C 的影响

(d) The values of L with different P_o
(d) P_o 变化对 L 的影响

Fig.4 The values of C and L with different parameters

图 4 不同模型参数下的 C 和 L 值

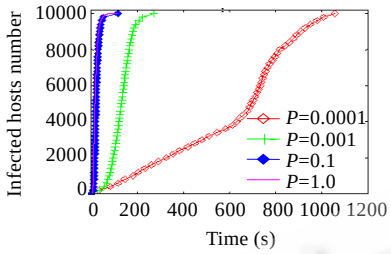
4 拓扑相关蠕虫传播仿真实验结果分析

数据包级蠕虫仿真可以比较方便地考虑物理网络因素,包括网络带宽、网络流量和网络延迟等对蠕虫传播的影响,并在文献[14–16]中有较为详细的描述,所以,本文主要讨论逻辑拓扑结构及蠕虫启动方式对蠕虫传播的

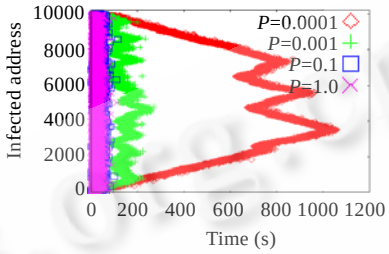
影响.

4.1 逻辑拓扑结构对蠕虫传播的影响

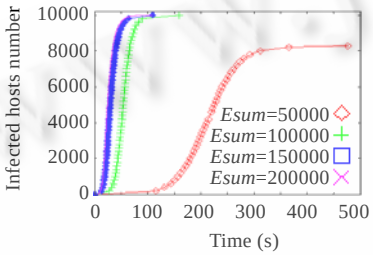
拓扑模型参数影响拓扑结构特性,进而可以影响蠕虫的传播过程.本节内容主要考察模型参数对蠕虫传播的影响.我们改变逻辑拓扑模型参数,进行蠕虫仿真实验.仿真平台采用第 3.1 节中描述的平台.实验规模为 10 000 节点,启动方式为人工 Non-Reinfection 方式,Social Engineering 信息获取时间指数分布,λ=1/10,蠕虫代码最终成功启动概率 P_{active}~N(0.5,0.09).为了验证 P 对传播特性的影响,选择 $Esum$=200000,P_o=1.0,P 分别取 0.0001,0.001,0.1,1.0 进行仿真实验.为了验证 $Esum$ 对传播特性的影响,选择 P=0.08,P_o=1.0,$Esum$ 分别取 50 000,100 000,150 000,200 000 进行仿真实验.为了验证 P_o 对传播特性的影响,选择 P=0.08,$Esum$=150000,P_o 分别取 0.0,0.05,0.7,1.0 进行仿真实验,实验结果如图 5 所示.



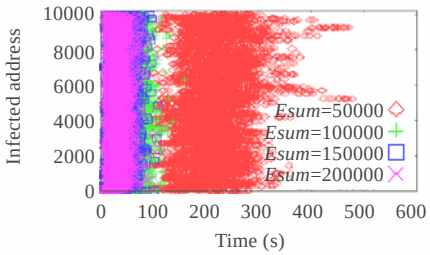
(a) Impact of P on the change of infected hosts
(a) P 对 infected hosts 变化的影响



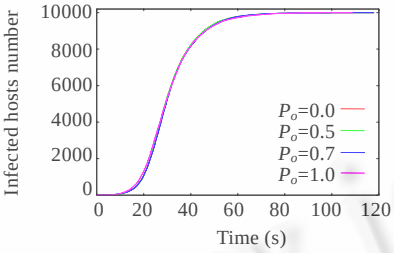
(b) Impact of P on the change of infected addr
(b) P 对 infected addr 变化的影响



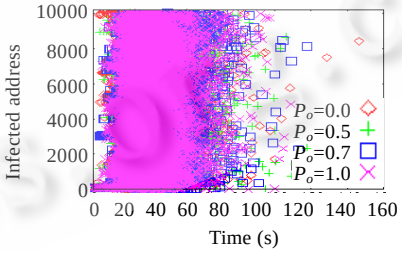
(c) Impact of $Esum$ on the change of infected hosts
(c) $Esum$ 对 infected hosts 变化的影响



(d) Impact of $Esum$ on the change of infected address
(d) $Esum$ 对 infected address 变化的影响



(e) Impact of P_o on the change of infected hosts
(e) P_o 对 infected hosts 变化的影响



(f) Impact of P_o on the change of infected address
(f) P_o 对 infected address 变化的影响

