基于Storm实时流式计算框架的网络日志分析方法

杨立鹏 张仰森 张 雯 王 建 曾健荣
(北京信息科技大学智能信息处理研究所 北京 100101)

摘 要 随着互联网的飞速发展,网络日志数据呈现爆炸式增长,网络日志蕴含着丰富的网络安全信息。通过对网络日志进行分析,提出了基于访问行为和网络关系的攻击IP识别模型和基于滑动时间窗口的IP真人属性判定模型。基于Storm实时流式计算框架,对所提模型进行算法实现,以构建分布式网络日志实时计算与分析平台,并对实现过程中遇到的技术问题给出了解决方案。通过真实数据对所构建的模型进行分析计算,结果表明,所构建的攻击IP识别模型的标注准确率达到98%,IP真人属性判定模型的标注准确率达到96%;构建的分布式网络日志实时计算与分析平台能够有效、实时地监控网络安全,并及时识别网络中存在的安全隐患。

关键词 Storm,IP真人率,攻击IP识别,分布式网络日志分析平台

中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.09.025

Web Log Analysis Method Based on Storm Real-time Streaming Computing Framework

YANG Li-peng ZHANG Yang-sen ZHANG Wen WANG Jian ZENG Jian-rong
(Institute of Intelligent Information Processing, Beijing Information Science and Technology University, Beijing 100101, China)

Abstract With the rapid development of the Internet, the network log data in the Internet show explosive growth, and the network log contains a wealth of network security information. By analyzing network log, this paper proposed an attack IP recognition model based on access behavior and network relationship and an IP real person attribute decision model based on sliding time window. Based on the Storm real-time flow computing framework, the proposed model was implemented in order to construct a real-time computing and analysis platform for distributed network logs, and a solution to the technical problems encountered in the implementation process was given. Through the analysis and calculation of the constructed model through real data, the results show that the accuracy of the constructed attack IP identification model is 98%, the accuracy rate of the IP real property judgment model reaches 96%, and the constructed distributed network log real-time computing and analyzing platform can effectively and timely monitor network security and timely identify potential security risks in the network.

Keywords Storm, IP real rate, Attack IP identification, Distributed network log analysis platform

1 引言

网络日志包含着丰富的数据信息,对这些数据进行挖掘和分析有助于及时发现网络的安全隐患,因此网络日志数据的采集与分析一直是国内外专家学者研究的重点和热点。中国互联网络信息中心(CNNIC)发布了第41次《中国互联网络发展状况统计报告》,报告中显示截至2017年12月,中国互联网的网民规模已经达到了7.72亿,该数字超过了中国总人口的一半。随着互联网公司产品的不断升级,公司产品的访问量直线上升,这会产生很多的日志信息。根据日志信息,我们可以查找系统开发过程中的错误,从而进行系统恢复,并检验系统运行的任务是否正常等。此外,我们还可以对日志进行实时分析,挖掘日志信息中的潜在价值。

对互联网的访问会产生巨额的日志信息,使得对日志的收集、处理、存储等都面临着巨大的挑战。在实际社会环境中,一个企业应用系统在运行过程中会打印很多日志信息,这些日志信息会存储在不同文件中,因此单一节点监控日志文件已经不能够满足企业的需求,需要使用分布式的日志文件收集。由于对大量的非结构化数据进行实时处理也具有挑战性,因此单机模式的处理亦不能满足企业需求,必须进行分布式、实时的收集处理。

为应对网络日志处理过程中的困难,本文将建立一个具有高效、实时可用、稳定性强的网络日志分析处理平台;同时,提出了基于网络IP访问行为和网络关系的攻击IP识别模型与基于滑动时间窗口的IP真人属性判定模型,进一步对网络攻击IP进行细粒度的属性提取与分析,以实时监控网络安

