高技术通讯 2013 年 第23 卷 第6 期:571 ~577

doi:10.3772/j.issn.1002-0470.2013.06.004

基于命令语法结构特征的IRC 僵尸网络控制命令识别方法①

闫健恩② 张兆心 许海燕

(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

摘 要 通过分析僵尸网络控制命令的语法结构特征,提出一种基于语法结构特征识别IRC 僵尸网络控制命令的方法。该方法首先分析命令关键字和命令参数的词法特征,对其进行归一化处理,其次从参数的类型和数量等语法结构特征出发,定义三种僵尸网络控制命令的文法形式化描述,以适合不同的命令语法结构,并基于LR 语法分析技术实现识别原型系统。最后经过实验测试,结果表明文法对僵尸网络控制命令有很好的识别能力,从而验证了方法的有效性,且性能能够满足实际的需要。

关键词 僵尸网络,控制命令,语法结构,形式化

0 引 言

僵尸网络(botnet)是指采用一种或多种传播手段使大量主机感染僵尸程序病毒,从而在控制者和被感染主机之间形成的一个可进行一对多控制的网络[1]。它可以用来进行分布式拒绝服务(distributed denial of service, DDOS)攻击、发送垃圾邮件、网络仿冒、窃取机密资料和散布蠕虫病毒等。根据国家互联网应急中心2010 上半年的抽样监测结果,我国有23 万多个IP 地址的主机感染僵尸程序[2],对互联网的安全构成严重威胁。僵尸网络已成为发起网络攻击的一个主要平台。目前流行的僵尸网络采用多种协议来实现控制与通信,主要有互联网在线聊天(Internet Relay Chat, IRC)协议、HTTP 协议和P2P 协议,这是目前互联网中的一个重要威胁[3]。IRC 僵尸网络简单、灵活、易控,因而它仍然是攻击者手中的重要手段。

目前跟踪与检测僵尸网络的手段主要分为两大类,第一类为主动探测,即主动参与僵尸网络的交互过程,从而发现其踪迹;第二大类是被动性的跟踪与检测,即通过监测网络数据流或相关终端的行为,发现僵尸网络的活动。被动性的检测又可以细分成以下三小类:(1)利用密网和蜜罐进行跟踪和检测;(2)从网络通信流量中检测僵尸网络;(3)基于行为检测僵尸网络。僵尸程序无论在客户端还是在网络通信中,其行为有一定的固定规律,可以通过分析这些规律对其进行检测。根据文献[4]僵尸网络昵称的特征提出了检测僵尸网络的方法。文献[5]实现了一个主动探测僵尸网络的系统,通过主动参与僵尸网络的交互过程,从而经过简单的几轮操作即可检测出僵尸网络活动的存在。德国的蜜网项目和北京大学的狩猎女神项目主要利用密网和蜜罐发现僵尸网络的踪迹,并对其进行跟踪和分析。在基于网络通信流量的检测方面,Livadas 和Strayr 等使用基于机器学习的方法进行僵尸网络的检测[6,7],Karasaridis 等人研究了骨干网IRC 僵尸网络的检测方法[8],另外,Binkley 和Singh 等人通过分析僵尸网络发起攻击时的异常TCP 流量进行检测僵尸网络[9]。Gu 等人完成的BotHunter[10]、BotSniffer[11]和BotMiner[12]主要基于网络流量检测僵尸网络。文献[13]提出通过分析僵尸网络通信信道中的统一资源链接(URL)数据来确定是否存在僵尸网络并确认其类型。另外一种基于主机入侵检测系统思想的僵尸网络检测和分类方法也被使用[14]。由于僵尸网络在实现控制的过程中,控制者必须在通信信道中发送控制命令,文献[15]提出了一种通过自动机识别僵尸网络控制命令的方式来进行检测。上述

① 863 计划(2007AA010503),国家自然科学基金(61100189),山东省中青年科学家奖励基金(BS2011DX001),威海市科技攻关(2010-3-96)和哈尔滨工业大学科研创新基金(HIT. HSRIF. 2011119)资助项目。

