利用接触跟踪机制实现Email蠕虫的检测

黄智勇[1]，曾孝平[1]，周建林[1]，石幸利[2]

(1. 重庆大学通信工程学院 重庆 沙平坝区 400044；2. 重庆科技学院学报编辑部 重庆 渝中区 401331)

【摘要】针对Email蠕虫逐渐成为一种主要的网络威胁，提出基于接触跟踪机制检测蠕虫的方法CTCBF。该方法利用“差分熵”对单个网络节点的异常连接行为进行检测，再通过异常节点之间的连接关系利用跟踪算法建立跟踪链，当跟踪链的长度达到设定阈值时，跟踪链上的可疑节点被确认为感染节点。针对阈值的不确定性，提出了一种动态阈值方法，根据不同的网络感染等级自适应调整阈值大小。仿真试验表明，该方法能够快速、准确地检测出蠕虫的传播行为，同时为未知蠕虫的检测提供了一种新的模式。

关 键 词 接触跟踪链； 检测； Email蠕虫； 熵； 仿真

中图分类号 TP393.08 **文献标识码** A doi:10.3969/j.issn.1001-0548.2011.03.021

Detecting Email Worm through Contact-Tracing Chain

HUANG Zhi-yong[1], ZENG Xiao-ping[1], ZHOU Jian-lin[1], and SHI Xing-li[2]

(1. College of Communication Engineering, Chongqing University Shapingba Chongqing 400044;
2. Journal of Chongqing University of Science & Technology Yuzhong Chongqing 401331)

Abstract Email worms have recently become the most serious security threat on the internet. In this paper, a contact-tracing chain based framework (CTCBF) is proposed to detect this worm through tracing the contact behaviors among peers. This framework uses the contact tracing chain to trace abnormal peers which are screened out by isolated monitoring, and develops “difference entropy” to group peers with the same abnormal behaviors. Peers are confirmed with infectious symptoms when the length of contact tracing chain which they belong to reaches the preset threshold. Through numerical simulations, we demonstrate that the proposed contact tracing framework can quickly detect the propagation of Email Worm.

Key words contact-tracing chain; detection; Email worm; entropy; simulation

近年来，Email蠕虫逐渐成为一种主要的网络攻击手段，各种Email蠕虫程序在网络中广泛传播[1]，如Melissa、Love Letter、W32/Sircam、SoBig、MyDoom、Bagle和Netsky等[2]。大量无用的Email数据存在于整个网络，直接导致网络数据传输阻塞，严重地影响了邮件系统网络的工作性能[3]，针对Email蠕虫的检测和控制成为研究热点之一。文献[4]介绍的“反馈防御系统”检测法(feedback defense system)，利用现有的入侵检测软件对可疑邮件进行拦截，再采用虚拟的蜜罐系统进行分析检测。该方法对Email用户有很好的保护作用，但是对于Email蠕虫在网络中转播的主动控制，效果不是很明显。文献[5]介绍的利用机器学习对网络异常流量进行监测方法，能够有效地减少检测误报率，但是该方法需要对网络流量进行统计，存在一定的检测延迟性。文献[6]提出利用熵值来归类垃圾邮件，即通过对垃圾邮件行为特征的分析(如在单位时间段里连续发送邮件的数量)，利用熵值的大小快速区分垃圾邮件和正常邮件。该归类方法的精度取决于阈值的大小，对于防御者来说，很难找到一个合适的阈值使检测结果同时达到最小的假阳性和最小的假阴性。针对这些问题，为了提高检测速度，并保证检测的精度，针对Email蠕虫的传播，本文提出了CTCBF 检测方法，该方法应用了传染病检测的接触跟踪机制[7-8]，通过建立跟踪链对异常的Email传播过程进行监控，并根据跟踪链的状态确认Email蠕虫感染节点。

1 CTCBF检测方法

1.1 检测系统基于假设

Email蠕虫最大的特点是能够利用Email的方式

收稿日期：2009 – 11 – 18； 修回日期：2010 – 07 – 12
基金项目：重庆市教委科学技术研究项目(KJ101403)
作者简介：黄智勇(1978 –)男，博士，主要从事网络信息安全方面的研究.

