doi:10.11959/j.issn.1000-436x.2017012

强抗毁性社交僵尸网络的构建及其防御

尹涛[1,2]，李世淙[3]，庹宇鹏[1,2]，张永铮[1,2]
（1. 中国科学院信息工程研究所，北京 100093；
2. 中国科学院大学，北京 100049；
3. 国家计算机应急技术处理协调中心，北京 100029）

摘 要：为打击僵尸网络，保障网络空间安全，提出一种新型的具备强抗毁性的社交僵尸网络（DR-SNbot），并给出了针对性的防御方法。DR-SNbot 基于社交网络搭建命令与控制服务器（C&C-Server, command and control server），每个 C&C-Server 对应一个不同的伪随机昵称，并利用信息隐藏技术将命令隐藏在日志中发布，进而提出一种新型的命令与控制信道。当 C&C-Server 不同比例地失效时，DR-SNbot 会发出不同等级的预警，通知攻击者构建新的 C&C-Server，并自动修复 C&C 通信以保障其强抗毁性。在实验环境中，即使当前 C&C-Server 全部失效，DR-SNbot 仍能在短期内修复 C&C 通信，将控制率维持在 100%。最后，基于伪随机僵尸昵称与合法昵称在词法特征上的差异性，提出一种僵尸昵称检测方法，可有效检测社交僵尸网络利用自定义算法批量生成的伪随机僵尸昵称。实验结果表明，该方法召回率达到 93%，准确率达到 96.88%。

关键词：网络安全；社交网络；僵尸网络；命令与控制信道；防御策略
中图分类号：TP393.08 **文献标识码**：A

Modeling and countermeasures of a social network-based botnet with strong destroy-resistance

YIN Tao[1,2], LI Shi-cong[3], TUO Yu-peng[1,2], ZHANG Yong-zheng[1,2]
(1. Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China;
2. University of Chinese Academy of Sciences, Beijing 100049, China;
3. National Computer Network Emergency Response Technical Team / Coordination Center of China, Beijing 100029, China)

Abstract: To defeat botnets and ensure cyberspace security, a novel social network-based botnet with strong destroy-resistance (DR-SNbot), as well as its corresponding countermeasure, was proposed. DR-SNbot constructed command and control servers (C&C-Servers) based on social network. Each C&C-Server corresponded to a unique pseudo-random nickname. The botmaster issues commanded by hiding them in diaries using information hiding techniques, and then a novel C&C channel was established. When different proportions of C&C-Servers were invalid, DR-SNbot would send out different levels of alarms to inform attackers to construct new C&C-Servers. Then, DR-SNbot could automatically repair C&C communication to ensure its strong destroy-resistance. Under the experimental settings, DR-SNbot could resume the C&C communication in a short period of time to keep 100% of the control rate even if all the current C&C-Servers were invalid. Finally, a botnet nickname detecting method was proposed based on the difference of lexical features of legal nicknames and pseudo-random nicknames. Experimental results show that the proposed method can effectively (precision: 96.88%, recall: 93%) detect pseudo-random nicknames generated by social network-based botnets with customized algorithms.

Key words: network security, social networks, botnet, command and control channel, countermeasure

---

收稿日期：2016-05-09；**修回日期**：2016-10-25
通信作者：庹宇鹏，tuoyupeng@iie.ac.cn
基金项目：国家自然科学基金资助项目（No.61572496）；国家高技术研究发展计划（“863”计划）基金资助项目（No.2013AA014703，No.2012AA012801）
Foundation Items: The National Natural Science Foundation of China (No.61572496), The National High Technology Research and Development Program of China (863 Program) (No.2013AA014703, No.2012AA012801)

1 引言

僵尸网络已经成为互联网面临的最大安全威胁之一，它常常被用来发起各种网络攻击，如DDoS攻击、信息窃取、发送垃圾邮件以及其他恶意攻击等[1]。近年来，诸如Facebook、Twitter这类的社交网站吸引了全世界数亿用户，社交网络服务的快速发展[2]以及社会工程学攻击的多样化[3]催生了社交僵尸网络的出现，并逐渐成为危及互联网安全的一种新威胁[4]，其特点如下[5]。

1) 社交网络拥有非常庞大并且高度分散的用户规模，方便将僵尸网络产生的流量隐藏在海量用户产生的合法流量中。

2) 在社交网络中，用户按照兴趣、习惯与关系进行分组，这使社交僵尸网络传播与窃取用户私密数据变得更加方便。

3) 社交网络平台具有开放性，攻击者可利用开放平台部署欺骗性的资源与应用以诱惑用户安装。此外，由于社交网络平台不会被轻易关闭，因此，社交僵尸网络可以长期生存。

当前，社交僵尸网络已成为僵尸网络研究领域的一个重要方向。在僵尸网络威胁与日俱增的同时，其结构也越来越复杂。早期的僵尸网络普遍基于控制简单、高效的中心结构，但其单点失效的问题使其面临命令与控制（C&C, command and control）服务器被斩首的威胁[6]。随后，出现了基于P2P协议的多种新型僵尸网络结构(Equity Botnet[7]、LC Botnet[8]、Hybrid Botnet[9]、Super Botnet[10])，它们避免了单点失效问题，具备较高的顽健性[11]。然而，P2P僵尸网络命令控制的复杂性导致了巨大的计算开销以及较低的控制效率[12]。

