基于节点自杀的对等网络蠕虫防治方法

吴国政，秦志光

(电子科技大学计算机科学与工程学院 成都 611731)

【摘要】结合P2P网络的特点，提出了基于节点自杀的P2P蠕虫防治方法。基于主动探测P2P蠕虫检测方法是一套分布式系统，该系统依赖于分散在P2P网络中的探测节点来收集和分析P2P蠕虫流量，从而建立P2P蠕虫的流量特征，并用于对P2P蠕虫的识别。基于节点自杀的P2P蠕虫防治方法首先采用节点污染技术，将具备蠕虫检测和防御能力的节点分散在P2P蠕虫传播网络中。当这些具备检测功能的节点发现可疑的蠕虫传播流量时，采用自杀方式退出P2P网络，并将相关信息向自己的邻居节点进行通告，从而可有效遏制P2P蠕虫的快速传播。

关 键 词 对等网络蠕虫； 对等网络蠕虫防治； 对等网络蠕虫传播； 蠕虫

中图分类号 TP391 **文献标识码** A doi:10.3969/j.issn.1001-0548.2012.01.024

Suicide Based P2P Worms Defensive Approach

WU Guo-Zheng and QIN Zhi-Guang

(School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 611731)

Abstract A suicide based defensive approach is proposed in this paper. In this novel defensive solution, a set of defensive nodes with capacity of detecting P2P worms will be deployed into a P2P network. While P2P worms break out, these defensive nodes will collect P2P worm data to construct a detection model to prevent the spreading of P2P worms. First, these defensive nodes will suicide to avoid being infected and exploited after P2P worm breaking out. Furthermore, each suicide node will also send a warning message to its neighbors to take preventive actions.

Key words P2P worms; P2P worm defensive algorithm; P2P worm propagation; worms

从现有P2P蠕虫防治方法看，不论是基于单机的还是基于节点协作的主动式防御方法，都依赖于识别P2P蠕虫的准确性。然而，在蠕虫尚未爆发前，由于不知道蠕虫特征，往往导致识别具有很大的滞后性，严重制约了主动式P2P蠕虫防治方法的实用性。被动式P2P蠕虫防御方法通过对蠕虫的传播网络等进行深入分析，以期把握其宏观规律，并应用于蠕虫防治。但是，该方式需要借助分布于整个互联网的安全协作组件，应用成本高，很难大规模部署。反应式P2P蠕虫防御方法通过对蠕虫的行为特征进行分析，进而依赖这些行为特征识别和防治蠕虫。但是，行为分析方法存在主动式防御方法同样的问题，即在蠕虫尚未爆发前很难知道其行为特征。

总之，从现有P2P蠕虫防御技术看，尚未找到行之有效的方法，相关工作也较少，因而对其进行深入研究具有很强的前瞻性和实用性。鉴于此，本文提出了基于节点自杀的P2P蠕虫防治方法。节点自杀方法借助于节点污染技术，在P2P网络中插入一定数量的防御节点，当这些防御节点检测到蠕虫传播后，以自杀方式暂时退出P2P网络，从而达到阻断蠕虫传播路径的目的，进而抑制蠕虫的传播。

1 相关研究工作

文献[1]研究了随机扫描和基于本地信息扫描两种流行的蠕虫攻击策略，并且阐述了一种具有区域概念的反应式防御体系，即基于对等网络系统的互联网主动蠕虫防御体系(PADE)。该体系的核心是利用对等计算提供的大量资源，高效快速地检测攻击并免疫。该防御体系将P2P网络划分为若干防御区域，每个防御区域包括普通防御节点和区域防御领导节点。普通防御节点仅进行本地蠕虫检测，并将异常情况报告给防御领导节点，执行来自领导节点

收稿日期：2010 – 04 – 15； 修回日期：2010 – 08 – 27
基金项目：国家863计划(2009AA01Z422)
作者简介：吴国政(1974 –)，男，博士，主要从事网络安全方面的研究.