进行主动自我传播，本文的检测机制是基于蠕虫的这一行为特征进行研究的。系统考虑了两种行为特征——感染特征和连接特征。感染特征指被监控节点出现了异常的主动连接其他网络节点的行为(如在单位时间段里主动连接其他节点的数目超过规定的阈值)；连接特征指节点被感染节点或者可疑节点连接的行为。根据这两种特征，对网络中的节点做如下假设：

1) 一个节点如果是感染节点，那么它一定会再次感染其他节点。

2) 一个节点如果出现了感染特征，那么它可能已经是感染节点。

3) 一个节点如果出现了连接特征，那么它存在被其他节点感染的可能性，一旦它又出现感染特征，那么它也可能成为新的感染节点。

1.2 CTCBF算法

CTCBF算法由单点检测算法和多点跟踪算法两部分组成：1) 利用单点检测算法对单个节点的感染特征进行检测；2) 利用跟踪算法提高检测精度，在单个节点感染特征的基础上，通过分析节点之间的连接特征，从而确认真正的感染节点。

1.2.1 单点检测算法

文献[6]利用熵值大小区分垃圾邮件，本文在其研究的基础上进行改进，引入“差分熵”和“相似度”两个概念。定义 $V(T)=\langle v(1),v(2),\cdots,v(n)\rangle$ 和 $V'(T)=\langle v'(1),v'(2),\cdots,v'(n)\rangle$ 两个序列，其中 $V(T)$ 为被检测序列，$V'(T)$ 为超出设定阈值 M 部分的序列，$v(t)$ 为在一个时间段 t 内节点与外部节点建立的连接数，T 为序列的长度。分别对两个序列求熵值，然后将熵值相比较，两者的相似度越高，$V(T)$ 序列为异常连接的可能性就越高。根据香农定理[9]定义熵值为：

$$H(V)=-n\sum_{t=1}^{n}\frac{v(t)}{\max\left(\sum_{t=1}^{n}v(t),1\right)}\log_2\left(\frac{v(t)}{\max\left(\sum_{t=1}^{n}v(t),1\right)}\right) \tag{1}$$

$v'(t)$ 定义为：

$$v'(t)=\max(0,v(t)-M) \tag{2}$$

对序列 $V'(T)$ 求熵值得：

$$H(V')=-n\sum_{t=1}^{n}\frac{\max(0,v(t)-M)}{\max\left(\sum_{t=1}^{n}\max(0,v(t)-M),1\right)}\times\log_2\left(\frac{\max(0,v(t)-M)}{\max\left(\sum_{t=1}^{n}\max(0,v(t)-M),1\right)}\right) \tag{3}$$

“差分熵”定义为：

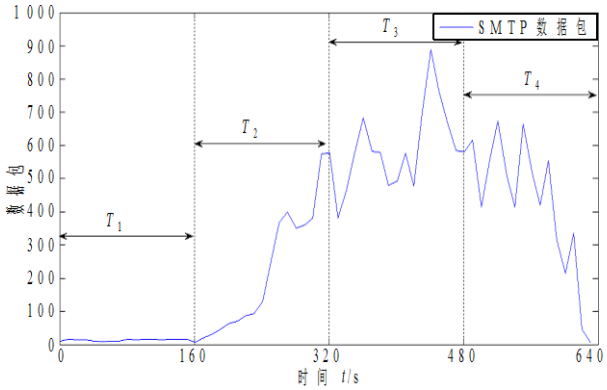
$$\mathrm{DH}=H(v)-H(v') \tag{4}$$

利用文献[10]的试验数据，本文进行仿真。如图1所示，整个观测周期划分为4个时间段：T_1、T_2、T_3、T_4，利用“差分熵”的定义分别对4个时间段的数据进行计算。分析仿真结果，得出以下结论：