Fig.5 Worm propagation with different topology parameters

图 5 不同参数下的蠕虫传播仿真结果

从图 5(a)、图 5(b)中可以看出,随着 P 由小到大发生变化,蠕虫传播速度逐渐增加,感染地址变化从规律性向随机性转变.图 5(c)、图 5(d)显示,随着 $Esum$ 从小到大发生变化,蠕虫传播速度逐渐增加,感染地址变化规律的随机性也逐渐增加,但是变化幅度明显小于 P 变化对蠕虫传播的影响.图 5(e)、图 5(f)显示,P_o 变化对蠕虫传播的影响较小.由第 3.3 节分析可知,P 增加,L 和 C 减小;$Esum$ 增加,L 减小,而 C 则少量增加.而 P_o 变化对 L 和 C

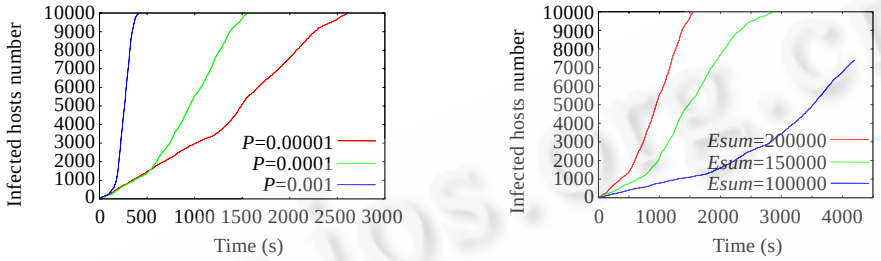
的影响较小.综合图 5 的实验结果,我们可以认为:随着 L 的减小,蠕虫传播速度增加;随着 C 值的减小,感染节点地址的随机性增加.

Esum 增加,每个节点感染后发送的蠕虫数据包数量增加,蠕虫传播速度加快.而在 *Esum* 较小的情况下,通过增加 P、减小 L 也可以增加蠕虫传播的速度.为了分析 *Esum* 和 L 对蠕虫传播的影响,我们进行对比实验:

① 设定 *Esum*=200000,P_o=1.0,变换 P 值进行仿真实验;

② 设定 P=0.01,P_o=1.0,变换 *Esum* 值进行实验,结果如图 6 所示.

从图 6(a)中可以看出,蠕虫传播明显地分为两个阶段:线性增长阶段和指数增长阶段,随着 P 的增加, L 减小,两个阶段的过渡点明显提前,这与 Newman 等人对 Small World "percolation"问题的分析相一致[23].图 6(b)显示,随着 *Esum* 的增加,蠕虫线性增长阶段的速度明显增加,过渡点也明显提前,这是因为在其他参数不变的条件下,*Esum* 增加,L 也会减小,所以过渡点提前.可见,L 通过改变蠕虫传播的增长方式影响蠕虫传播速度;*Esum* 通过增加线性初始增长阶段的速度提高蠕虫传播的速度.



(a) Worm propagation with different L (b) Worm propagation with different *Esum*

(a) 不同 L 下的蠕虫传播 (b) 不同 *Esum* 下的蠕虫传播

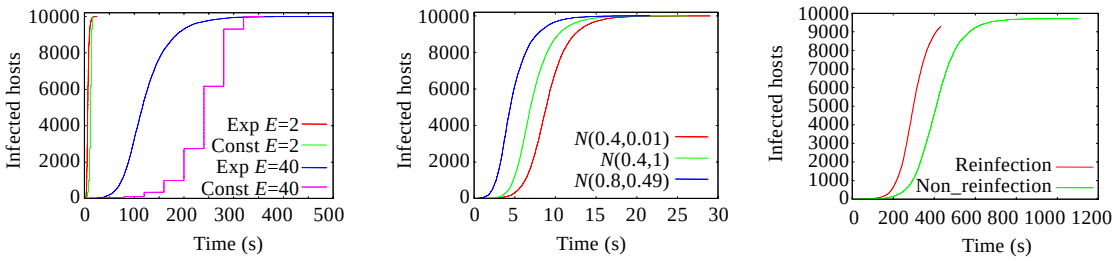
Fig.6 The results of comparison experiments

图 6 对比实验结果

4.2 蠕虫代码启动对蠕虫传播的影响

下面分析蠕虫代码启动方式对蠕虫传播的影响.实验规模为 10 000 节点,P=0.08,*Esum*=150000,P_o=1.0.为了考察 Social Engineering 信息获取时间 T 对蠕虫传播的影响,调整λ分别为 1/2,1/40 进行仿真实验.文献[17]指出:Social Engineering 信息获取时间平均值相同,方差越大,蠕虫传播速度越快.为了验证该结论,设定 Social Engineering 信息获取时间分别取常数 2,40 进行仿真对比.为了考察 P_{active} 对蠕虫传播的影响,分别取 P_{active}~N(0.8,0.49),P_{active}~N(0.4,1),P_{active}~N(0.4,0.01)进行仿真实验.最后,为了考察 Non-Reinfection 和 Reinfection 启动方式对蠕虫传播的影响,分别在两种启动方式下进行仿真实验.实验结果如图 7 所示.

