

基于通信相似度的僵尸网络节点检测方法

金渝筌^{1,2}, 谢彬¹, 朱毅²

(1. 中国工程物理研究院电子工程研究所, 四川 绵阳 621900;

2. 四川大学, 四川 成都 610065)

摘要: 目前, 僵尸网络检测方法大多依靠对僵尸网络通信活动或通信内容的分析, 前者对数据流的特征进行统计分析, 不涉及数据流中的内容, 在检测加密类型方面具有较强优势, 但准确性较低; 后者依赖先验知识进行检测, 具有较强的准确度, 但检测的通用性较低。因此, 根据杰卡德相似度系数定义了通信相似度, 并提出了一种基于用户请求域名系统 (DNS, domain name system) 的通信相似度计算方法, 用于基于网络流量的僵尸网络节点检测。最后, 基于 Spark 框架对所提出的方法进行了实验验证, 实验结果表明该方法可以有效地用于僵尸网络节点检测。

关键词: 僵尸网络; 相似度检测; DNS 流量检测; 网络安全

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2018078

Method of botnet network nodes detection base on communication similarity

JIN Yuquan^{1,2}, XIE Bin¹, ZHU Yi²

1. Institute of Electronic Engineering in China Academy of Engineering Physics, Mianyang 621900, China

2. Sichuan University, Chengdu 610065, China

Abstract: At present, the botnet detection method mostly relies on the analysis of the network communication activity or the communication content. The former carries on the statistical analysis to the characteristic of the data flow, does not involve the content in the data flow, has the strong superiority in the detection encryption type aspect, but the accuracy is low. The latter relies on the prior knowledge to examine, has the strong accuracy, but the generality of detection is low. The communication similarity was defined according to Jaccard similarity coefficient, and a method of calculating communication similarity based on user request DNS (domain name system) was proposed, which was used for botnet node detection based on network traffic. Finally, based on the spark framework, the experimental results show that the proposed method can be used in the detection of botnet nodes effectively.

Key words: botnet, similarity detection, DNS flow detection, network security

1 引言

自僵尸网络 1993 年被首次提出至今已经过了很长时间, 但是攻击者不断更新僵尸网络的隐藏、逃匿、加密等技术, 致使研究人员对僵尸网

络的探索从未停止。僵尸网络使攻击者拥有了一个攻击平台, 利用这个平台攻击者可以发起各类攻击而造成各类安全问题: ① 发起分布式拒绝服务攻击 (DDoS, distributed denial of service), 相比拒绝服务攻击, 使用僵尸网络发起分布式拒

收稿日期: 2018-09-17; 修回日期: 2018-09-30

通信作者: 谢彬, jinfire@qq.com

绝服务攻击更具威胁,如 Mirai 僵尸网络造成了“美国断网”事件;② 发送垃圾邮件,如 Rustock 僵尸网络,其每天可发送近 300 亿垃圾邮件;③ 窃取资源,僵尸网络可以从僵尸主机中窃取包括个人信息、敏感数据、计算资源等各类网络资源;④ 恶意软件发布,僵尸网络可以用作恶意软件的发布平台,如著名的 Zeus^[1]就是恶意软件的发布平台。目前,越来越多的攻击者利用僵尸网络进行各种恶意攻击,僵尸网络的检测已成为网络安全研究人员所关注的重点之一,研究僵尸网络检测技术对于学术和工程实践均具有重要意义。

本文针对僵尸网络的通信特点,根据杰卡德相似度系数(Jaccard 系数)定义了通信相似度,并对 DNS 通信数据进行分析,提出了基于用户请求 DNS 的通信相似度的僵尸网络检测方法。且通过实验验证了该方法能够有效地用于僵尸网络节点的检测。