② 男,1977 年生,博士生,讲师;研究方向:网络信息安全,僵尸网络等;联系人,E-mail:yje@hitwh.edu.cn

(收稿日期:2012-06-25)

方法都有不同的角度,因而都有其局限性。密网蜜罐的跟踪,依赖于密网蜜罐的部署分布。基于主动探测的检测方法对通信过程加密的通信信道无能为力,而基于网络通信流量检测过程中,又依赖大量的网络流量数据,对数据的搜集和分析提出了挑战,并且在实时检测上不能满足要求。对于基于行为的检测方法,行为的连续性和考察的周期是对方法有效性的一个考验,而在某些 bot 中发现存在内置的昵称列表,如 H-Bot。基于僵尸网络控制命令进行检测的方法,虽然有效,但它依赖于控制命令的长度以及构成命令本身字符串的特征,对于不具备特征的字符串无法进行准确识别,同时随着命令长度的增加,自动机状态会急剧增多,影响检测效率。鉴于此,本文提出了一种基于僵尸网络控制命令语法结构特征的识别方法,该方法基于网络流检测,提取 IRC 终端的交互命令,通过命令语法结构的分析,判断其是否为僵尸网络控制命令,克服只分析字符串特征的缺陷。实验表明,该方法在识别僵尸网络控制命令的准确性和性能上都可以满足需求。

1 IRC 协议僵尸网络控制命令分析

僵尸网络一个最大的特点就是可控性。为了达到控制的目的,感染了 bot 的客户端,要主动或被动地与控制者进行通信。基于 IRC 的僵尸网络,bot 程序会主动加入到指定的 IRC 频道中。通过频道,控制者向被控制的 bot 客户端发送控制命令,从而实现恶意目的。这些控制命令是实现恶意行为的重要途径。

通过分析获取的 75 个可运行的 bot 程序样本,提取获得僵尸网络控制命令信息 4080 条,将僵尸网络的控制命令从功能上分为三大类:第一类是一般控制类命令,这类命令主要是控制者通过其控制被控僵尸主机进行一些简单的操作,如登陆服务器、更改昵称或者窃取被控僵尸主机上的一些信息,比如 .sysinfo,该命令是搜集主机的硬件和软件信息;第二类是更新下载类命令,这类命令是通知被控僵尸主机进行 bot 程序升级或下载一些程序,如. download http://www.host.net/file.exe c:\windows\devldr32.exe 1,该命令控制僵尸主机到指定的网站下载某个程序到本地的某个位置;第三类是传播攻击类命令,这类命令主要通知被控僵尸主机发起相应的攻击活动和主动传播,比如分布式拒绝服务(DDoS)攻击、发送垃圾邮件等,如. udp 127.0.0.1 1000 4096 100,该命令控制僵尸主机使用 UDP 协议,向指定的目标地址发送 1000 个数据包,数据包大小为 4096B,延迟为 100ms。

分析僵尸控制命令的结构发现,它们的结构一般为:<命令关键字 > <命令参数 >。命令组成的单词个数统计见图 1 所示。其中只有命令关键字的命令条数为 1474 条,其余都是具有命令参数。继续对构成僵尸网络控制命令的命令关键字进行分析,发现这些命令关键字中大部分包含一些特殊的字符,统计数据如表 1 所示。根据这些特殊字符的有无及所在位置,把它们分成三类:

(1)无特殊字符和固定前缀,就是由英文字符构成的单词。

(2)在命令前加特殊符号前缀,如“.”、“!”、“-”、“_”、“MYM”、“#”、“?”和“&”等。例如. download、! down 等。

(3)其他格式,有固定字符前缀,且单词中间有特殊字符如“.”,例如:irc.nick、ddos.syn 等。

由此可见僵尸网络控制命令关键字中有前缀的占了大多数。如果将第三类命令关键字(“ddos.”、“irc.”)也作为一种特殊前缀来看的话,那么具有前缀的命令关键字就占到了总数的 90% 以上。对第一类僵尸控制命令分析,发现其中有少量命令,占整个僵尸命令的很小一部分,而且这几个僵尸程序的功能也比较简单,只有少数几个功能,控制能力也不强。