从图 7(a)可以看出,Social Engineering 信息获取时间 T 越长,蠕虫传播的每一个环节耗时越长,因而整体传播速度下降.在均值相同的条件下,T 分布方差越大,蠕虫传播越快.根据文献[24],蠕虫感染初始阶段,感染的节点越多,蠕虫传播的整体速度越快.T 分布方差越大,其在平均时间前、后被感染的节点数量就相应增加,提前被感染的节点增加如同增加了初始被感染节点的数量,所以有更高的传播速度.图 7(b)显示,P_{active} 均值减少,可以明显地降低蠕虫传播速度,与我们的理解相符.但是,在 P_{active} 均值较低的情况下,减小 P_{active} 分布方差可以进一步降低蠕虫传播速度.这说明:一方面,只有增加组织用户的整体安全意识才能最大限度地限制蠕虫传播过程;另一方面,仅用平均隔离效果评价拓扑相关蠕虫防御措施的有效性存在很大的局限性,只有在更多的节点上实施有效的蠕虫防御措施,才能获得最佳总体效果.图 7(c)显示,Reinfection 启动方式相对于 Non-Reinfection 启动方式有较高的蠕虫传播速度,但增加效果不明显.因此说,当每个节点的感染概率很高时,Reinfection 启动方式几乎没有任何改善效果,反而降低了蠕虫传播的隐蔽性.



(a) Impact of T on the worm propagation
(a) T 对蠕虫传播的影响

(b) Impact of P_{active} on the worm propagation
(b) P_{active} 对蠕虫传播的影响

(c) Reinfection vs. non-infection
(c) Reinfection 和 Non-reinfection 启动方式对比

Fig.7 The impact of worm activation model on worm propagation
图 7 蠕虫代码启动模型对蠕虫传播的影响

5 结束语

本文在全面分析目前蠕虫仿真模型相关研究的基础上提出了一个完整的拓扑相关蠕虫传播仿真模型,并在 NS2 的基础上,通过选择性抽象进行了数据包级拓扑相关仿真系统实现,最后通过该仿真系统分析了逻辑拓扑结构和蠕虫代码启动方式对蠕虫传播的影响.本文的主要工作和创新点是:①从漏洞主机发现、蠕虫代码传播、蠕虫代码启动 3 个方面,提出了一个完整的数据包级拓扑相关蠕虫仿真模型;② 基于 Watts 的 Small World 模型,提出了基于有向图的 Small World 拓扑结构生成算法,并在数据包级蠕虫仿真平台上进行了实现,分析了模型参数和结构属性之间的关系;③ 在通用网络仿真器 NS2 的基础上,通过选择性抽象,克服规模限制瓶颈,实现了一个完整的数据包级仿真系统;④ 通过该仿真系统,实验分析了逻辑拓扑结构和蠕虫代码启动方式对蠕虫传播的影响.本仿真系统为拓扑相关蠕虫研究提供了一个有效的仿真实验平台.

References:

[1] Eichin M, Rochlis J. With microscope and tweezers: An analysis of the Internet virus of November 1988. In: Proc. of the IEEE'89 Computer Society Symp. on Security and Privacy. Washington: Computer Society Press of the IEEE, 1989. 326−343. http://csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePath=/dl/proceedings/&toc=comp/proceedings/sp/1989/1939/00/1939toc.xml

[2] CERT. CERT advisory CA-1999-04 melissa macro virus. 1999. http://www.cert.org/advisories/ca-1999-04.html

[3] CERT. CERT advisory CA-2000-04 love letter worm. 2000. http://www.cert.org/advisories/CA-2000-04.html

[4] Qing SH, Wang C, He JB, Li DZ. Research and development of instant messaging worms. Journal of Software, 2006,17(10): 2118−2130 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/2118.htm

[5] The W32/Sobig.E@MM virus spreads steadily; Forges its "from:" field. http://virusbusters.itcs.umich.edu//sobig-e.html

[6] Wong C, Bielski S, McCune JM, Wang CX. A study of mass-mailing worms. In: Proc. of the 2004 ACM Workshop on Rapid Malcode. New York: Publications Dept., ACM, Inc., 2004. 1−10. http://portal.acm.org/citation.cfm?id=1029618

[7] Jiang T, Kim W, Lhee J, Hong M. E-mail worm dectection using the analysis of behavior. In: Proc. of the 2nd Int'l Conf. on Distributed Computing & Internet Technology. LNCS 3816, Berlin, Heidelberg: Springer-Verlag, 2005. 348−356. http://www.springerlink.com/content/jr9435t76537/?p=be5b723ce25946d288ea6ee0667cb743&pi=0

[8] Newman MEJ, Forrest S, Balthrop J. Email networks and the spread of computer viruses. Physical Review E, 2002,66(3).