2 相关研究

学者在研究僵尸网络的过程中对其定义略有不同,方滨兴等^[2]认为僵尸网络是一种通过入侵网络空间内若干非合作用户终端构建的、可被攻击者远程控制的通用计算平台;诸葛建伟等^[3]认为僵尸网络是攻击者出于恶意目的,传播僵尸程序控制大量主机,并通过一对多的命令与控制信道所组成的网络;Hoque 等^[4]认为僵尸网络是指一系列由恶意实体控制感染了恶意软件的主机实体。从整体而言,僵尸网络由 3 个部分组成:僵尸主机(Bots)、攻击者(Botmaster)以及远程控制方法。僵尸主机是指对于被入侵事件并不知情的僵尸网络受控主机。攻击者是指对僵尸网络具有实际操控权的控制者。远程控制方法即命令与控制(C&C, command and control)信道,是指攻击者用来对僵尸主机发布命令的方法。僵尸网络起源于帮助 IRC 网络聊天室的管理员高效管理网络的程序“Eggdrop”^[5],而 1998 年出现的 GTBot 则是第一个恶意僵尸网络。时至今日依然有部分僵尸网络使用 IRC 协议进行通信,如 Chuck Norris、LightAidra、Tsunami 等^[6]。

僵尸网络检测技术是为了发现僵尸网络的存在,并且检测出其中部分僵尸主机节点^[2]。目前,

僵尸网络检测技术主要分为 5 种类型:基于蜜网进行捕获分析、基于通信内容的僵尸网络检测、基于异常行为的僵尸网络检测、基于安全设备日志的僵尸网络检测以及基于网络流量分析的僵尸网络检测。

基于通信内容的僵尸网络检测是指通过深度报文识别(DPI, deep packet inspect)等技术使用预先设置的特征码进行正则匹配或规则匹配,从而发现僵尸网络。该方法常用于各大厂商(如 Cisco、华为、深信服等)的防火墙等安全设备中,其检测效率高、精度准,但是需要维护庞大的僵尸网络特征库并实时更新,对新出现的僵尸网络检测效果较差。

基于异常行为的僵尸网络检测是指通过识别网络中主机的异常通信行为对僵尸网络进行检测的方法。北京邮电大学的王新良提出了一种基于指纹特征自提取的僵尸网络检测方法,并结合僵尸网络流统计特征和僵尸网络行为特征对僵尸网络进行检测,大幅度提升了僵尸网络检测率。Zeidanloo 等^[7]基于同一个僵尸网络的数据分组具有一些相同的特征这一判断,对数据分组的主机地址、端口号、数目等特征进行了分析,然后通过比较计算得到僵尸网络流量的相似性并以此进行检测。基于异常行为的僵尸网络检测由于需要对攻击行为进行预先研究和分析,不适用于检测新型僵尸网络。

基于安全设备日志的僵尸网络检测是指通过邮件网关日志、防火墙日志等安全设备日志分析,建立安全事件与僵尸网络之间的关联,从而对僵尸网络进行检测。Gu 等^[8]基于同一局域网中 Bots 网络行为在时空上的关联性假设,结合扫描、二进制文件下载、spam 等异常事件日志进行关联分析,提出了 IRC、HTTP 流量中僵尸网络的识别方法。邓国强^[9]提出了一种邮件行为异常和邮件内容相似性匹配相结合的僵尸网络检测方法,该方法不仅可以发现多个具有一定规模的僵尸网络,还能用于垃圾邮件检测。

基于网络流量分析的僵尸网络检测是通过网络流量分析,检测网络中是否存在僵尸网络 C&C 通信,并通过 C&C 信道的检测和分析定位僵尸主机,发现僵尸网络。Narang 等^[10]基于正常 P2P

通信成员与僵尸网络 P2P 通信成员之间特征不同,提出了一种对网络会话进行统计分析的检测僵尸网络的方法。此方法可以对加密的僵尸网络进行检测,但因其忽略了大量信息,对于一些隐蔽性较强的僵尸网络检测率不高。张维维等^[11]从域名的依赖性和使用位置两方面刻画 DNS 活动行为模式,并基于有监督的多分类器模型提出了一种 DAOS 算法,该算法可以有效地检测僵尸网络、钓鱼网站以及垃圾邮件等恶意活动。

在上述僵尸网络检测方法中,网络流量分析方法是一种非侵入式方法,以旁路方式部署在小型网络边界及骨干网上,既可以细粒度地监测特定用户网络,又可以并发检测僵尸网络,是当前主流的僵尸网络检测方法之一,也是本文用于研究僵尸网络节点检测的方法。

3 基于通信相似度的僵尸网络节点检测

本节首先简要介绍了僵尸网络结构、杰卡德相似系数等概念;然后基于杰卡德相似系数定义了通信相似度,并提出了用于僵尸网络节点检测的基于 DNS 流量的检测方法;最后基于 Spark 框架对提出的僵尸网络节点检测方法进行了实现和验证。