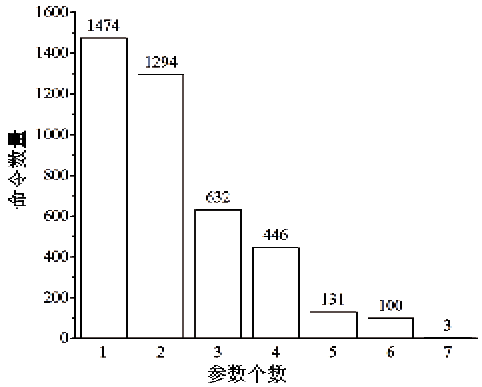

图 1 僵尸网络控制命令单词数量图

表 1 命令关键字类别百分比表

类别 / 数量	第 I 类	第 II 类	第 III 类
Bot 数量	5	59	11
百分比(%)	6.66	78.67	14.67

对命令参数的组成结构进行分析,得到:

(1) 无参数。这类命令没有参数,僵尸程序根据命令关键字进行相应的处理。

(2) 1个参数。这个参数可以是一个字符串或者数字,例如. dns www. google. com。

(3) 2个参数。参数也以数字和字符串的形式出现。

(4) 3个参数。这类命令多见于攻击类和更新下载类命令,如. download http://www. host. net/file. exe c:\windows\devldr32. exe 1。

(5) 4个或4个以上参数。此类命令大多数为攻击类和更新下载类命令,如. ping 127.0.0.1 1000 4096 100。而且超过5个参数的僵尸网络命令非常少。

通过以上的分析可以看出,绝大部分的僵尸网络控制命令的结构都是类似的,而且在命令参数部分的结构也具有相似性,比如具有3个及3个以上参数的命令参数中,第2个参数是字符串,第3、4个参数多为数字,因此对它们从结构上进行形式化,抽象出共同特征。命令参数单词的字符构成分类统计情况,如图2所示。

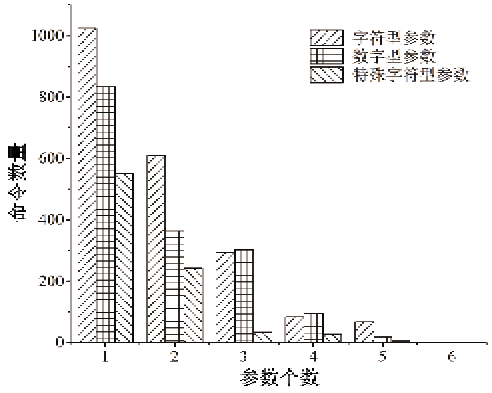

图2 僵尸网络控制命令参数构成统计图

2 IRC协议僵尸网络控制命令的识别方法

在上节分析的基础上,下面给出基于语法结构特征的命令识别方法。分两步进行,首先对控制命令进行形式化描述,然后描述方法的实现。

2.1 僵尸网络命令的形式化描述

实现僵尸网络控制命令结构的自动识别,首先对其进行形式化描述,同时,为了结构分析的需要,针对僵尸网络命令中的命令关键字及命令参数进行归一化处理。根据实际的统计,将其分为三类:命令关键字、普通字符串型参数、数字型参数和特殊字符串型参数。

因此定义僵尸网络控制命令单词的分类函数 $f(X)$ 如下:

$$f(X)=\begin{cases} k & X\in[!\ |\$|\#|.|\_|-|?]([a-Z]|[0-9]^{+} \\ n & X\in([0-9]^{+}(.[0-9]^{+}) \\ w & X\in([a-Z]|[0-9])^{+}[.|/|\backslash|:]^{+}([a-Z]|[0-9])^{+} \\ t & X\in([a-Z,0-9]^{+}) \end{cases} \tag{1}$$