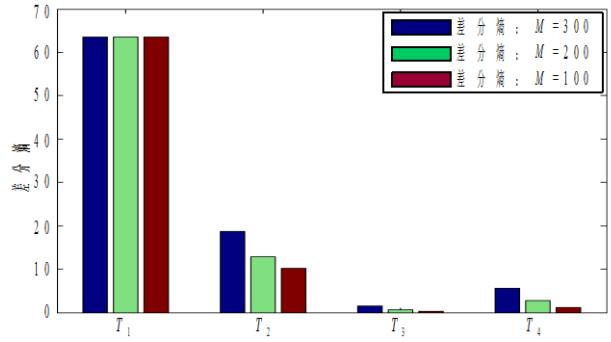
1) 当 $v(t)>>M$ 时，$V(t)$ 与 $V'(t)$ 的相似度较高，$\mathrm{DH}\to 0$。

2) 当 $v(t)<<M$ 时，$V(t)$ 与 $V'(t)$ 的相似度较低，DH 由正常连接的序列分布决定。

3) M 的大小影响“差分熵”的结果，M 越小，$V(t)$ 与 $V'(t)$ 的相似度越高，DH 也就越小。M 的值可以根据网络状况进行设置。由于跟踪算法能在单点检测的基础上提高检测精度，所以单点检测算法允许出现较大的假阳性，M 可以设置为一个较小的值。



a. 网络数据序列



b. 阈值对差分熵值的影响

图1 差分熵检测效果

1.2.2 多点跟踪算法

多点跟踪的目的是在假阳性较高的单点检测机制基础上，利用跟踪链提高检测精度。

定义 1 网络中的任意节点 $r\in S$ 都可能与其

他节点发生连接，并且成为任意跟踪链的根节点，所以每个节点分配 $S-1$ 跟踪链存储空间。

定义 2　网络中的任意节点 $i \in S, i \neq r$ 都可能成为以 r 为根节点的跟踪链节点。任意节点状态可表示为函数 $A_i = f(C_i(r), L_i(r), P_i(r)), i \in S$ 。

其中，$C_i(r)$ 为节点 i 的类型，$L_i(r)$ 为节点 i 相对于根节点 r 的级数，$P_i(r)$ 为节点 i 的父节点。

定义 3　根据网络节点的行为特征，将节点划分为正常类型(NS)；连接类型(CS)；可疑类型(SS)；感染类型(IS) 4个类型。

1) NS：没有出现感染特征和连接特征；

2) CS：出现连接特征，但没有出现感染特征；

3) SS：出现感染特征；

4) IS：出现过感染特征，且所在跟踪链被确认为感染链。

根据定义，算法初始化为：

for (all nodes $r, r \in S$) do
  for (all nodes $i, i \in S$) do
    $P_i(r) = i.\ C_i(r) = \text{NS}.$
    if $(i = r)$
      $L_i(r) = 0.$
    else
      $L_i(r) = -1.$
    end if.
  end for.
end for.

算法中，$L_i(r) = -1$ 表示节点 i 没有被任何以 r 为根节点的跟踪链节点感染；$L_i(r) = 0$ 表示每个节点相对于自己处于第0级；$P_i = i$ 表示将所有节点初始化为根节点，每个初始节点的类型均定义为NS。

当节点 j 出现感染特征，并且节点 j 和节点 i 出现连接特征，算法描述如下：

1) 以节点 j 为根节点建立新的跟踪链。

$$P_i(j) = j, C_i(j) = \text{CS}.$$

2) 检测所有与节点 j 和 i 相关的跟踪链节点，并更新相应的节点信息。

for (all nodes $r,\ r \in S$) do
  if $(P_i(r) \neq j$ and $L_i(r) = -1)$ then
    $L_j(r) = L_{P_j(r)}(r) + 1.$
    $P_i(r) = j.$
    $C_j(r) = \text{SS}.$
  end if.
end for.

3) 跟踪链确认。

if $(L_j(r) \geqslant K)$ then
  do $\{C_j(r) = \text{IS}.\ j = P_j(r).\}$
  while $(L_j(r) \neq 0$ and $C_j(r) = \text{SS})$
end if.