3.1 相关概念

3.1.1 僵尸网络结构

目前在僵尸网络研究中,其结构主要分为 3 种模式:集中模式、纯 P2P 模式以及混合模式,如图 1 所示。

集中模式采用客户端-服务器(C/S)模式,在该模式中所有的僵尸主机与控制服务器连接获取控制命令,如图 1(a)所示。在纯 P2P 模式中每个僵尸主机既充当僵尸程序的客户端也充当僵尸网络的服务器,如图 1(b)所示。混合模式的僵尸网络结构将僵尸网络分为多层,既具有专门的控制服务器又使用僵尸主机作为客户端和服务端,如图 1(c)所示。其中,使用分层结构混合模式的僵尸主机为本文检测内容。

3.1.2 杰卡德相似系数

杰卡德相似系数是 2 个集合交集与并集的元素数目之比,用以测量 2 个集合在共同项目上的重叠度^[12],在计算过程中,杰卡德相似系数仅计算符号度量或布尔值度量的个体间的相似度,不会考虑用户对于项目的评分取值。杰卡德相似系数常用于图像处理、目标识别、文本相似度检测、遗传相似度检测等研究领域。杰卡德相似系数等于 2 个集合 A 和 B 交集元素的个数在 A 、 B 并集中

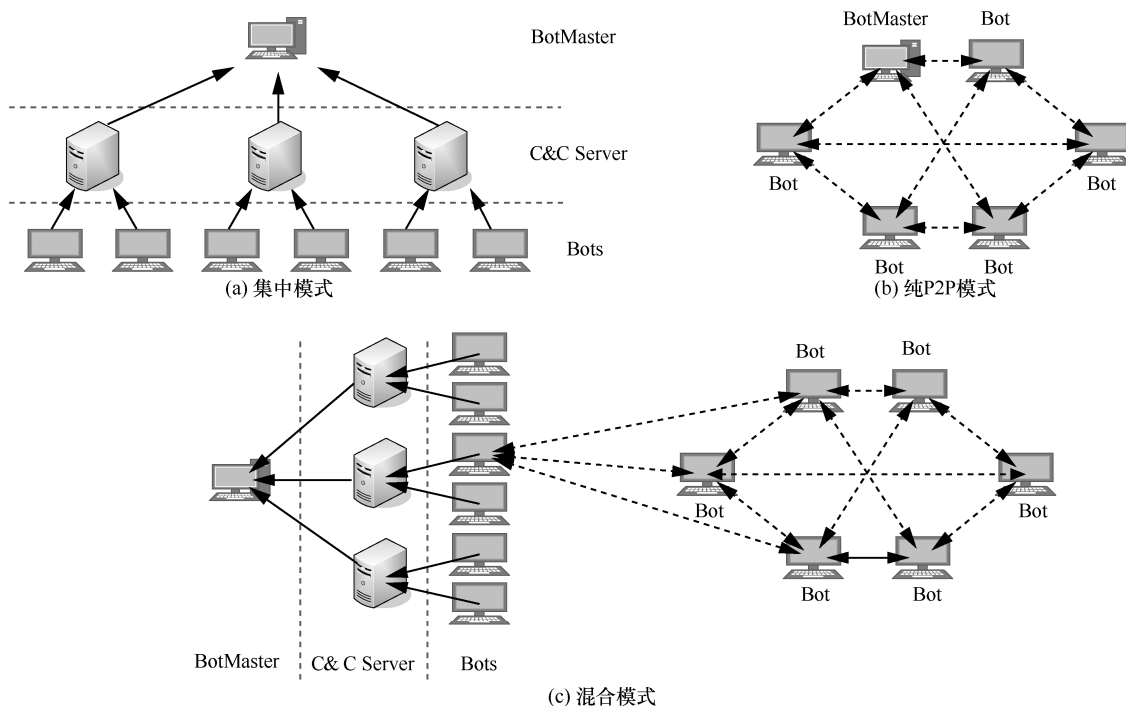


图 1 僵尸网络结构

所占的比例,用符号 $J(A,B)$ 表示,具体计算公式为

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

与杰卡德相似度相反的是杰卡德距离,杰卡德距离用来度量 2 个集合之间的差异性,它是杰卡德相似度系数的补集。假设存在集合 A 和集合 B 是 2 个 n 维向量,且所有维度的取值都是 0 或 1,则:

- 1) p 表示集合 A 和 B 都是 1 的维度个数;
- 2) q 表示集合 A 是 1 而集合 B 是 0 的维度个数;
- 3) r 表示集合 A 是 0 而集合 B 是 1 的维度个数;
- 4) s 表示集合 A 和 B 都是 0 的维度个数。

那么,集合 A 、 B 的杰卡德相似度系数可以表示为

$$J(A,B) = \frac{p}{p+q+r} \quad (2)$$

集合 A 、 B 的杰卡德距离可以表示为

$$J(A,B) = \frac{q+r}{p+q+r} \quad (3)$$

3.2 僵尸网络节点通信相似度定义

僵尸网络中的僵尸主机节点在产生共同活动(如 DDos 攻击、垃圾邮件发送等)时,其节点会产生共同或类似的网络通信行为,基于此提出了使用僵尸网络节点通信相似度对僵尸网络进行检测的方法。由于主机在通信过程中大多情况下使用 DNS 服务,因此本文的通信相似度检测基于 DNS 流量进行,即通信相似度是指客户机在 DNS 服务中请求域名的相似度。

本文中僵尸网络节点通信相似度定义如下。

僵尸网络节点通信相似度是指在 DNS 服务中 2 个客户端 IP_1 、 IP_2 请求域名及请求域名次数组成的集合 A 、 B 之间的相似性,其中 $A, B = \{(domain_1, count_1), (domain_2, count_2), \dots, (domain_n, count_n)\}$, n 表示集合 A 、 B 中域名的总个数。

对于集合 A 、 B 之间的相似性,为了方便理解和计算,本文采用杰卡德距离首先计算出集合 A 、 B 之间的差异性,然后取其补集计算杰卡德相似度。

3.3 基于通信相似度的计算方法

由于杰卡德相似度系数不考虑用户评分的影响,本文仅计算符号度量或布尔值度量的个体间

的相似度,对杰卡德相似度系数的计算进行了修改以适应僵尸网络节点通信相似度的计算。

僵尸网络节点相似度计算方式如下。

设 2 个 IP 的请求域名及请求域名次数集合为 A 和 B , A 、 B 集合中共有 n 个域名,对任意域名 d 满足 $d \in A \cup B$, 域名的编号为 i , 则域名 d 的差异值 $diff(i)$ 计算方法为

1) 当 $d \in A \cap B$ 时, 域名 d 的差异值为 $diff(i) = |count_A - count_B|$;

2) 当 $d \in A - B$ 时, 域名 d 的差异值为 $diff(i) = |count_A|$;

3) 当 $d \in B - A$ 时, 域名 d 的差异值为 $diff(i) = |count_B|$ 。

设 $Sum(A.count)$ 表示 A 集合中 $count$ 的和, $Sum(B.count)$ 表示 B 集合中 $count$ 的和, 则集合 A 、 B 的差异度计算如式(4)所示, 集合 A 、 B 的相似度计算如式(5)所示。

$$D = \frac{\sum_{i=1}^n diff(i)}{Sum(A.count) + Sum(B.count)} \quad (4)$$

$$S = 1 - \frac{\sum_{i=1}^n diff(i)}{Sum(A.count) + Sum(B.count)} \quad (5)$$

3.4 基于 Spark 的计算方法实现

本文使用 Spark 实现僵尸网络节点的通信相似度计算,其计算流程如图 2 所示。首先从 DNS 流量中获取所需信息,包括客户端 IP、请求域名以及请求域名次数;然后将提取的信息以客户端 IP 为主要键值进行笛卡尔积计算形成客户端与集合的数据集,接着分别计算每对节点的通信相似度 S ;最后过滤出节点通信相似度大于 0.9 的 IP 对作为最终输出结果。

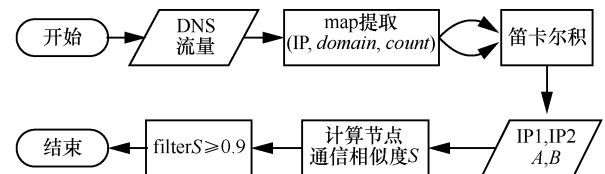


图2 节点通信相似度计算流程

在僵尸网络节点相似度计算过程中,计算量往往较大,而且在计算过程中会使用笛卡尔积进行数据关联操作,这会使计算数据以平方倍数增加,造成无法计算或计算时间过长的问题。针对