即,以特殊字符“.”、“!”、“-”、“$”、“#”、“?”和“_”开头的字符与数字组成的单词聚类为“k”,这类单词常见为命令关键字,例如“! download”等;整数或者小数这样的单词聚类为“n”,一些端口号、数据包长度多见于此类单词,称为数字型参数;对IP地址、统一资源链接(URL)以及系统文件路径这样的单词,其中包含一些特殊符号,故聚类为“w”,称为特殊字符串型参数,例如 www. google. com、c:\windows\devldr32. exe 等;其他由数字或者字符及其他一些非命令关键字中的开头字符构成的单词聚类为“t”,称为普通字符串型参数。经过分类聚合后,僵尸网络的控制命令就可以使用分类后的单词表示。例如. download http://www. host. net/file. exe c:\windows\devldr32. exe 1 就可以表示成 kwwn 的形式。

2.2 基于命令语法结构的僵尸网络命令识别方法

经过以上的统计与分析,下面讨论如何将僵尸网络控制命令的语法结构形式化,给出一般的语法定义,从结构角度对僵尸网络命令进行识别。

首先定义僵尸网络控制命令语法结构的文法G。文法由4部分构成,结构定义如下:

$$G=(Vt, Vn, P, S) \tag{2}$$

其中 Vt 表示文法中终结符号集合,描述构成语言句子的单词集合;Vn 表示文法中的非终结符号集合,描述文法中的语法成分;P 是文法的产生式集合,描述文法的语法结构规则;S 是文法的开始符号,表示所有的语法分析都从它开始。有了文法的形式定义,就可以根据实际语言的情况,定义对应的文法来进行其语法结构。在本文中,使用LR(Left-Right)语法分析技术,实现文法的语法分析。在具体分析句子前,给出以下定义。

定义1 分析成功:如果给定的一个句子能够通过语法分析器分析,返回“accept”时,则称分析成功。分析成功说明该句子符合文法的语法结构。

定义 2 分析失败:如果给定的一个句子能够通过语法分析器分析,返回“fail”时,则称分析失败。分析失败说明该句子不符合文法的语法结构。

当僵尸网络控制命令符合该文法的语法结构时,语法分析器返回结果为“accept”,如果不符合文法的语法结构,则返回“fail”。

定义 3 误报:如果一个句子不是僵尸网络命令,但语法结构分析为分析成功,则称该行为为误报。

定义 4 漏报:如果一个句子是僵尸网络命令,但语法结构分析为分析失败,则称该行为为误报。

根据前面对僵尸网络控制命令分析的结果,每个都是 < 命令关键字 > < 命令参数 > 这样的结构,并且不同的命令,命令参数的个数也不一样。结合前面对构成僵尸网络控制命令单词的分类,可以这样描述僵尸网络控制命令:它是由命令关键字开头的,由普通字符串参数、数字型参数或特殊字符串参数组成的字符串。所有这些字符串的集合构成了僵尸网络的控制命令集合。由此,将所有的僵尸网络控制命令的文法分成三类进行定义:

(1) 一般结构的僵尸网络控制命令文法;

(2) 具有强特征僵尸网络控制命令文法;

(3) 多参数特征僵尸网络控制命令文法。

根据式(2)定义第一类僵尸网络控制命令的文法 G1,定义为

$$\begin{aligned} &S \to k \mid kA \mid tA \\ &A \to t \mid n \mid w \mid A \mid nA \mid wA \end{aligned} \tag{3}$$

对僵尸网络控制命令识别的准确性,取决于文法是否准确描述它的语法特征,而文法定义的完整与准确,得益于对僵尸网络控制命令的特征的分析。经过实验分析(在实验数据部分给出),发现文法 G1 对僵尸网络控制命令的识别误报率较高,因此需要文法能描述其有效关键特征,从而提高识别准确性。下面给出第二个识别僵尸网络控制命令的文法定义,观察其实际的分析效果。