的防御命令。领导节点通过普通防御节点发来的信息，判断发起何种防御响应，如通知其他管理区域内的防御节点在确定范围内进行蠕虫免疫。区域内或区域间可以进行防御合作，防御节点就能同时对已染节点和易染节点进行快速免疫，并最大限度地避免重复免疫。如果防御节点无法免疫，则会直接通知相关节点或通知响应的系统管理员。但是，该方法属于已知蠕虫特征的防御法，需要节点具备蠕虫检测能力，因此仅适合于已知蠕虫的防治。此外，该方法假设的属于不同“区域”的节点之间的协作，在实际网络中也很难实现。

文献[2]主要针对结构化对等网络中蠕虫的表现进行分析，并提出了对应的防御方法。该文献认为P2P网络领域中有广泛使用基于P2P的文件共享系统以及使用分布式哈希表(DHT) 来提高查询效率两种趋势；并研究如何使这些技术在对抗互联网蠕虫方面更加有力。该设计能够用来实现一些已有的关注于控制蠕虫传播的防御策略。

文献[3-5]提出了利用软件多样性延缓蠕虫传播的思想，认为绝大多数蠕虫都无法运行于相同的平台上，可以设计一种特殊的拓扑来隔离蠕虫。文献[3]提出使用不同的P2P软件客户端来避免蠕虫快速繁殖；文献[4]对Chord进行扩展，提出一种被称为Verme的协议，即将系统中的节点按照其弱点的类型进行分组，每一个组叫做一个Island，具有相同弱点的Islands彼此不会相邻；文献[5]提出了一个专门的di-jest系统帮助节点选择邻居，在建立连接前由di-jest服务器使用一定的评估方法计算两个节点是否具有足够的差异性，从而避免脆弱性相同的节点相邻，延缓蠕虫传播。文献[6-7]讨论了如何利用P2P在其通信结构上的优势进行Internet蠕虫的防治。文献[6]提出了一套基于节点间合作的本地防御蠕虫策略，可成功地避免假阳性(false positives)风险。该文献通过研究蠕虫在被系统请求激活方面的临时固定性和相似性作为关联特征，如果独立节点间行为存在此关联特征，则可检测出相关的快速传播蠕虫。文献[6]使用Windows XP(SP2)对该方法进行评估，并将9种不同蠕虫和10种商业软件，以及15种平台的本地进程混合，两个节点间通过交换它们相互间行为(由频繁的系统请求进行定义)的快照，就可以判断出是否在执行蠕虫，准确性达76%～97%；并且该检测方法避免了误判的可能性。

文献[8]认为P2P网络提供了非常适合蠕虫快速传播的环境，同样地该环境也适合Anti-worm的传播，从而提出了一种基于P2P的Anti-worm对抗策略，并且提出了TF-SEI模型来模拟worm和Anti-worm的繁殖。文献[8]假设系统中存在恶性蠕虫*A*和良性蠕虫*B*，在良性蠕虫*B*出现之前，蠕虫*A*的繁殖遵循TF模型，良性蠕虫*B*出现后杀死蠕虫*A*并且对已染节点打补丁使其免疫，此时蠕虫*A*的繁殖遵循SEI模型。通过与传统的Anti-worm进行对比，得出基于P2P的Anti-worm在控制蠕虫传播上有更好效果，即使Anti-worm的释放滞后于蠕虫的出现一段时间，也能控制蠕虫的繁殖。

2 基于节点自杀的对等网络蠕虫防治基本思想

P2P蠕虫可以得到快速传播的原因之一是利用了节点的邻居路由表，从而不需要进行节点探测即可获得传播源。因此，不论采用哪种传播策略，节点的邻居节点表都是P2P蠕虫进行有效传播的重要信息。此外，这些邻居表中的节点感染蠕虫后，一般也会通过自己的邻居节点表快速传播蠕虫代码。

如果能够减少邻居节点表中可感染蠕虫的节点数，或者在邻居节点感染蠕虫后能够阻止这些节点继续传播蠕虫代码，则P2P蠕虫的传播就可以得到有效遏制。

此外，在识别出P2P蠕虫代码的特征后，虽然可对传播蠕虫的节点进行隔离，达到有效遏制蠕虫传播的目的。但是，这种检测和防御方法依赖于对运行在网关位置的蠕虫隔离设备(如P2P蠕虫流量检测与过滤设备)。P2P网络具有分布式系统的特征，其节点遍布整个互联网。因此为了达到蠕虫防治的目的，必须大规模部署P2P蠕虫检测和隔离软件，这不仅要求各安全组织相互协作，也涉及高成本投入的问题。