为打击僵尸网络，保障网络空间安全，本文围绕中心结构社交僵尸网络的构建与防御展开讨论。本文的主要贡献如下。

1) 提出了一种基于社交网络的C&C信道，C&C消息经过Botmaster的加密及签名处理后，被隐藏在社交网络日志中发布，具备较高的安全性及隐蔽性。

2) 提出了一种分治自动重构机制，能够预警DR-SNbot面临的不同等级的安全威胁，并自动恢复已瘫痪的C&C通信，从而将控制率维持在100%。

3) 提出了一种僵尸昵称检测方法，能够有效检测社交僵尸网络利用自定义算法批量生成的伪随机僵尸昵称，召回率达到93%，准确率达到96.88%。

2 相关工作

从公开发表的文献看，社交僵尸网络研究领域有如下具有代表性的工作。

在C&C信道研究方面，文献[13]设计了一种基于Twitter的移动僵尸网络，但命令是以明文形式发布在Twitter上。文献[14]分析了基于Twitter的僵尸网络Naz Bots，它使用Base64对C&C信道流量进行编码，但这种编码方式本身就是可逆的。文献[15]研究了Koobface的传播及命令控制机制，但它仅对C&C信道进行了弱加密。文献[16]设计了一种隐蔽的社交僵尸网络(Stegobot)，它将控制命令隐藏在图片中进行传播，但是图片占用的空间较大，会明显增加C&C通信流量，容易被检测出来。

在重构机制研究方面，文献[17]基于Conficker的Domain-Flux技术，提出了Url-Flux的概念，进而设计了一种中心结构的移动社交僵尸网络——AndBot。然而，文献[17]仅实现了一种静态的用户名生成算法，一旦生成的用户名列表被防御者抢注或者封锁，AndBot将失去控制。文献[18]在AndBot基础上，提出了一种可重构的混合C&C结构的僵尸网络——CoolBot，设计了一种基于时间同步的动态用户名生成算法，有效弥补了AndBot的局限性。然而，CoolBot依赖于2套互补的C&C机制，其中任何一套机制自身都不具备可重构性，一旦2套机制同时失效，则CoolBot也会失去控制。本文提出的重构机制，基于类似的动态昵称生成算法，但具备自主恢复的功能和极强的抗毁性。

综上所述，由于现有工作对C&C信道的安全性考虑不够充分且缺乏行之有效的重构机制，导致僵尸网络的抗毁性不强。为此，本文提出并实现了一种强抗毁性的社交僵尸网络。

3 DR-SNbot体系结构

3.1 基本框架

DR-SNbot包括Botmaster、C&C-Server与Bot 3部分。Botmaster为僵尸网络的控制端，用于发布攻击命令。C&C-Server为命令与控制服务器，本文为某注册昵称对应的社交网络虚拟主机，即每个C&C-Server对应一个不同的注册昵称，C&C-Server用于存储并转发命令。Bot是僵尸程序，通常运行

在诸如 PC 机、笔记本、智能手机等终端上，用于从 C&C-Server 下载命令并解析执行。DR-SNbot 的基本框架如图 1 所示。

为方便讨论，在此介绍几个概念。

1) 有效 C&C-Server。有效 C&C-Server 为能提供稳定的存储转发服务的 C&C-Server，它们能被正常访问，如图 1 中不带阴影的 C&C-Server。

2) 失效 C&C-Server。失效 C&C-Server 为因为注册昵称被封锁或网络中断时间超时而导致服务中断的 C&C-Server，它们不能被访问，如图 1 中带阴影的 C&C-Server。

3) 可控 Bot。可控 Bot 为能够探测到有效 C&C-Server 的 Bot，它们能接受 Botmaster 的控制命令，处于可控状态。

4) 失控 Bot。失控 Bot 为不能探测到有效 C&C-Server 的 Bot，它们不能接受 Botmaster 的命令控制，处于失控状态。

5) C&C-Server 负载。C&C-Server 负载为一个 C&C-Server 的可控 Bot 数量。显然，失效 C&C-Server 的负载为 0。

6) 控制率。控制率是可控 Bot 数量与 Bot 总数的比值。

3.2　命令与控制信道

C&C 信道负责传输僵尸网络的内部消息，为防止第三方冒充 Botmaster 发布命令或窃听 C&C 通信内容，攻击者通常会引入数字签名及加密技术保障通信的安全性。然而，由于 DR-SNbot 的命令是发布在社交网络上的，对所有用户公开，因此，其 C&C 信道还必须具备较高的隐蔽性，以防止 C&C-Server 因发布恶意消息而被用户举报，进而被封锁。

社交网络为用户提供了诸多方便交流的平台，网络日志即是其中一种。用户发布的日志最终会嵌入到网页中，网页属于有格式的文本文档，不仅包括基本的数据元素，还包括控制数据显示格式与效果的网页标签。由于标签及其属性值通常不会显示在浏览器中，因此，可将命令隐藏于网页标签中。