该问题，本文对僵尸网络节点相似度在 Spark 中的计算流程进行优化，使计算时间下降到可计算范围内。优化后的节点通信相似度计算流程如图 3 所示。

算法实现的具体步骤如下。

步骤 1 读取 DNS 流量。

步骤 2 提取 DNS 流量中的有效信息，包括客户端 IP、请求域名、客户端请求域名次数以及客户端请求 DNS 次数。

步骤 3 根据客户端请求 DNS 次数的对数结果将数据进行分块。

步骤 4 对每一个分块计算其自身以及相邻分块的笛卡尔积。

步骤 5 对步骤 4 的计算结果求取相似度 S 。

步骤 6 获取相似度 S 不小于 0.9 的数据计算结果，并将所有计算结果进行 union 操作。

步骤 7 输出最终计算结果，节点通信相似度计算完毕。

本文主要考虑通过 IP 请求 DNS 的次数对数据进行分割，以减少进行笛卡尔积计算后数据的增量。本文中的节点通信相似度计算在杰卡德相似度系数计算基础上充分考虑了用户的评分情

况，其计算结果与用户使用 DNS 的次数呈正相关，当用户使用次数具有较大差异时，其节点通信相似度不可能较大。基于此，某 DNS 服务器在 2017 年 2 月 1 日至 2 月 10 日对用户请求分布进行了统计，统计结果如图 4 所示。

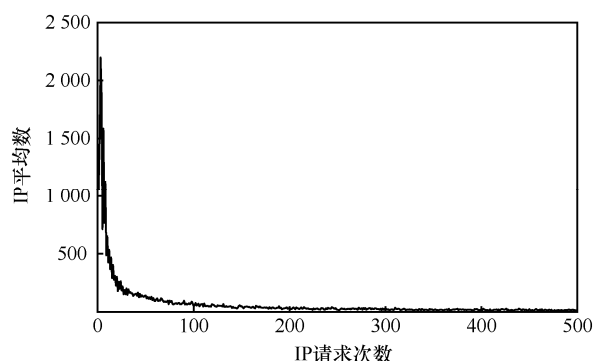


图 4 IP 请求次数与 IP 平均数分布

由图 4 可以看出，IP 请求 DNS 服务次数与 IP 数目呈现明显的幂律分布，因此本文采用 IP 请求次数的对数进行数据分割。再使用分割后的数据集求取其自身与相邻分割模块的笛卡尔积，然后计算其相似度并按照相似度大于 0.9 的条件对处理数据进行筛选，最后将处理结果进行 union