根据前面对僵尸网络控制命令的命令关键字的分析,绝大部分都包含特殊的前缀,结合对僵尸网络命令中单词的分类,给出第二类僵尸网络控制命令的文法 G2,定义为

$$\begin{aligned} &S \to k \mid kA \\ &A \to k \mid w \mid n \mid t \mid AB \\ &B \to k \mid t \mid w \mid n \end{aligned} \tag{4}$$

经过实验结果分析,发现该文法对僵尸网络控制命令的识别效果很好,达到了预期的目的。但是,在已经掌握的僵尸网络控制命令中,还存在一些特殊的命令,这几十条命令无固定前缀,因此没有上面文法定义中的有效特征,虽然在整个掌握的命令中,只占有很小部分,但在实验中也发现对它们的识别全部漏报,所以对于文法 G2 来说,虽然误报率下降,但是对于这种没有关键结构特征的命令是不能准确识别的,缺乏识别命令的完备性,故需要再次分析僵尸网络控制命令的结构特征,定义新的文法,降低命令识别的漏报率。

继续分析僵尸网络控制命令的单词构成,发现已有的命令单词总数不超过 7 个,即命令参数不超过 6 个,而且具有 3 个及以上参数的命令中,在第 2、第 3 和第 4 个位置上出现数字型参数和特殊字符串参数的数量比例很高,因此通过在这几个位置的参数特征对语法结构进行再次定义。定义第三类僵尸网络控制命令的文法 G3:

$$\begin{aligned} &S \to k \mid kA \mid t \mid tA \\ &A \to k \mid w \mid n \mid tB \mid wB \mid nB \\ &B \to n \mid w \mid wC \mid nC \\ &C \to n \mid w \mid wD \mid nD \\ &D \to n \mid w \mid wE \mid nE \\ &E \to n \mid w \mid t \mid EF \\ &F \to n \mid w \mid t \end{aligned} \tag{5}$$

文法定义命令长度超过 2 个单词时在第 2、第 3 和第 4 命令参数位置上,参数为数字型参数和特殊字符串参数。经过实验检测,文法 G3 在误报率和漏报率上达到很好的均衡,可以满足识别的需要。

3 实验及结果分析

下面分别使用僵尸网络控制命令、标准英文语句和 IRC 聊天记录三组数据,检验给出的僵尸网络控制命令的文法,能否正确地识别出合法的僵尸网络命令。其中,标准英文语句来源于《疯狂英语 900 句》,都是由标准的英文单词构成,包含一些常用的标点符号,如“!”、“.”和“,”;IRC 聊天语句来源于登录公共 IRC 服务器上的聊天数据,由英文单词或者包含一些特殊符号的字符构成。测试环境为双核 CPU 频率 1.8MHZ,1G 物理内存,程序使用 VC + + 实现。

3.1 可行性测试

下面分析识别方法的可行性。实验数据中僵尸网络控制命令 196 条,标准英文语句 286 条、IRC 聊天记录 297 条,另外一组实验数据是 15 条无前缀的僵尸网络控制命令(在获取的僵尸网络控制命令

中,有几十条是无固定前缀的,它们只占整体的很小一部分,且功能相对简单)。使用LR语法分析技术实现文法的语法分析程序。

文法G1对实验数据的识别情况如表2所示。

表2 文法G1的识别率

类别/数量	僵尸控制命令	标准英文语句	IRC聊天记录	无前缀控制命令
总数	196	286	297	15
成功数量	177	278	232	15
失败数量	19	8	65	0
识别率(%)	90.31	97.2	78.11	100

通过表2中的分析结果可以看到,文法G1虽然对真正的僵尸网络命令分析成功率达到了90%以上,但对非僵尸网络控制命令语句的分析成功率也非常高,甚至对于标准的英文语句达到了97.2%的分析成功率。