图 2 给出了本文 C&C 信道的设计方案，命令的发布过程包括如下 6 个步骤。

1) 预处理：Botmaster 加密命令，并附上其数字签名，构成密文。

2) 信息隐藏：Botmaster 将密文隐藏到一个特殊网页标签的属性域中（如< a href = 密文 >< /a >），然后将该标签插入到一个正常的日志中。

3) HTTP POST：Botmaster 通过 HTTP POST 方法将日志上传到 C&C-Server 中。

4) HTTP GET：Bot 通过 HTTP GET 方法从 C&C-Server 中下载日志。

5) 信息提取：Bot 定位日志中的特殊网页标签（如<a>），并从其属性域中将密文提取出来。

6) 后处理：Bot 首先验证 Botmaster 的签名，若验证通过，则解密消息得到命令，否则丢弃消息。

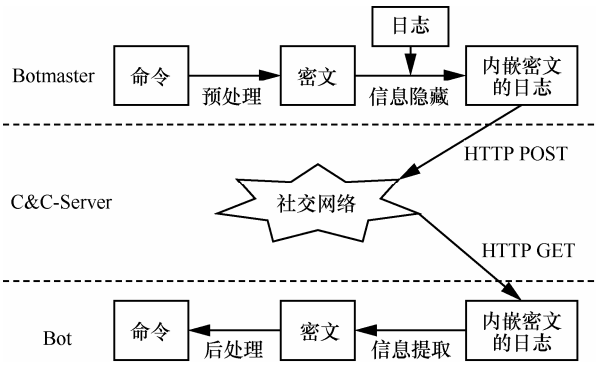


图 2　命令与控制信道设计方案

在上述方案中，所有密钥事先硬编码在 Botmaster 和 Bot 程序中，因此，无需考虑密钥分配的问题。该方案不仅能够防止第三方冒充 Botmaster、窃听 C&C 通信内容，还巧妙地将 C&C 消息隐藏在社交网络日志中，从而具备较高的安全性。

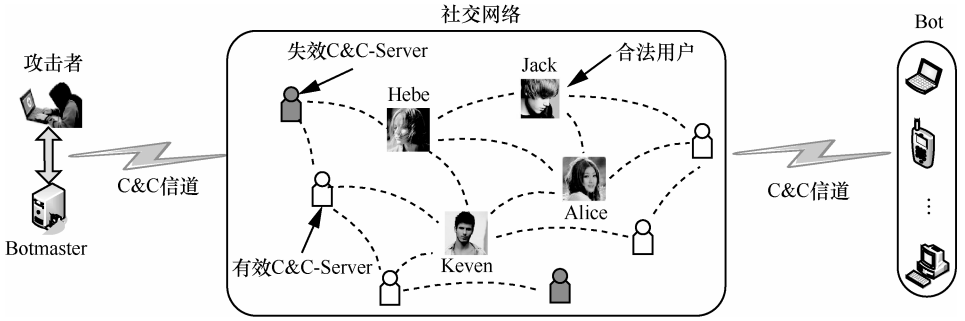


图 1　DR-SNbot 的基本框架

4 分治自动重构机制

考虑到一个 C&C-Server 占用的网络资源（空间、带宽等）是有限的，其支持的并行请求数量会受限，导致负载超过阈值的 C&C-Server 会失效。更进一步，若当前所有 C&C-Server 全部失效，则 C&C 通信将瘫痪，所有 Bot 将失去控制。针对上述问题，本文从以下 2 方面进行研究。

1) Bot 分治：将 Bot 均匀独立地分配给各 C&C-Server，并尽量保持各 C&C-Server 的负载不超过给定阈值。

2) 自动重构：在 C&C 通信瘫痪后，DR-SNbot 能够不依赖第三方自动完成整个网络的重构。

由此，本文提出了一种分治自动重构机制，该机制包括昵称生成算法、Botmaster 与 Bot 3 部分。

4.1 昵称生成算法

昵称生成算法（NGA, nickname generation algorithm）分别硬编码在 Botmaster 与 Bot 中。算法有 2 个输入参数，第 1 个参数 *seed* 为一个数字字符串，用于同步 Botmaster 与 Bot 生成的 *NicknameList* 与 *UrlList* 的时间，搜索引擎对某关键词给出返回的记录、社交网络热门主题排名等均可用于生成 *seed*。第 2 个参数 *length* 为整型，用于确定算法输出的 *NicknameList* 与 *UrlList* 的长度。*nickname* 是经过加密编码后的伪随机字符串，可有效降低昵称之间的相关性，增加防御者对昵称进行语义分析的难度。*nickname* 用于在社交网络上注册 C&C-Server，*url* 用于访问 C&C-Server。以新浪博客为例，给出了 NGA 算法的伪代码。

昵称生成算法伪代码

PROGRAM NGA(string *seed*, int *length*)

1) for *index* ← 1 to *length* do

2) *number* ← long(*seed* + string(*index*)) //构造大整数

3) *bin_string* ← E_{AES}(*number*) //用 AES 算法加密大整数

4) *nickname* ← Encode(*bin_string*) //进行伪随机编码

5) *NicknameList*.append(*nickname*) //将昵称添加到列表中

6) *url* ← 'http://blog.sina.com.cn/' + *nickname* //构造 *url*

7) *UrlList*.append(*url*) //将 *url* 添加到列表中

8) end for

9) output(*NicknameList*, *UrlList*)

4.2 Botmaster 设计

为方便讨论，首先将 *UrlList* 划分为 3 个集合：s_1、s_2、s_3，其中，s_1 为处于休眠状态的 *url* 集合，对应的 *nickname* 还未注册；s_2 为处于有效状态的 *url* 集合，对应的 *nickname* 已注册；s_3 为处于失效状态的 *url* 集合，对应的 *nickname* 已注册但失效了。