[9] Watts D, Strogatz S. Collective dynamic of Small World networks. Nature, 1998,393(4):440−442.

[10] Zou CC, Gong W, Towsley D. Code red worm propagation modeling and analysis. In: Atluri V, ed. Proc. of the 9th ACM Symp. on Computer and Communication Security. New York: Publications Dept., ACM, Inc., 2002. 138−147.

[11] Chen Z, Gao L, Kwiat K. Modeling the spread of active worms. In: Proc. of the 22nd Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2003), Vol.3. Washington: Computer Society Press of the IEEE, 2003. 1890−1900. http://ieeexplore.ieee.org/xpl/RecentCon.jsp?punumber=8585&conhome=8585

[12] Nicol D. The impact of stochastic variance on worm propagation and detection. In: Proc. of the 2006 ACM Workshop on Rapid Malcode. New York: Publications Dept., ACM, Inc, 2006. 57−63. http://portal.acm.org/citation.cfm?id=1103626

[13] Ebel H, Mielsch LI, Bornholdt S. Scale-Free topology of e-mail networks. Physical Review E, 2002,66(3).

[14] Riley GF, Sharif MI, Lee W. Simulating Internet worms. In: Proc. of the 12th IEEE/ACM Int'l Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems. Washington: Computer Society Press of the IEEE, 2004. 268−274. http://ieeexplore.ieee.org/xpl/RecentCon.jsp?punumber=9336&conhome=8783

[15] Liljenstam M, Yuan Y, Premore B, Nicol D. A mixed abstraction level simulation model of large-scale Internet worm infestations. In: Proc. of the 10th IEEE Int'l Symp. on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems. Washington: Computer Society Press of the IEEE, 2002. 109−116.

[16] Perumalla KS, Sundaragopalan S. High-Fidelity modeling of computer network worms. In: Proc. of the 20th Annual Computer Security Applications Conf. Washington: Computer Society Press of the IEEE, 2004. 126−135. http://ieeexplore.ieee.org/xpl/RecentCon.jsp?punumber=9473

[17] Zou CC, Towsley D, Gong W. Email virus propagation modeling and analysis. Technical Report, TR-03-CSE-04, 2003. http://www.ists.dartmouth.edu/library/258.pdf

[18] Weaver N, Paxson V, Staniford S, Cunningham R. A taxonomy of computer worms. In: Proc of the 2003 ACM Workshop on Rapid Malcode. New York: Publications Dept., ACM, Inc., 2003. 11−18. http://portal.acm.org/citation.cfm?id=948187

[19] Smith RD. Instant messaging as a scale-free network. 2006. http://arxiv.org/abs/cond-mat/0206378

[20] W32.Serflog.A@mm. http://securityresponse.symantec.com/avcenter/venc/data/w32.serflog.a@mm.html

[21] W32.Aplore@mm. http://securityresponse.symantec.com/avcenter/venc/data/w32.aplore@mm.html

[22] W32.AimVen@mm. http://securityresponse.symantec.com/avcenter/venc/data/w32.aimven.worm.html

[23] Newman MEJ, Watts DJ. Scaling and percolation in the Small World network model. Physical Review E, 2002,60(6):7332−7342.

[24] Staniford S, Paxson V, Weaver N. How to own the Internet in your spare time. In: Boneh D, ed. Proc. of the USENIX Security Symp. 2002. Berkeley: The USENIX Association, 2002. 149−167.

附中文参考文献:

[4] 卿斯汉,王超,何建波,李大治.即时通信蠕虫研究与发展.软件学报,2006,17(10):2118−2130. http://www.jos.org.cn/1000-9825/17/2118.htm

王跃武(1975－),男,河南汝阳人,博士生,主要研究领域为网络安全技术,大规模网络蠕虫仿真.

向继(1976－),男,博士生,主要研究领域为网络安全技术,网络蠕虫仿真.

荆继武(1964－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为网络与系统安全技术,PKI 技术,入侵容忍技术,蠕虫仿真技术.

刘琦(1978－),女,博士生,主要研究领域为网络安全技术,网络蠕虫仿真.