使用上述实验数据组,文法G2得到的实验结果如表3所示。

表3 文法G2的识别率

类别/数量	僵尸控制命令	标准英文语句	IRC聊天记录	无前缀控制命令
语句总数	196	286	297	15
成功数量	196	0	6	0
失败数量	0	286	291	15
识别率(%)	100	0	2.02	0

表3中的数据表明,对于非僵尸网络控制命令的识别率急剧降低,且对于标准英文语句和IRC聊天记录的过滤达到了很高的水平,达到这个目标的重要原因在于文法中的产生式S→k和S→k A,这样的语法结构定义,对僵尸网络控制命令的命令关键字进行了严格的定义,即这些关键字都有固定的前缀。

文法G3对4组实验数据的识别情况如表4所示。

表4 文法G3的识别率

类别/数量	僵尸控制命令	标准英文语句	IRC聊天记录	无前缀控制命令
总数	196	286	297	15
成功数量	141	59	56	14
失败数量	55	227	241	1
识别率(%)	71.94	20.63	18.85	93.33

由表4中数据可以发现,文法G3识别僵尸网络命令语句的成功率能达到70%以上,并且对无前缀的僵尸网络控制命令也能进行准确识别。

三种文法对实验数据的误报率和漏报率,分别如表5和表6所示。

表5 三种文法的误报率

	标准英文语句	IRC聊天记录
G1	97.2%	78.11%
G2	0	2.02%
G3	20.63%	18.85%

表6 三种文法的漏报率

	僵尸控制命令	无前缀控制命令
G1	9.69%	0
G2	0	100%
G3	29.06%	6.67%

从表5的数据分析可以看到,文法G1的误报率非常高,分析文法定义发现,出现高误报的原因是对僵尸网络控制命令关键特征描述的不充分,即定义的语法结构约束过于宽泛,从而造成高误报率。表6中的漏报数据显示,文法G2虽然对非僵尸网络控制命令的句子有很好过滤能力,不过,这样的代价是对无前缀特征的僵尸网络控制命令不能识别。因为文法对命令的强特征定义,所以对识别没有该特征的控制命令无效。对于文法G3,通过分析误报率和漏报率的数据,可以看出其识别误报率控制在20%以下,对僵尸网络控制命令的识别可以达到70%以上,故三种文法定义中,文法G3识别命令的综合能力最好,适合僵尸网络控制命令的识别与检测。

3.2 性能测试

接下来测试识别方法的处理性能。选取bot控制命令、IRC聊天记录和标准英文语句三类5组数据,每组320条语句,分别考察每种文法对不同字符长度句子的处理能力。第一组数据为15个字符一下的语句;第二组为15-20个字符的语句;第三组为20-30个字符的语句;第四组为30-40个字符的语句;第五组为40-60个字符的语句。常用的僵尸网络控制命令的长度一般都少于60个字符,个别命令长度超过60字符,因此主要测试60个字符以下长度的命令和语句的处理性能。

三种文法构建的语法分析程序对不同分组的

bot 命令识别性能如图 3 所示。可以看出，在小于 15 个字符的 bot 命令（一般为单个或 2 个单词的 bot 控制命令）识别中，文法 2 的分析器识别速度明显高于其他两个，这是由于文法 2 是强特征文法，当识别 bot 命令时，立刻就会匹配文法结构，因此速度明显加快。而随着参数数量的增加，这种速度的优势就不很明显，不同文法的结构定义，对处理性能的影响不明显。图 4 分别分析了三种文法对不同分类各组数据分析的性能。由图 4 可以看到，文法 1 构建的分析器 H1 对 bot 控制命令的分析速度，明显高于 IRC 聊天语句和标准英文语句的分析速度，这说明 bot 命令特征的存在，系统状态变化数量少，分析器很快识别出来 bot 命令。图 5 中，可以发现 IRC 聊天记录和标准英文语句识别速度几乎为 bot 命令识别速度的 2 倍，这是由于文法 2 对 bot 命令强特征的定义，非 bot 命令的语句不具备这样的特征，因此分析器 H2 可以立即识别处理，快速排除非 bot 命令语句。图 6 表明，在识别字符或参数数量增多的语句时，文法 3 定义的分析器 H3 可以快速排除非 bot 控制命令，提高识别的处理效率。