假设 C&C-Server 负载的阈值为 P，即一旦某个 C&C-Server 的负载超过 P，则该 C&C-Server 失效。记 *UrlList* 中第 i 个元素对应的 C&C-Server 的负载为 c_i。为了评估 DR-SNbot 的状态以增强其抗毁性，引入以下指标。

1) α：装载率，$\alpha=\frac{c_i}{P}$，阈值为 $\overline{\alpha}$。

2) β：报警率，$\beta=\frac{\#\left(\alpha\geqslant\overline{\alpha}|s_2\right)}{|s_2|}$，阈值为 $\overline{\beta}$，其中，$\#(\alpha\geqslant\overline{\alpha}|s_2)$ 表示 s_2 中满足条件 $\alpha\geqslant\overline{\alpha}$ 的元素个数。

3) γ：失效率，$\gamma=\frac{|s_3|}{length}$，阈值为 $\overline{\gamma}$，其中，*length* 为 NGA 输出的 *UrlList* 长度。

Botmaster 首先调用一次 NAG，获取当前的 *NicknameList* 和 *UrlList*，接着随机选取部分 *nickname* 注册，然后开始周期性地监控上述指标。当监控到 $\beta\geqslant\overline{\beta}$ 时，则 C&C-Server 负载已趋近饱和，Botmaster 发出 β 警报通知攻击者注册新的 *nickname*，并发送 balance 命令平衡各 C&C-Server 负载。当监控到 $\gamma\geqslant\overline{\gamma}$ 时，则 *UrlList* 中失效的 *url* 过多，已不具备控制大规模 Bot 的能力，Botmaster 发出 γ 警报通知攻击者更新 *seed*，预先注册新一批的 *nickname*。下面给出了以时间作为 *seed* 的 Botmaster 伪代码，其中第 1)行的 GetCurrentMonth() 函数用以返回由当前年月构成的数字字符串（如“201606”），与之对应的第 9)行的 GetNextMonth(*seed*) 函数用以返回由 *seed* 的下一月份构成的数字字符串（如“201607”）。

Botmaster 伪代码

PROGRAM BOTMASTER(float $\overline{\beta}$, float $\overline{\gamma}$, int *length*)

1) [*NicknameList*, *UrlList*] ← NGA(GetCurrentMonth(), *length*)

2) 攻击者随机选取若干 *nickname* 注册

3) while monitoring *UrlList* do
4)　if $\beta \geqslant \overline{\beta}$ then　// β 警报
5)　　攻击者另取若干 *nickname* 注册
6)　　发送一条 balance 命令
7)　end if
8)　if $\gamma \geqslant \overline{\gamma}$ then　// γ 警报
9)　　*seed* ← GetNextMonth(*seed*)　//更新 *seed*
10)　　[*NicknameList*, *UrlList*] ← NGA(*seed* , *length*)
11)　　攻击者选取若干 *nickname* 注册
12)　end if
13) end while

4.3　Bot 设计

Bot 首先调用一次 NGA，获取当前的 *NicknameList* 和 *UrlList*。若 *UrlList* 失效率达到阈值 $\overline{\gamma}$，则更新 *seed*，再次调用 NGA 获取新一批 *NicknameList* 和 *UrlList*。Bot 随机遍历 *UrlList* 访问 C&C-Server。Bot 连上 C&C-Server 后，便开始一个会话过程，周期性地爬取该 C&C-Server 上的最新日志解析命令。会话期间，如果 Bot 收到 Botmaster 者的 balance 命令，则终止会话。Bot 伪代码如下所示，由于在算法迭代过程中失效的 *url* 会从初始 *UrlList* 中自动删除（第 14)~16)行），因此，s_3 中元素个数可表示为 $|s_3| = length - |UrlList|$，进而第 3 行中失效率 γ 的表达式等价于：$\gamma = \dfrac{length - |UrlList|}{length} = 1 - \dfrac{|UrlList|}{length}$。

Bot 伪代码

PROGRAM BOT(float $\overline{\gamma}$, int *length*)
1)　[*NicknameList, UrlList*] ← NGA(GetCurrentMonth(), *length*)
2) while TRUE do
3)　if $1 - \dfrac{|UrlList|}{length} \geqslant \overline{\gamma}$ then　// γ 警报
4)　　*seed* ← GetNextMonth(*seed*)　//更新 *seed*
5)　　[*NicknameList*, *UrlList*] ← NGA(*seed* , *length*)
6)　end if
7)　从 *UrlList* 中有放回地随机抽取一个 *url*
8)　while *url* 有效 do
9)　　从该 *url* 中下载最新的命令
10)　　if 是 balance 命令 then
11)　　　break
12)　　end if
13)　end while
14)　if *url* 失效 then
15)　　*UrlList*.delete(*url*)
16)　end if
17) end while

5　实验与评估

5.1　基于新浪博客构建 DR-SNbot

新浪博客是国内主流的博客之一，本文基于新浪博客搭建 C&C-Server，并借助 Windows Live Writer 软件，将命令隐藏在博文中群发至多个博客。此外，通过设置博客权限为只对自己开放，Bot 将视为相应 C&C-Server 已失效。