4) 被确认跟踪链的节点信息初始化。

while $(C_j(r) = \text{IS}$ and $j \neq r)$ do
  $\{C_j(r) = \text{NS}.\ L_j(r) = -1.\ j = P_j(r).\}$

① 节点之间建立跟踪链的联系必须存在因果关系，即一个NS类型的节点必须首先被其他节点感染以后才可能去感染另外的节点。

② 由于重复感染的存在，跟踪链上的节点类型一旦被确定为感染类型，所有节点信息应立刻被重新初始化。采用该算法可以发现更多的感染路径，加快了检测速度。

③ 阈值 K 的大小直接决定了系统的性能，K 值越大，跟踪链的误报率越低，但是检测速度会降低；K 值越小，检测的速度提高，系统的精度会降低。本文将介绍一种动态调节阈值 K 的方法来平衡检测精度和速度之间的关系。

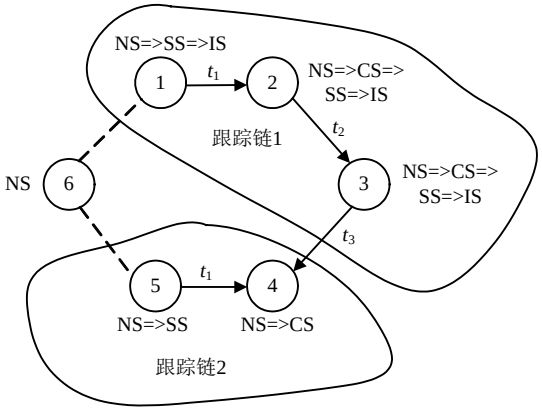


图2　跟踪链建立过程

图2描述了一个简单的跟踪链建立过程。设置 $K = 2$，$t_1 < t_2 < t_3$，初始状态下，每个节点均处于类型NS，图中箭头代表异常连接，虚线代表没有连接，L_i 代表节点 i 的级数。图中存在两条跟踪链，节点1、2、3构成链1，节点5、4构成链2。t_1 时刻，节点1和节点5出现感染特征，分别与节点2和节点4发生连接关系，将节点1和节点5作为两条跟踪链的根节点，则 $L_1 = L_5 = 0$，节点1和节点5的类型由NS变为SS，节点2和节点4的类型由NS变为CS；t_2 时刻，节点2出现感染特征，并且与节点3发生连接关系，节点2的类型由CS变为SS，$L_2 = 1$，同时节点3的类型由NS变为CS；t_3 时刻，节点3出现感染特征，连

接节点4，则 $L_3=2$，同时节点3的类型由CS变为SS，跟踪链1中的类型为SS的节点数目达到阈值 K，所以跟踪链1被确认为感染链，链上的节点被确定为感染节点，节点1、2、3的类型由SS变为IS。跟踪链2的长度由于没有达到阈值K,所以不能被确认为感染链。节点6为孤立节点，既没有出现连接特征也没有出现感染特征，所以节点类型保持为NS。

1.3 动态阈值

典型的蠕虫传播周期分为初始期、上升期、饱和期3个阶段。定义 ΔI 为单位时间段内增加的感染节点数目，用 ΔI 代表网络的感染等级。

1) 初始期：感染节点数目不多，增长速度不快，ΔI 比较小，网络感染等级低。

2) 上升期：感染节点数目逐渐增多，增长速度急剧加快，ΔI 迅速增大，网络感染等级增加。

3) 饱和期：感染节点数目的增加速度减慢，ΔI 逐渐减小，网络感染等级逐渐降低。

本文的策略是在网络感染等级较低时，采用较大的阈值 K 以提高跟踪链的精度，减少误报率；在网络感染等级较高时，采用较小的阈值 K 以提高跟踪链的速度，减少更多节点被感染的可能性。分别定义 $K_{\min}$ 和 $K_{\max}$ 为上限阈值和下限阈值，动态阈值 $K(t)$ 的算法为：

$$K(t+1)=\min\left(\max\left(K(t)\times\frac{\max(\Delta I(t),1)}{\max(\Delta I(t),1)},K_{\min}\right),K_{\max}\right) \tag{5}$$