另外一种蠕虫防御方式是将蠕虫隔离功能分散到P2P网络的节点中。如果节点*a*是P2P网络的一个节点，则该节点可以运行蠕虫检测和隔离软件，一旦节点*a*发现蠕虫爆发，可将节点自己的蠕虫流量抛弃，从而遏制蠕虫的传播。基于节点污染的蠕虫防御方法实际上就是该防御方法的一种变体。但是，简单的流量抛弃有时候可能无法阻止蠕虫的传播。例如，即便节点*a*抛弃了蠕虫流量，由于P2P网络以及P2P蠕虫网络均具有快速愈合的功能，因此蠕虫节点可以很快地找到其他节点转发或者传播自己的蠕

虫流量。

本文提出了基于节点自杀的P2P蠕虫检测与防御方法。在该方法中，首先通过节点污染技术，将具有蠕虫检测功能的节点插入到P2P网络中。一旦这些插入的节点检测到蠕虫爆发，采取自杀的方式强制自己退出P2P网络，并同时通告自己的邻居节点进行蠕虫防御。后者为了防止蠕虫感染，可暂时自杀。因此，基于自杀的P2P蠕虫防御方法可以快速遏制蠕虫传播。

3 基于节点自杀的蠕虫防治算法

基于节点自杀的蠕虫方法分为节点污染和节点自杀两个阶段。

3.1 节点污染

节点污染是指为了检测和防御蠕虫攻击，将某些具备P2P蠕虫流量采集、分析和识别能力的节点通过某种非正规方式加入到P2P网络中，成为该网络的正式成员。

以BitTorrent网络为例，运行BItTorrent客户端软件的节点只需要向Tracker服务器注册热门现在资源的种子信息，就可以成为该BitTorrent网络的节点。为了尽可能多地部署这种节点，有两种节点污染技术：虚拟机技术和节点虚拟技术。

基于虚拟机的节点污染技术利用虚拟机技术，在一台物理计算机(一般采用高性能服务器)上运行多个P2P客户端软件，并通过网络地址虚拟或网络地址转换技术，在一台物理计算机上构造出多个具有独立功能的P2P节点。这些节点由于具有P2P客户端的功能，因此可以加入到网络中，进行消息接收和转发。

为了提高节点污染的规模，也可以对客户端功能进行剪裁，仅保留P2P客户端的基本功能，从而减少资源开销。但是，受制于单台计算机物理资源的局限性，基于虚拟机的节点污染技术很难实施大规模节点污染，因而可加入到网络的节点数一般有限。

3.2 节点自杀

一旦节点污染成功，P2P网络中就存在一定数量的，具备P2P蠕虫流量采集、分析和识别的防御节点。一旦P2P蠕虫爆发，这些防御节点可依据侦测到的P2P蠕虫信息，采取自杀方式防止蠕虫继续传播。最简单的节点自杀算法如下：

```
while(true)
{
侦测P2P网络
if(有P2P蠕虫传播)
{
通告邻居节点
随机睡眠一段时间
}
}
```

首先，节点除了提供必要的P2P应用程序的功能之外，还必须侦测P2P网络，以明确是否有蠕虫传播。一旦检测到蠕虫传播，则首先通告自己的邻居节点，以警示这些节点采取相应的防御措施；随后，节点以自杀方式随机睡眠一段时间，从而避开蠕虫的感染和传播。

节点的邻居节点接收到蠕虫爆发的通告信息之后，可以采取同样的自杀方式，以避免自己被蠕虫感染。当然，这些节点也可以将自己的自杀消息扩散到自己的邻居节点，因而自杀方式可以快速扩散到整个网络中。为了避免大规模节点自杀导致整个网络瘫痪，可以限制扩散的深度。在实际应用中，一般只需要扩散一层即可，即节点的自杀通告仅发送到自己的邻居节点。节点接收到蠕虫爆发通告后的处理算法为：

```
while(true)
{
侦测P2P蠕虫爆发通告
if (收到P2P蠕虫爆发通告)
{
随机睡眠一段时间
}
}
```