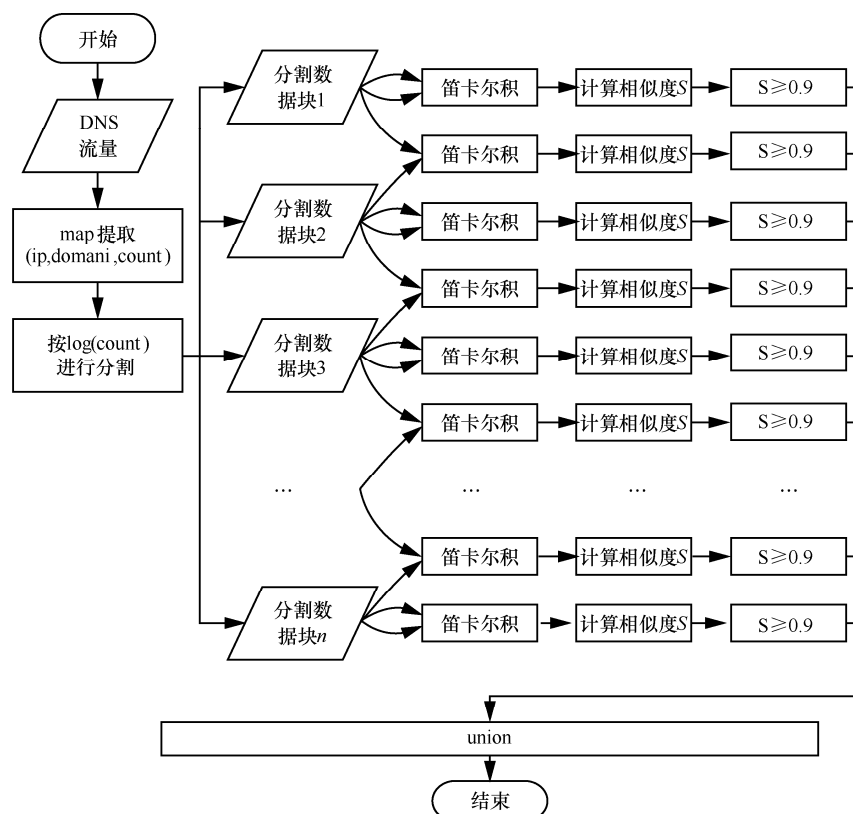


图 3 基于 Spark 的通信相似度计算实现流程

操作, 输出最终的节点通信相似度检测结果。

3.5 计算方法有效性验证

3.5.1 实验数据说明

本文中所有数据采集、实验过程, 以及使用的 DNS 流量数据均基于网络业务与安全态势大数据分析平台^[13]。该平台部署了 Hadoop 实现对网络流量的大规模存储, 同时通过 Spark、MapReduce 等计算框架实现网络安全大数据分析。对于数据存储主要使用 HDFS、Hive、MySQL 等存储组件, 其中, HDFS 和 Hive 负责大规模数据的分布式存储, MySQL 用于存储平台分析结果和少量分析配置信息。

3.5.2 节点通信相似度有效性验证实验

1) 实验数据集

为了验证算法的有效性, 本文使用 2017 年 3 月 21 日的 DNS 流量数据进行计算, 网络业务与安全态势大数据分析平台显示当日数据出现了小规模 DNS 异常请求, 其为 DDoS 攻击事件。在当日数据中发现有 24 个 IP 在 DNS 请求上具有极强的通信相似性, 如图 5 所示, 图中 24 个 IP 大量访问部分域名, 且访问内容具有很强的相似性。此次攻击与本文提出的僵尸网络节点通信的相似度非常相似。

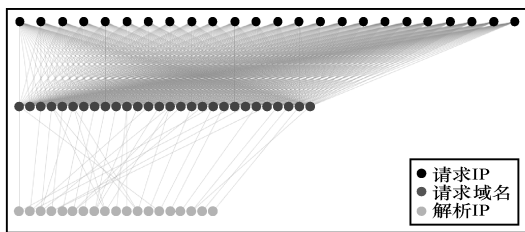


图5 2017年3月21日疑似DDoS攻击事件

2017年3月21日的DNS流量数据规模如表1所示。

表1 节点通信相似度有效性验证数据规模

DNS 流量	数据规模
总 DNS 回执信息数	220 308 358
总 DNS 数据存储量	约 15 GB
请求 IP 数	94 965

2) 实验结果与分析

根据本文提出的节点通信相似度算法, 对 2017 年 3 月 21 日的 DNS 流量进行统计, 并以 IP 为点, 相似度 S 不小于 0.9 为边建立关系图, 如

图 6 所示。图中圈内为 24 个根据网络业务与安全态势大数据分析平台找出的 DDoS 攻击 IP, 即僵尸网络节点。

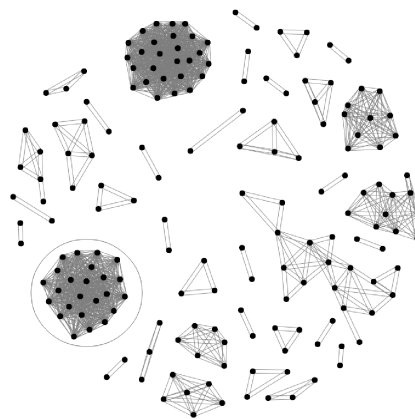


图6 节点相似度关系

由此, 通过本文提出的节点通信相似度算法可以检测出僵尸网络。此外, 圈外还存在大量 IP 相似关系, 经分析其行为与 24 个事先找出的 IP 行为十分相似, 但访问量低, 非 DDoS 攻击, 即非僵尸网络节点。综上, 本文提出的节点通信相似度算法是一种有效的僵尸网络节点检测方法。