由于实际环境的限制，本文共征集到 20 台 PC 机。同时种植上 Bot 程序，并设定 Bot 轮询周期为 5 min，昵称生成算法的序列号为 10。此外，为所有 Bot 申请一个公共邮箱，并将用户名及密码硬编码到 Bot 程序中。Bot 每隔 5 min 爬取一次博文，若成功解析到命令，会登录公共邮箱，并向命令中指定的目的邮箱发送反馈信息“*nickname*+*Bot_MAC*”。实验步骤如下。

1) 14:00：以月份“201603”为参数调用昵称生成算法，得到 10 个昵称。从中随机选取 5 个昵称注册 C&C-Server，依次编号为 CS_1、CS_2、CS_3、CS_4、CS_5，并向它们群发命令，同时启动所有 Bot 程序。

2) 14:20：统计并清空目的邮箱，将 CS_3 与 CS_5 对应的博客权限设置为只对自己开放。

3) 14:40：统计并清空目的邮箱，将 CS_1、CS_2 和 CS_4 对应的新浪博客权限设置为只对自己开放，用剩余的 5 个昵称注册新的 C&C-Server，依次编号为 CS_6、CS_7、CS_8、CS_9、CS_{10}，然后向它们群发命令。

4) 15:00：统计目的邮箱，终止所有 Bot 程序。

表 1 给出了对应时刻统计到的各 C&C-Server 的负载情况。其中，符号“/”表示对应的 C&C-Server 尚未注册或已“失效”；符号“→”表示在采取相应动作后，负载情况发生了变化。

表 1 C&C-Server 负载变化时刻

编号	14:00	14:20	14:40	15:00
CS_1	0	4→4	7→/	/
CS_2	0	2→2	8→/	/
CS_3	0	5→/	/→/	/
CS_4	0	4→4	5→/	/
CS_5	0	5→/	/→/	/
CS_6	/	/→/	/→0	3
CS_7	/	/→/	/→0	5
CS_8	/	/→/	/→0	4
CS_9	/	/→/	/→0	5
CS_{10}	/	/→/	/→0	3
总计	0	20→10	20→0	20

从表 1 可以看出，反馈信息总数经历了“0→20→10→20→0→20”的变化过程，说明在 C&C-Server 经历了 2 次不同程度(40%, 100%)失效的情况下，控制率仍能恢复 100%，进而验证了 DR-SNbot 在真实环境中的强抗毁性。

5.2 模拟实验

考虑到真实环境中实验规模受限，本节设计了模拟实验对 DR-SNbot 的抗毁性进行更为深入的评估。

5.2.1 实验设置

DR-SNbot 的抗毁性用于评估其在当前 C&C-Server 部分或全部失效后，能够恢复控制的 Bot 数量占 Bot 总量的百分比。模拟实验设置 DR-SNbot 的规模为 1 000，Botmaster 和 Bot 程序中的 *length* 参数设置为 30，即每次调用 NGA 获取到的 *NicknameList* 和 *UrlList* 的长度均为 30。设置装载率(α)、报警率(β)、失效率(γ)的阈值分别为：$\overline{\alpha}=0.8,\ \overline{\beta}=0.7,\ \overline{\gamma}=0.6$。Bot 的轮询周期设置为 1 s，C&C-Server 负载的阈值 P 设置为 100，Botmaster 周期性地监控每个 C&C-Server 的负载，一旦某个 C&C-Server 被监控到同时与超过 100 个 Bot 通信时，则该 C&C-Server 自动失效，对应的 Bot 转为失控状态。同时，Botmaster 周期性地监控是否有 β 警报和 γ 警报产生，监控周期分别设置为 20 s 和 60 s。初始时刻，设置 $|s_1|=25,\ |s_2|=5,\ |s_3|=0$，当需要创建新的 C&C-Server 时，Botmaster 每次新建 3 个。

5.2.2 实验结果

为实时观测 DR-SNbot 的状态，实验过程中周期性地采样各时刻的 Bot 总数与可控数量，采样周期设置为 1 s，图 3 给出了各采样时刻的 Bot 总数与可控数量。

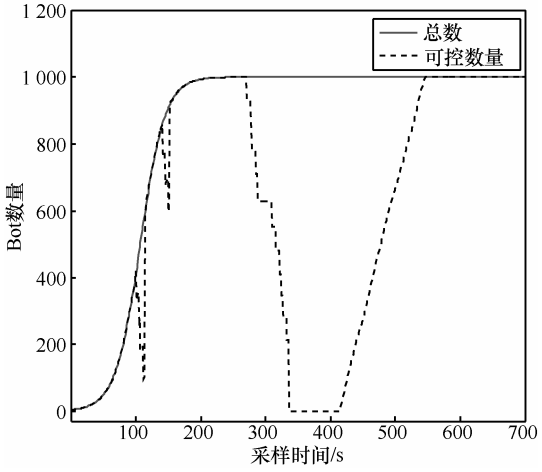


图 3 Bot 总数—可控数量变化

从图 3 可以看出，实验过程大致可分为以下几个阶段。

1) 感染阶段（0~200 s）

此阶段为 DR-SNbot 规模的增长期，整个增长过程类似人类传染病的传播过程，呈现出 S 型增长。感染初期(0~100 s)，Bot 数量较少，2 条曲线基本重合，控制率保持在 100%；感染高峰期(100~150 s)到来后，Bot 大量爆发导致部分 C&C-Server 因负载过重而失效，可控 Bot 数量出现不同程度的跌落，当创建新的 C&C-Server 后，又恢复对失控 Bot 的控制，因此，这阶段控制率出现起伏。感染末期(150~250 s)新感染 Bot 数量逐渐减少，曲线再次重合，控制率基本恢复至 100%