简单节点自杀算法的缺点是非常明显的，即节点可以发布恶意自杀信息，破坏整个P2P网络的正常通信。为了防治节点发布恶意自杀信息，需要采取安全防范措施。在进行P2P蠕虫防御时，需要部署一个中心服务器提供P2P蠕虫特征信息，且为每个通过节点污染方法加入到P2P网络的节点生成一对密钥，公钥(k_x)和私钥(k'_x)，其中公钥k_x的有效性由服务器签名保证。基于公钥-私钥对的节点自杀算法为：

连接中心服务器获得公钥k_p、私钥k'_p及公钥签名sig(k_p)

```
while (true)
{
侦测P2P网络
```

if (有P2P蠕虫传播)
{
$m=k_x'$ {通告信息}
将m、k_p以及sig(k_p)发送给自己的邻居节点
随机睡眠一段时间
}
}

$k\{m\}$ 表示用密钥k加密消息m，sig(k)表示服务器对密钥k的签名。当节点发布自杀通告时，采用自己的私钥对通告消息进行签名，并将签名消息连同自己的公钥及服务器对自己公钥的签名发送给自己的邻居节点。由于私钥的保密性，只有知道私钥的节点本身可以对消息进行签名。如果节点想伪造自杀通告来破坏正常通信，则接收到通告消息的节点可以通过检查其公钥的有效性，从而发现恶意节点。节点收到基于公钥的节点自杀通告消息后的响应算法为：

while(true)
{
侦测P2P蠕虫爆发通告
if (接收到P2P蠕虫爆发通告消息)
{
向服务器发送sig(k_p)及k_p，查询公钥k_p的可信性
if (k_p可信)
{
用k_p解密消息m，得到蠕虫爆发通告消息
随机睡眠一段时间
}
}
}

首先，节点将收到的公钥验证信息sig(k_p)以及公钥k_p发送给服务器，以验证公钥可信性。如果公钥k_p可信，则解密自杀通告，自己也将以自杀方式阻止蠕虫的传播；如果公钥不可信，该通告消息将被忽略。服务器签名保证了只有可信节点才能够发送自杀通告，可以防治恶意节点发送伪造消息破坏P2P网络的正常通信。由于需向服务器验证公钥可信性，因此会带来一定的流量开销和时间延迟。此外，由于服务器在整个验证过程中的重要性，因此可能会成为整个P2P蠕虫防御系统的瓶颈。

4 仿真分析

4.1 仿真流程处理

一般来说，节点接收到蠕虫消息即被传染，转换为感染节点。在基于自杀的防御方法中，不同状态的节点在收到蠕虫消息时有不同的处理方式。

1) 感染节点接收到蠕虫消息后不处理。

2) 如果当前防御节点数量未达到设置的上限，一般节点和免疫节点以一定概率转换成防御节点。如果转换成防御节点，则通告所有邻居有蠕虫(通过发送蠕虫通告消息，消息中的hops指通告消息继续传播的跳数，最初消息的hops为传播范围减1)，节点睡眠0～sleepTime个周期。如果没有转换成防御节点或者防御节点的数量已经达到上限，一般节点若为可感染版本则转换成感染节点，否则不变；免疫节点继续保持免疫状态。

3) 防御节点再次接收到蠕虫消息时，通告所有邻居蠕虫(发送蠕虫通告消息)，节点睡眠0～sleepTime个周期。

不同状态的客户节点接收到蠕虫通告消息的处理方式一致，如果通告消息的hops>0，节点转发通告消息(通告消息的hops减1)，节点睡眠0～sleepTime个周期。除了上述的处理流程外，仿真流程每个周期从一般客户节点、感染节点及免疫节点中选择一定数量的节点转换成免疫节点(免疫节点可重复被选中)。

4.2 仿真实验参数

仿真分析中，仿真规模最大为10 000个节点，仿真周期为1 000 周期(cycle)。蠕虫贪婪程度为1/cycle，即每周期从节点的邻居列表里随机选取一个节点进行感染。免疫率为50/cycle，即每周期在系统中随机选取一些节点作为免疫节点，上个周期免疫的节点本周期仍然可能选中，虽然状态不变，但是占用一个免疫节点数，免疫率从第49周期开始加入实验。节点邻居列表上限为50，节点请求更新邻居列表间隔为20 cycle，邻居列表满后每次只覆盖邻居列表中的2个邻居。客户节点启动周期范围为50，客户节点在周期0～50内随机启动。初始感染节点个数为1，感染节点启动周期为49，感染节点在第49个周期启动运行，此时所有客户节点已经启动。可感染客户节点的比例为0.8。