上限阈值和下限阈值的设定限制了动态阈值参数的波动范围。区别于单点检测，需要满足 $K_{\min}\geqslant 1$，阈值 K 越大，检测精度会越高，但同时也降低了检测灵敏度。$K_{\max}$ 值不能设置过大，可以综合其他因素进行设定，如跟踪算法的效率、网络拓扑结构、蠕虫传播效率等。

2 试验仿真

本文应用C语言编写仿真程序来验证跟踪算法的性能。设定网络总节点数 $S=6\ 400$；网络中的节点可以描述为 $\{(A_i,P_{r_i},D_i),i\in S\}$，$A_i$ 表示节点 i 的状态；P_{r_i} 表示节点 i 被感染的概率，每个节点的感染概率不能确定，但是可以作如下假设：

1) 网络中节点数目足够多；

2) 节点具有分布性，且节点之间的连接行为相互独立。

基于以上两点假设，定义 P_{r_i} 服从高斯分布 $N(\mu_p,\sigma_p^2)$[11]，仿真试验中设置为 $N(0.2,0.04^2)$。D_i 代表节点 i 的度数，节点度数越高，感染其他节点的可能性越大。仿真实验将验证跟踪链的效率和跟踪链的鲁棒性。

2.1 跟踪链效率仿真

图3为在不同阈值 K 的情况下，跟踪链的跟踪效率的变化，阈值越大，跟踪链被确认需要的时间也越长。$K=11$时，跟踪链的检测速度明显慢于 $K=4$ 时的检测速度。采用动态阈值算法，由于仿真仅仅验证动态阈值算法在不同感染等级下对跟踪链的影响，所以，根据前文介绍的阈值设定原则，在有效范围内设置参数 $K_{\max}=10$，$K_{\min}=3$，初始状态下，$K(0)=K_{\max}$。仿真结果显示算法能够根据不同的感染等级 ΔI 调整动态阈值达到调整检测速度的目的。

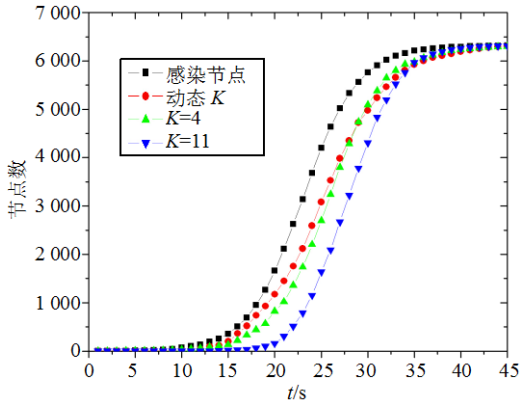


图3 不同的阈值K检测效果对比

2.2 跟踪链鲁棒性仿真

跟踪链的鲁棒性直接影响检测的效率，考虑影响跟踪链鲁棒性的两个因素：

1) 节点度数。节点度数 D 体现为蠕虫发送Email的目标地址数目，通常蠕虫不仅仅从被感染节点获取Email地址，也可以从网络收集得到。攻击者为了提高攻击效率，会收集更多的Email地址，而节点度数的提高更有利于跟踪链的建立。

2) 节点有效率。前面讨论的情况都基于所有网络节点能够参与跟踪链建立的假设，但实际情况并非如此。如一些节点没有安装检测软件，一些节点参与了跟踪链的建立，但是由于受到攻击(如DOS攻击)而失效，这类网络节点统称为失效节点。能够有效参与跟踪链建立的节点称为有效节点，定义 N_{enable} 为有效节点数目，定义节点有效率 $q=\dfrac{N_{\text{enable}}}{S}$，节点有效率越高，建立跟踪链的可能性就越大。定义检测率 $R=\dfrac{N_{\text{detected}}}{N_{\text{infected}}}$，其中 N_{detected} 为通过跟踪链确定的感