2) 稳定阶段（250~700 s）

此阶段无新增 Bot，Bot 总数维持 1 000 不变。为更进一步评估 DR-SNbot 的抗毁性，在到达稳定期后，在短时间内让所有 C&C-Server 失效，可控数量随即骤少。当可控 Bot 数量降至 0 后，经历了一个缓冲期(330~420 s)，由于缓冲期内 C&C-Server 全部失效，因此可控 Bot 数量一直为 0。当 BotMaster 监控到当前 C&C-Server 全部失效后，会立即创建新的 C&C-Server。缓冲期过后，DR-SNbot 进入恢复期(410~550 s)，不同于感染期(0~250 s)的 S 型增长，此阶段所有 Bot 都已经活跃，它们只是随机寻找有效的 C&C-Server 连接，因此呈现线性增长的规律。最后 DR-SNbot 再次进入稳定期(550~700 s)，其控制率恢复为 100%，恢复期的长短与 Botmaster

的监控周期及 Bot 的轮询周期有关。模拟实验结果表明，DR-SNbot 具备极强的抗毁性。

6　僵尸昵称检测方法

诸如 DR-SNbot 和 CoolBot[18]这类社交僵尸网络，基于自定义算法动态生成伪随机昵称，用以注册 C&C-Server，具备较强的抗毁性。然而，同一算法批量生成的伪随机僵尸昵称与绝大多数用户注册的合法昵称在词法特征上存在较大差异性[19]。这种差异性往往可体现为 2 类昵称中频繁子串的分布规律不同，为此，本文有针对性地提出了一种僵尸昵称检测方法，该方法基本原理如图 4 所示。

6.1　基本原理

僵尸昵称检测方法主要分为 4 个模块。采集模块一方面从社交网络中采集用户昵称，用作合法样本，另一方面通过逆向僵尸网络的昵称生成算法，批量生成相应的僵尸昵称，用作僵尸样本。所有样本均被分为训练样本和测试样本，前者用于阈值学习，后者用于方法测试。挖掘模块利用频繁子串挖掘算法从合法训练样本中挖掘出频繁表，表中包含样本中出现最频繁的各阶子串（阶代表子串长度）以及对应的频数（出现次数），并将频繁表发送给学习模块。学习模块基于阈值学习算法从训练样本中学习得到阈值。检测模块中，若测试样本中某个昵称的可信度低于该阈值，则判定为僵尸昵称；否则判定为合法昵称。

6.2　频繁子串挖掘算法

假设合法训练样本规模为 *SIZE*，用 *MIN_SUP* 代表最小相对支持度(根据专家经验设定)，该算法基于 Apriori[20]的思想，从样本中挖掘出所有出现频数不低于 *SIZE·MIN_SUP* 的各阶子串，步骤如下。

1) 构造字符表：根据社交网络的昵称命名规则，字符表通常由字母(不区分大小写)、数字、下划线以及连接符构成。

2) 生成 1 阶频繁表 L_1：遍历样本，统计字符表中各字符出现的频数，从中挑出频数不低于 *SIZE·MIN_SUP* 的字符，将该字符及其对应的频数添加到 L_1 中。

3) 置 $k=2$。

4) 若 L_{k-1} 不为空，则转步骤 5)，否则转步骤 8)。

5) 生成 k 阶候选表 C_k：任取 L_{k-1} 中的 2 个子串，若满足剪枝规则，其中一个子串的后缀与另一个子串的前缀相同，则按图 5 所示拼接成新的 k 阶字符串，并添加到候选表 C_k 中。