仿真分析考查与防御相关的几个参数对实验结果的改变情况。参数包括：

1) 防御节点比例上限(de)，即防御节点占仿真规模的比例上限，一旦防御节点数量到达上限，其他节点不再转换成防御节点；

2) 防御节点转换概率(be)，即一般节点和免疫

节点转换成防御节点的概率；

3) 随机睡眠周期上限(s)，即防御节点以及接收到通告消息的节点的睡眠周期数，从0～sleepTime。

4.3　防御节点比例上限对蠕虫传播的影响

仿真实验中，防御节点转换概率取0.4，随机睡眠周期上限为4 cycles，通告消息的转发范围设定为1跳。仿真结果如图1所示。

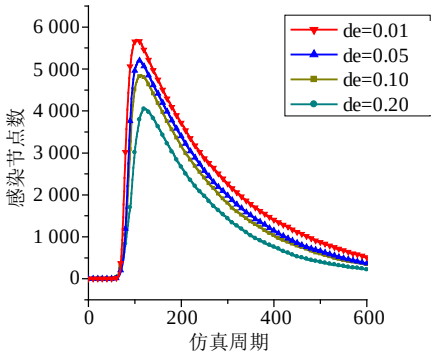

图1　防御节点比例对蠕虫传播的影响

(be=0.4，*s*=4 cycle)

从图1可以看出，防御节点比例上限越大，传播能到达的波峰越低，到达波峰的周期越晚，但不太明显。

4.4　防御节点转换概率对蠕虫传播的影响

仿真分析中，防御节点比例上限设为0.20，随机睡眠周期上限为8 cycle，通告消息的转发范围限定为1跳时，对应的防御节点转换率对蠕虫传播的影响如图2所示。防御节点比例上限设为0.10，随机睡眠周期上限为4 cycle，通告消息的转发范围限定为1跳时，节点防御转换率对蠕虫传播的影响如图3所示。

从图2和图3中可以看出，节点转换成防御节点的概率越大，传播能达到的峰值越低，到达波峰的周期也越晚。转变概率设置分别为0.1、0.2、0.4时，峰值的变化及到达波峰的周期变化比较缓慢。转换概率设置为0.8时，与之前的几个参数值有明显的不同。

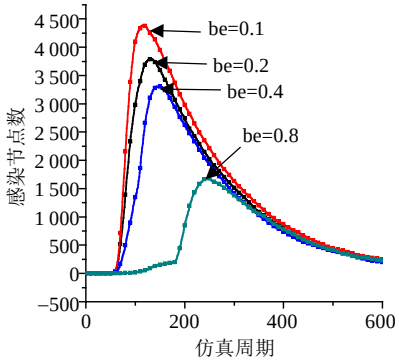

图2　防御节点转换率对蠕虫传播的影响

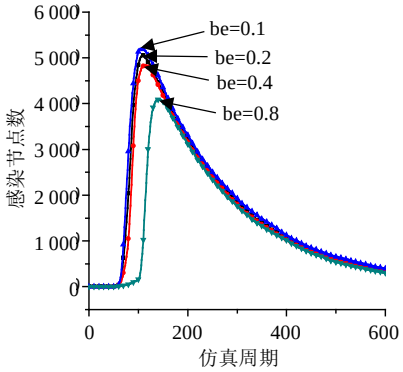

图3　节点防御转换率对蠕虫传播的影响

4.5　接收到通告消息后节点睡眠周期上限对蠕虫传播的影响

仿真实验中防御节点比例上限设定为0.20，防御节点转换概率为0.40，通告消息的转发范围限定为1时，节点睡眠周期对蠕虫传播的影响如图4所示。当防御节点比例上限为0.10，防御节点转换概率为0.10，通告消息的转发范围为1时，节点睡眠时间对蠕虫传播的影响如图5所示。