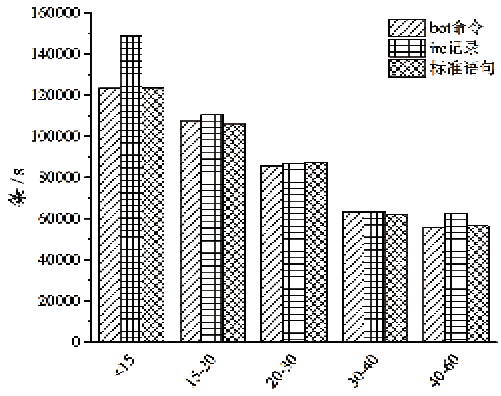

图 3 三种文法对 bot 命令识别性能分析图

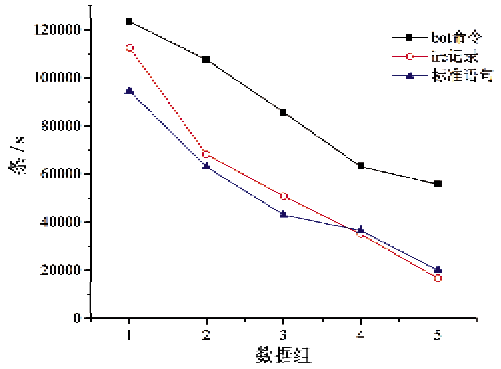

图 4 文法 G1 语法分析性能图

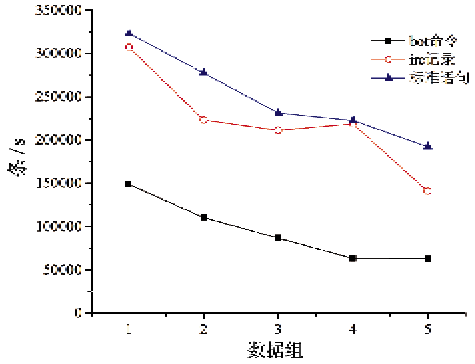

图 5 文法 G2 语法分析性能图

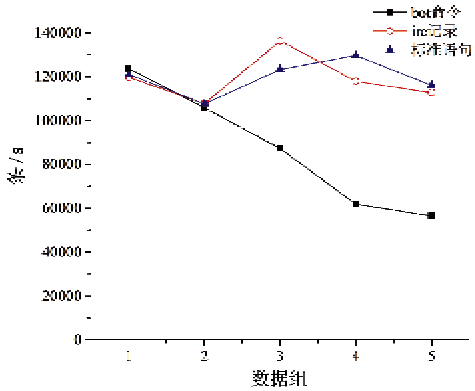

图 6 文法 G3 语法分析性能图

通过测试结果分析可以看到，使用语法结构特征对僵尸网络控制命令识别的准确较高，方法执行效率也可满足实际需要，因此该方法是可行有效的，具有实际应用价值。

4 结 论

尽管 P2P 和 HTTP 僵尸网络逐渐增多，但基于 IRC 的僵尸网络以其独有的特点，良好的操控性，仍然是目前已检测到的僵尸网络中最主要的形式。本文通过对获取的僵尸程序的僵尸网络控制命令分析，从命令结构的角度分析了僵尸网络控制命令与正常 IRC 频道聊天语句和正常的英文语句之间的差别，提出了从语法结构出发，定义识别僵尸网络控制命令的不同文法，分析获取 IRC 聊天频道中语句的语法结构，从而判定是否为僵尸网络控制命令，经过实验检验，达到了检测的目的。不过，该方法存在一定的局限，一旦出现非结构化命令的僵尸网络或僵尸频道使用加密通信，则检测方法失效。对于这样的问题，可以采用其它的方法辅助进行检测，如昵称相似度检测或者根据网络流量特征检测等检测方法。另外，系统将进一步在大规模网络模拟系统下