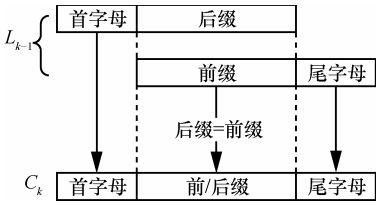


图 5　候选表 C_k 生成示意

剪枝规则有效避免了生成的候选表 C_k 过大，降低了算法的复杂度，该规则基于如下 2 个观测结论。

① 一个 k 阶字符串仅能被拆分为 2 个 k–1 阶

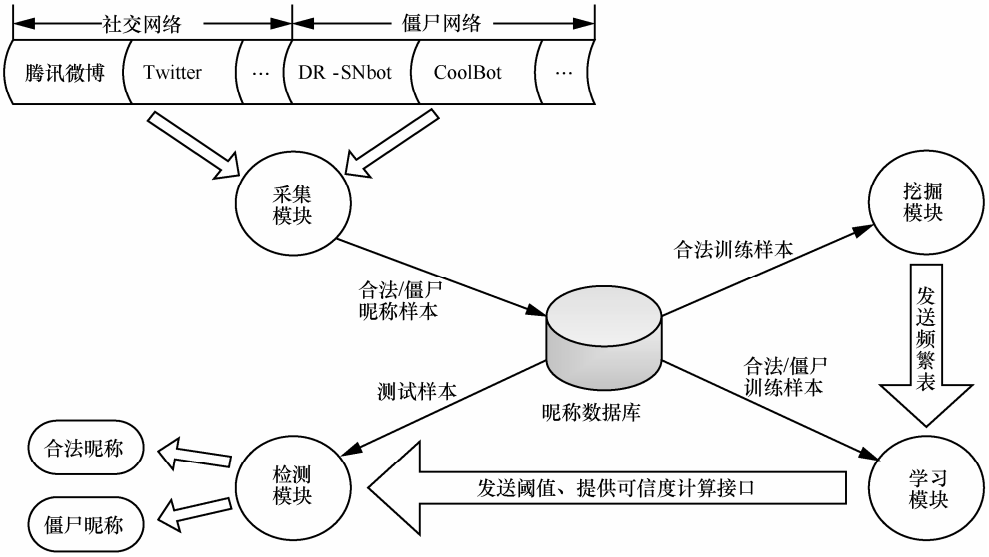


图 4　僵尸昵称检测方法原理

子串，且其中一个子串的后缀必然与另一子串的前缀相同。

② 若一个字符串为 k 阶频繁字符串，则其任意 k–1 阶子串均为频繁字符串。

6）生成 k 阶频繁表 L_k：遍历样本，统计 C_k 中各字符串出现的频数，从中挑选频数不低于 $SIZE{\cdot}MIN\_SUP$ 的字符串，将该字符串及其频数添加到 L_k 中。

7) 置 $k=k+1$，并转步骤 4）。

8) 将生成的所有频繁表发送到学习模块，算法结束。

6.3 阈值学习算法

阈值学习的关键在于量化昵称子串在频繁表中的分布，以得到区分合法训练样本和僵尸训练样本的阈值，学习步骤如下。

1) 构造子序列：假设频繁表中字符串的最大阶数为 $k(k\geqslant 1)$，将训练样本中的字符串依次分割为 1~k 阶子序列。表 2 给出了昵称“abc123”的 1 阶和 2 阶子序列划分结果。

表 2　　子序列划分示例

昵称	1 阶子序列	2 阶子序列
abc123	a, b, c, 1, 2, 3	ab, bc, c1, 12, 23

2) 计算可信度：通过查找相应的频繁表，可计算出训练样本中任意字符串的 1~k 阶频数均值，然后累加各阶均值并进行归一化处理，得到相应字符串的可信度。

3) 获取阈值：设定合适的可信度区间大小 $ZoneSize$，统计各区间内的昵称数量，绘出合法训练样本及僵尸训练样本的（可信度—字符串数量）曲线，取曲线交叉点的横坐标作为阈值。

6.4 实验结果

本文利用腾讯微博开放平台提供的 API 接口，采集到 92 659 个用户昵称，用作合法样本。此外，实现了 DR-SNbot 的昵称生产算法（NGA），批量生成 600 000 个僵尸昵称，用作僵尸样本。首先，各取 50 000 个昵称用作训练样本，用于学习阈值。设定最小相对支持度 MIN_SUP=0.03，分区大小 $ZoneSize$=0.005，训练结果如图 6 所示。

从图 6 中可以看出，曲线交点的横坐标大概在 0.05 处，因此设定阈值为 0.05。接着，从剩余样本中各取 100 个昵称构成测试样本，测试结果如图 7 所示。

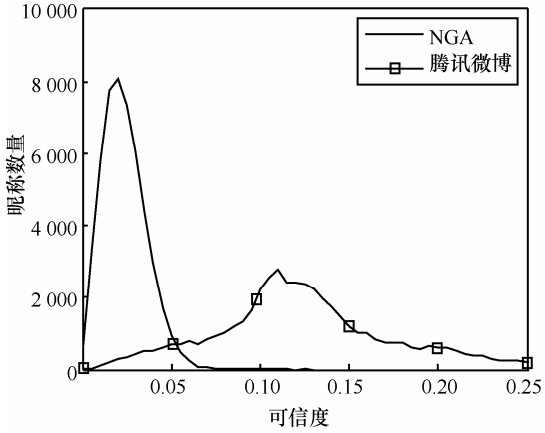


图 6　训练结果

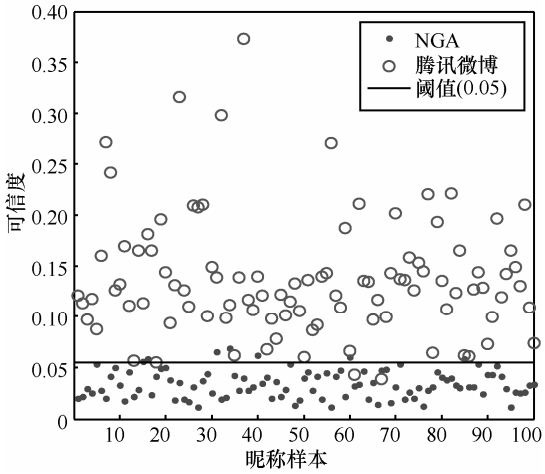


图 7　测试结果

从图 7 可以看出，该方法共检测出 96 个僵尸昵称，其中包括 93 个 NGA 生成的昵称和 3 个腾讯微博昵称，召回率为 93%，准确率为 96.88%。实验结果表明，该方法可有效检测出利用伪随机算法生成的僵尸昵称。

7　结束语

本文提出了一种强抗毁性的社交僵尸网络（DR-SNbot），利用网页信息隐藏、加密技术及数字签名技术构建了安全的 C&C 信道，并提出了分治自动重构机制增强了 DR-SNbot 的抗毁性。本文分别在新浪博客和模拟环境中进行了实验，实验结果表明，DR-SNbot 在 C&C-Server 部分甚至全部失效后，控制率仍能恢复 100%。最后，本文提出了僵尸昵称检测方法并利用真实数据进行了测试，实验结果表明，该方法召回率达到 93%，准确率达到 96.88%。本文工作旨在进一步推动社交僵尸网络积极防御技术的发展。

参考文献：

[1] FABIAN M, TERZIM s. A multifaceted approach to understanding the botnet phenomenon[C]//2006 ACM SIGCOMM Internet Measurement Conference (IMC). 2006.