染节点数目，N_{infected}为被感染节点数目。

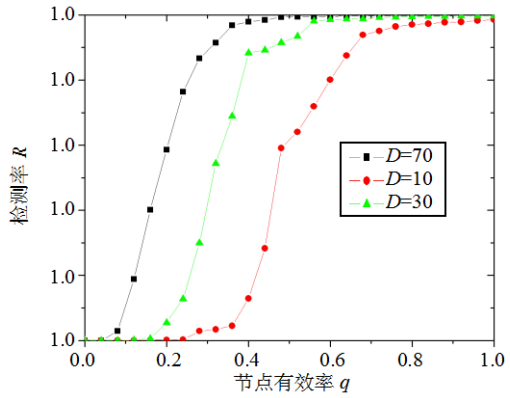


图4　跟踪链鲁棒性仿真

如图4所示，产生的3个随机网络平均节点度数分别为$D=70, 30, 10$。在范围[0,1]内逐步增加节点有效率q，分别运行跟踪算法。结果显示: 1) 当$D=70$时，即使存在大量的无效节点，算法仍然维持比较高的检测率R(当$q>0.3$时，$R>0.95$); 2) 节点度数减小，要维持高的检测率$R>0.95$，需要存在大量的有效节点，当$D=30$时，$q>0.5$，当$D=10$时，$q>0.7$。从攻击者的角度看，高的节点度数更有利于维持攻击网络的鲁棒性，同样，高的节点度数也更有利于跟踪链发现更多的感染节点。刻意地降低节点度数会降低跟踪链的鲁棒性，同时也会使攻击网络更容易被破坏。另外，减小失效节点的数量能够增强跟踪链的鲁棒性。

3　总结与展望

本文阐述了基于CTCBF机制检测Email蠕虫的方法，检测系统由单点检测和多点跟踪两部分组成。分别介绍了利用“差分熵”归类的单点检测算法和利用接触跟踪机制的多点跟踪算法。为了动态适应网络环境变化，采用了动态阈值算法。与单点检测机制相比较，多点跟踪机制通过传输跟踪链能够有效减小单点检测误差引起的误报率。通过仿真，认为CTCBF机制能够快速准确实现对Email蠕虫传播的检测。今后的研究工作还需要进一步提高单点检测算法和多点跟踪算法的检测效率，特别是在复杂多变的网络环境下能够保证算法的高效性。

参　考　文　献

[1] KHERA R.　Messaging anti-abuse working group[EB/OL]. [2009-09-25]. http://www.maawg.org.

[2] CERT/CC　advisories[EB/OL].　[2009-09-27].　http://www.cert.org/ advisories.

[3] SYMATEC. Internet security threat report trends Jan-June’ 07 [EB/OL]. [2009-08-02]. http://www.symantec.com.

[4] ZOU C, Gong W, TOWSLEY D. Feedback email worm defense system for enterprise networks[R]//Umass: ECE, 2004.

[5] GUPTA A, SEKAR R. An approach for detecting self-propagating Email using anomaly detection[C]// Proceedings of Recent Advances in Intrusion Detection. Pittsburgh PA: Springer, 2003: 55-72.

[6] HUSNA H, PHITHAKKITNUKOON S, DANTU R. Traffic shaping of spam botnets[C]//Proceedings of CCNC 2008, 5th IEEE. Las Vegas, NV: IEEE, 2008: 786-787.

[7] HYMANA J, LI Jia, STANLEY E. Modeling the impact of random screening and contact tracing in reducing the spread of HIV[J]. Mathematical Biosciences, 2003, 181: 1-16.

[8] EAMES K, KEELING J. Contact tracing and disease control[C]//Proceedings of the Royal Society. London: PubMed, 2003: 443-454.

[9] SHANON C. A mathematical theory of communication[J]. Bell System Technical Journal, 1948: 379-423.

[10] ZHANG Jun. Storm Worm & Botnet Analysis[EB/OL]. [2009-03-02]. http://www. securitylabs.websense.com/content.

[11] ZOU C, TOWSLEY D, GONG W. Modeling and simulation study of the propagation and defense of internet email worm[J]. IEEE Transactions on Dependable and Secure Computing, 2007, 4(2): 105-118.

编　辑　张　俊