进行测试,检验方法的性能和准确性。

参考文献

[1] 诸葛建伟,韩心慧,周勇林等. 僵尸网络研究. 软件学报,2008,19(3):702-715

[2] 国家互联网应急中心. 中国互联网网络安全报告. http://www.cert.org.cn/UserFiles/File/2010 first half.pdf: CNCERT/CC, 2010

[3] Felix B, Igor S, Pablo G. et al. Challenges and limitations in current Botnet detection. In: Proceeding of the 22nd International Workshop on Database and Expert Systems Applications, Toulouse, France, 2011.95-101

[4] Goebel J, Holz T. Rishi: Identify bot contaminated hosts by IRC nickname evaluation. In: Proceedings of the First Workshop on Hot Topics in Understanding Botnets, Cambridge, USA, 2007

[5] Gu G, Yegneswaran V, Phillip P, et al. Active Botnet probing to identify obscure command and control channels. In: Proceedings of Annual Computer Security Applications Conference, Texas,USA, 2009.241-253

[6] Livadas C, Walsh R, Lapsley D, et al. Using machine learning techniques to identify Botnet traffic. In: Proceedings of the 2nd IEEE LCN Workshop on Network Security, Tampa, USA, 2006.967-974

[7] Strayer W T, Walsh R. Detecting botnets with tight command and control. In: Proceedings of the 31st IEEE Conference on Local Computer Networks, Tampa, USA, 2006.195-202

[8] Karasaridis A, Rexroad B, Hoeflin D. Wide-scale Botnet detection and characterization. In: Proceedings of the First Workshop on Hot Topics in Understanding Botnets, Cambridge, USA, 2007

[9] Binkley J R, Singh S. An algorithm for anomaly-based Botnet detection. In: Proceedings of the 2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet, San Jose, USA, 2006.43-48

[10] Gu G, Porras P, Yegneswaran V, et al. BotHunter: Detecting malware infection through ids-driven dialog correlation. In: Proceedings of the 16th USENIX Security Symposium, Boston, USA, 2007.167-182

[11] Gu G, Zhang J, Lee W. BotSniffer: Detecting Botnet command and control channels in network traffic. In: Proceedings of the 15th Annual Network and Distributed System Security Symposium, San Diego, USA, 2008.269-286

[12] Gu G, Perdisct R, Zhang J, et al. BotMiner: Clustering analysis of network traffic for protocol and structure-independent Botnet detection. In: Proceedings of the 17th USENIX Security Symposium, San Jose, USA, 2008.139-154

[13] Tsai M, Chang K, Lin C, et al. C&C Tracer: Botnet command and control behavior tracing. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Anchorage, USA, 2011.1859-1864

[14] Fedynyshyn G, Chuah M, Tan G. Detection and classification of different Botnet C&C channels. In: Proceedings of the 8th International Conference, Banff, Canada, 2011.228-242

[15] Udhayan J, Anitha R, Hamsapriya T. Lightweight C&C based Botnet detection using Aho-Corasick NFA. *International Journal of Network Security & Its Applications*, 2010,2(4):164-177

A method for identification of IRC botnets' control commands based on syntax features

Yan Jian'en, Zhang Zhaoxin, Xu Haiyan

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

Abstract

To solve the problem of detection of an IRC Botnet's control commands, a syntax feature-based identification method is presented. The method, firstly, analyzes the lexical features of keywords and parameters of IRC Botnet control commands, and then unifies them for input processing. Secondly, starting from the features of syntax structures such as the type and amount of parameters, three kinds of control commands' formalized grammar descriptions are defined to fit different syntax structures, furthermore a prototype system based on the LR parsing technique is designed and implemented. The availability of the method was verified by experiment and the experimental results showed that the grammar had the good ability in recognizing the Botnet control commands, and its performance met the pactical requirement.

Key words: Botnet, control command, syntax structure, formalization