[2] KWAK H, LEE C, PARK H, et al. What is twitter, a social network or a news media[C]//19th International Conference on World Wide Web, ACM. 2010: 591-600.

[3] ABRAHAM S, CHENGALUR-SMITH I. An overview of social engineering malware: Trends, tactics and implications[J]. Technology in Society, 2010, 32(3): 18196.

[4] LI S, YUN X, HAO Z, et al. Modeling social engineering botnet dynamics across multiple social networks[J]. Information Security and Privacy Research, 2012: 261-272.

[5] ATHANASOPOULOS E, MAKRIDAKIS A, ANTONATOS S, et al. Antisocial networks: turning a social network into a botnet[C]//Information Security, ICS. 2008: 146-160.

[6] MA X, GUAN X, TAO J, et al. A novel IRC botnet detection method based on packet size sequence[C]//2010 IEEE International Conference on Communications (ICC). 2010: 1-5.

[7] ZOU C, CUNNINGHAM R. Honeypot-aware advanced botnet construction and maintenance[C]//Dependable Systems and Networks, 2006. DSN 2006. International Conference. 2006: 199-208.

[8] SU C, ZHANG L F, GUAN Y, et al. A framework for P2P botnets[C]//International Conference on Communications and Mobile Computing. 2009.

[9] WANG P, SPARKS S, ZOU C. An advanced hybrid peer-to-peer botnet[J]. Dependable and Secure Computing, IEEE Transactions, 2010, 7(2): 11127.

[10] VOGT R, AYCOCK J. Attack of the 50 foot Botnet[EB/OL]. http://pages.cpsc.ucalgary.ca/~aycock/papers/50foot.pdf.

[11] 李书豪, 云晓春, 郝志宇, 等. MRRbot: 基于冗余机制的多角色 P2P 僵尸网络模型[J]. 计算机研究与发展, 2011, 48(8): 1488-1496.
LI S H, YUN X C, HAO Z Y, et al. MRRbot: a multi-role and redundancy-based P2P botnet model[J]. Journal of Computer Research and Development, 2011, 48(8): 1488-1496.

[12] HA D, YAN G, EIDENBENZ S, et al. On the effectiveness of structural detection and defense against P2P-based botnets[C]//IEEE/IFIP International Conference on Dependable Systems & Networks, 2009. DSN09. 2009: 297-306.

[13] 李跃, 翟立东, 王宏霞, 等. 一种基于社交网络的移动僵尸网络研究[J]. 计算机研究与发展, 2012, 49(Suppl.): 1-8.
LI Y, ZHAI L D, WANG H X, et al. Mobile botnet based on SNS[J]. Journal of Computer Research and Development, 2012, 49(Suppl.): 1-8.

[14] KARTALTEPE E, MORALES J, XU S, et al. Social network-based botnet command-and-control: emerging threats and countermeasures[C]//Applied Cryptography and Network Security. 2010: 511-528.

[15] THOMAS K, NICOL D. The koobface botnet and the rise of social malware[C]//Malicious and Unwanted Software (MALWARE), 2010 5th International Conference. 2010: 63-70.

[16] NAGARAJA S, HOUMANSADR A, PIYAWONGWISAL P, et al. Stegobot: a covert social network botnet[C]//Information Hiding, Springer. 2011: 299-313.

[17] CUI X, FANG B, YIN L, et al. AndBot: towards advanced mobile botnets[C]//4th USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET' 11). 2011.

[18] LIU C, LU W, ZHANG Z, et al. A recoverable hybrid C&C botnet[C]//2011 6th International Conference on Malicious and Unwanted Software (MALWARE). 2011: 110-118.

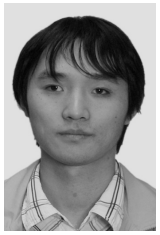
[19] YADAV S, REDDY A K K, REDDY A L, et al. Detecting algorithmically generated malicious domain names[C]//10th ACM SIGCOMM Conference on Internet Measurement. 2010: 48-61.

[20] BORGELT C, KRUSE R. Induction of association rules: apriori implementation[C]//Compstat. Physica-Verlag HD. 2002: 395-400.

作者简介：

尹涛（1989-），男，重庆人，中国科学院信息工程研究所博士生，主要研究方向为网络与信息安全。

李世淙（1981-），男，山东临沂人，国家计算机应急技术处理协调中心工程师，主要研究方向为网络安全事件监测、网络行为分析。



庹宇鹏（1984-），男，河北廊坊人，中国科学院信息工程研究所助理研究员，主要研究方向为网络异常检测、移动互联网大数据挖掘。

张永铮（1978-），男，黑龙江哈尔滨人，博士，中国科学院信息工程研究所研究员、博士生导师，主要研究方向为网络安全态势感知。