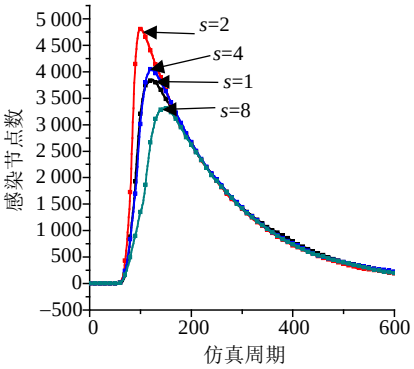

图4　节点睡眠周期对蠕虫传播的影响

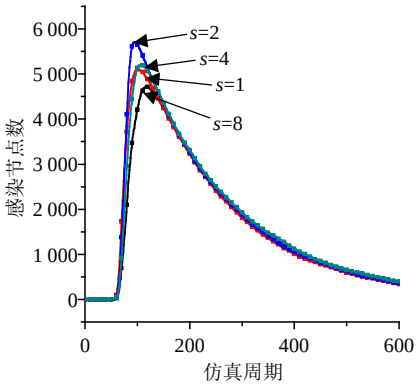

图5　节点睡眠时间对蠕虫传播的影响

从图中可以看出，睡眠周期上限从2增加到8的过程中，上限越大，曲线的增长越缓慢，峰值越低，波峰到达的周期也越晚。*s*=1 cycle时，节点随机睡眠周期范围为0～1 cycle，所以节点并没有睡眠。从

图中可以发现，节点不睡眠仿真的结果，与睡眠上限为4 cycle的仿真结果很接近。

5 小 结

P2P蠕虫的检测与防治是快速抑制蠕虫传播的重要方法。本文提出了基于自杀的P2P蠕虫防治方法。该方法首先采用节点污染的方法将部分具有蠕虫检测功能的节点加入到P2P网络中，一旦发现蠕虫爆发，不仅节点自己采用自杀方式退出P2P网络，也通过自己的邻居节点防御蠕虫，从而可阻止蠕虫的继续传播。仿真结果表明，基于节点自杀的防御机制，可有效减少P2P蠕虫的感染范围，从而遏制蠕虫的传播。

参 考 文 献

[1] WEI Yu, CHELLAPPAN S, XUN Wang, et al. On defending peer-to-peer system-based active worm attacks[C]//Proc of IEEE Global Telecommunications Conference (GLOBECOM). St.Louis , Missouri, USA: IEEE, 2005

[2] KANNAN J, LAKSHMINARAYANAN K. Implications of peer-to-peer networks on worm attacks and defenses [EB/OL]. [2010-01-11]. http://www.cs.berkeley. edu/～kubitron/courses/cs294-4-F03/projects/karthikjayanth.pdf, 2003.

[3] ZHOU Ying, WU Zhong-fu. Breaking monocultures in P2P networks for worm prevention[C]//Proceedings of the Fifth International Conference on Machine Learning and Cybernetics. Dalian: [s.n.], 2006.

[4] FREITAS C R P F, RODRIGUES R, RODRIGUES L. Verme: Worm containment in peer-to-peer overlays[C]// IPTPS'07: Proceeding of the 6th International Workshop on Peer-to-Peer Systems. [S.l.]: [s.n.], 2007.

[5] MCILWRAITH D, PQAUIER M. Di-jest: Autonomic neighbour management for worm resilience in p2p systems [C]//WoWMoM'08: International Symposium World of Wireless, Mobile and Multimedia Networks. [S.l.]: [s.n.], 2008.

[6] MALAN D J, SMITH M D. Host-based detection of worms through peer-to-peer cooperation[C]//Proceedings of the 2005 ACM Workshop on Rapid Malcode (WORM). Fairfax, VA, USA: ACM, 2005: 72-80.

[7] SHAKKOTTAI S, SRIKANT R. Peer to peer networks for defense against internet worms[C]//Proceedings from the 2006 Workshop on Interdisciplinary Systems Approach in Performance Evaluation and Design of Computer & Communications Systems. Pisa, Italy: [s.n.], 2006.

[8] YAO Yu, WU Li-qiong. A WAW model of P2P-based anti-worm[C]//Proceedings of the IEEE International Conference on Networking, Sensing and Control. [S.l.]: IEEE, 2008.

编 辑 税 红