3.5.3 优化算法有效性验证实验

1) 实验数据与环境

为了验证优化算法的有效性, 本文使用 2017 年 3 月 21 日的 DNS 流量数据进行计算, 其数据规模如表 1 所示。为进行算法优化先后的对比, 需要保证优化前后算法运行环境的一致性, 因此本实验中对 Spark 程序的运行环境进行了限定, 具体配置如表 2 所示。

表2 Spark 程序运行环境

运行的项目	规模
driver 个数	1
driver 内存	4 GB
driver 内核数	3
executor 个数	6
单个 executor 内存	13 GB
单个 executor 内核数	3

2) 评价标准

本文对节点通信相似度算法的优化既是对算法本身优化, 又包括在使用 Spark 过程中

对 Spark 程序的优化。针对算法本身优化，采用计算速度提升率来衡量，设优化前算法耗时为 T_1 ，优化后算法耗时为 T_2 ，则算法速度提升率为

$$\Delta v = \frac{T_1 - T_2}{T_2} \times 100\% \quad (6)$$

针对 Spark 程序的优化，本文考虑在 Spark 运行过程中 shuffle 量的减少率。在 Spark 运行过程中执行 group 等操作时，需要把相同键值的元素聚集到同一个 partition 下，造成了数据在内存中的重新分布，这一过程称为 shuffle 操作。shuffle 操作是 Spark 中最耗时的操作，应该尽量减少 shuffle 操作。设算法优化前 shuffle 量为 a ，优化后 shuffle 量为 b ，shuffle 量的减少率为

$$\Delta s = \frac{a - b}{a} \times 100\% \quad (7)$$

3) 实验结果与分析

根据本文提出的节点相似度算法及其优化算法，分别对 2017 年 3 月 21 日 DNS 流量进行统计，其时间消耗如图 7 所示，根据式(6)计算可得其算法速度提升率为 1756.60%，因此，本文提出的优化方案大大降低了算法时间消耗。

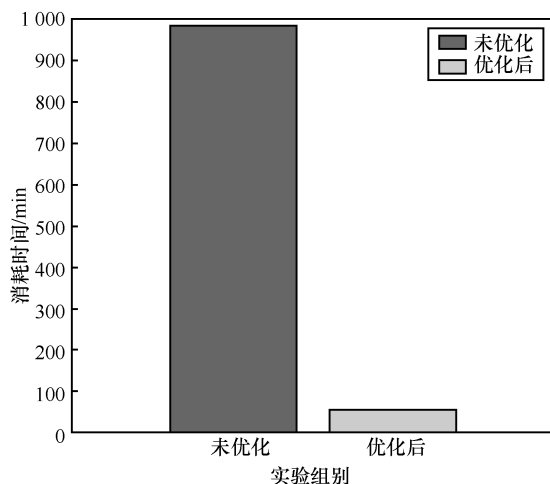


图 7 计算时间消耗

对于 Spark shuffle 消耗，本实验也做了相关统计，其结果如图 8 所示，根据式(7)计算得出 shuffle 减少率为 83.33%，因此，本文提出的优化方案大大降低了在 Spark 计算中的 shuffle 消耗。综上所述，本文提出的僵尸网络节点相似度优化算法是有效的。

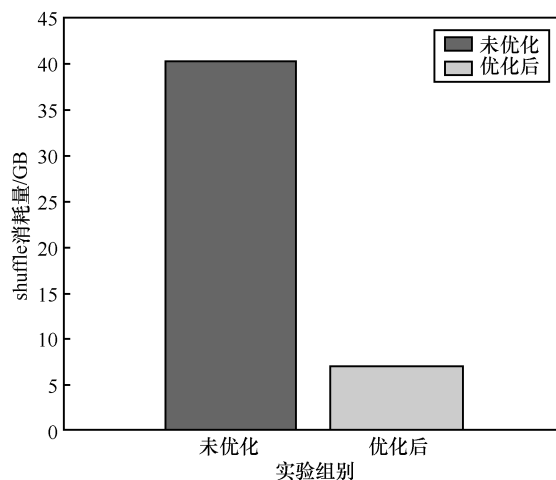


图 8 shuffle 消耗量

4 结束语

本文根据杰卡德相似度系数定义了通信相似度，并提出了一种基于用户 DNS 流量请求的通信相似度计算方法，用于僵尸网络节点检测。该方法基于 Spark 框架进行了实现，并基于网络业务与安全态势大数据分析平台的网络流量监控数据进行了验证，验证结果表明该方法可以有效地用于僵尸网络节点的检测。

本文中对节点通信相似度的计算以及拓扑结构分析过程，由于数据量庞大、特征时间跨度较大以及算法未能分布式实现等问题，仅对单日的数据进行检测与发现。对于威胁性更大、处于爆发阶段的僵尸网络需要在尽量短的时间内实现其检测和发现，因此，未来将针对算法和特征进一步实现优化，使其可以分布式、实时地对僵尸网络进行检测。

参考文献：

- [1] FALLIERE N, CHIEN E. Zeus: king of the bots, 2009[EB/OL]. http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/zeus_king_of_bots.pdf. 2014: 07-30.
- [2] 方滨兴, 崔翔, 王威. 僵尸网络综述[J]. 计算机研究与发展, 2011, 48(8): 1315-1331
FANG B X, CUI X, WANG W. Survey of botnets[J]. Journal of Computer Research and Development, 2011, 48(8): 1315-1331.
- [3] 诸葛建伟, 韩心慧, 周勇林, 等. 僵尸网络研究[J]. 软件学报, 2008, 19(3): 702-715
ZHUGE J W, HAN X H, ZHOU Y L, et al. Research and Development of Botnets[J]. Journal of Software, 2008, 19(3): 702-715
- [4] HOQUE N, BHATTACHARYA D K, KALITA J K. Botnet in DDoS attacks: trends and challenges[J]. IEEE Communications Surveys & Tutorials, 2015, 17(4): 2242-2270.

- [5] IRC Eggdrop: open source bot[EB/OL]. <http://www.eggheads.org/>, 1993.
- [6] 李可. 基于行为分析的僵尸网络对抗技术研究[D]. 北京邮电大学, 2017.
LI K. Research on botnet countermeasure technology based on behavior analysis[D]. Beijing University of Posts and Telecommunications, 2017.
- [7] ZEIDANLOO H R, MANAF A B A. Botnet detection by monitoring similar communication patterns[J]. International Journal of Computer Science & Information Security, 2010, 7(3): 88-96.
- [8] GU G, ZHANG J, LEE W. BotSniffer: detecting botnet command and control channels in network traffic[J]. Computer Science and Engineering, 2008.
- [9] 邓国强. Spamming botnet 网络行为分析及检测算法设计与实现[D]. 华中科技大学, 2013.
DENG G Q. Network behavior analysis of the spamming botnet and the design and implementation of spamming botnet detection algorithms[D]. Huazhong University of Science and Technology, 2013.
- [10] NARANG P, RAY S, HOTA C, et al. PeerShark: detecting peer-to-peer botnets by tracking conversations[J]. Medical & Biological Engineering & Computing, 2014, 51(10):1105-1119.
- [11] 张维维, 龚俭, 刘尚东, 等. 面向主干网的 DNS 流量监测[J]. 软件学报, 2017, 28(9): 2370-2387.
ZHANG W W, GONG J, LIU S D, et al. DNS surveillance on backbone[J]. Journal of Software, 2017, 28(9): 2370-2387.
- [12] 嵇威华, 吕国芳. 基于广义 Jaccard 系数处理冲突证据方法[J]. 控制工程, 2015(1): 98-101.
- JI W H, LV G F. Conflicting evidence combination method based on generalized Jaccard coefficient[J]. Control Engineering of China, 2015(1): 98-101.
- [13] 王煜骢, 陈兴蜀, 罗永刚, 等. NTCI-Flow:一种可扩展的高速网络流量处理框架[J]. 四川大学学报(工程科学版), 2017(s1): 168-174.
WANGY C, CHEN X S, LUOY G, et al. NTCI-Flow: a scalable high-speed network traffic processing framework[J]. Journal Sichuan University (Advanced Engineering Sciences) 2017(s1): 168-174.

作者简介



金渝笙 (1979-), 男, 四川三台人, 硕士, 中国工程物理研究院电子工程研究所工程师, 主要研究方向为网络安全和应用软件安全。

谢彬 (1966-), 女, 四川安岳人, 硕士, 中国工程物理研究院电子工程研究所高级工程师, 主要研究方向为信息安全、软件安全、安全保密测评。

朱毅 (1991-), 男, 四川内江人, 四川大学硕士生, 主要研究方向为网络安全、大数据分析。