到稿日期:2018-07-04 返修日期:2018-10-08 本文受国家自然科学基金(61772081)资助。

杨立鹏(1991-),男,硕士生,主要研究方向为大数据处理;**张仰森**(1962-),男,博士后,教授,CCF高级会员,主要研究方向为中文信息处理、大数据处理,E-mail:zhangyangsen@163.com(通信作者);**张 雯**(1997-),女,硕士生,主要研究方向为大数据处理;**王 建**(1993-),男,硕士生,主要研究方向为大数据处理、事件检测;**曾健荣**(1993-),男,硕士生,主要研究方向为大数据处理。

全，及时识别网络中的安全隐患。

2　相关工作

在对网络攻击 IP 进行识别[1]方面，目前已经取得了不少成果，主要可以分为以下 3 种网络攻击 IP 识别方法。

(1)基于特征识别的网络攻击 IP 识别

首先，通过对大量的网络日志数据进行训练、学习，生成网络攻击 IP 规则集合。如果某个模式与规则集合中某条规则互相匹配，则认为该模式属于该规则所对应的攻击行为[2]。Mazzariello 等将这种分析方法应用于云环境中，用于识别 DOS 攻击(拒绝服务攻击)[3]。这种分析方法也可同步应用于异常入侵检测，通过构建高扩展性的入侵检测架构检测异常入侵，并支持虚拟化。基于特征识别的日志分析方法的优势是识别攻击行为的正确率较高，误报率较低，且易于实现和管理，比如规则集合可以随时进行更新；缺陷是只能识别已知的攻击行为，无法处理新型攻击方式。

(2)基于异常检测的网络攻击 IP 识别

通过数据挖掘、统计模型和隐马尔可夫模型等对日志数据进行处理[4]，并根据处理结果对观测到的行为模型进行判断，将判断结果划分为常规行为或者攻击行为。2007 年，Dutkevyc 就使用这种分析方法对多维流量日志进行分析，为高速企业网络提供实时入侵检测服务。Zheng 等[5]于 2007 年提出了轻量级入侵检测系统，其原理也是基于异常检测来对日志数据进行分析处理。该系统能够实时、高效地进行入侵检测。与基于特征识别的分析方法相比，本文方法的主要优势是能够处理未知攻击行为，误报率也较低；但其识别已知攻击行为的正确率较低，并且更难实现。

(3)基于关联规则的网络攻击 IP 识别

有些攻击行为包含多种已知攻击形式，或者已知攻击形式的多个变种。为了应对这种攻击，可以使用基于关联规则的日志分析方法[6]。其核心思想是：对日志数据进行处理和计算，经过候选集生成和情节向下封闭检测两个阶段，根据支持度大小挖掘出频繁项集，然后根据置信度大小来生成关联规则，最终根据关联规则来进行攻击行为的检测。2002 年，Han 等[7]提出了一种入侵检测技术，首先通过特征生成算法生成大量的特征，然后生成特征频繁项集和关联规则，最终根据关联规则来检测入侵行为。然而，该方法所用到的特征生成算法的时间消耗比较严重。研究者们发现，可以通过减少数据项扫描次数的算法来解决性能问题，但是该算法产生了很多无效模式，误报率较高。孙学波提出的基于 Hadoop 的 Apriori 算法研究与优化[8]可以减少短模式(Short Pattern)的产生，并且能解决传统数据挖掘算法在处理大量数据时面临的内存占用、计算性能等方面的问题。基于关联规则的分析方法简单易理解，并且能发挥预测的作用；但是 I/O 负载大，容易产生过多的候选项目集，增加存储负担。

上述方法虽然都对识别网络攻击 IP 做出了贡献，但是少有研究对网络 IP 进行细粒度的属性提取分析以及攻击 IP 真人率的判定。本文对网络攻击 IP 识别进行研究，同时对网络攻击 IP 是否为真人攻击进行研究。

在利用大数据平台处理网络日志技术研究方面，Storm 具有较低的时延，并且 Storm 是一种实时、高效的分布式计算框架[9]，能实现流数据的处理，且处理数据的时延仅在毫秒级别。Storm 可以读取很多数据流信息，能完美地与 Kafka 和 Hive 结合。Kafka[10]具有高吞吐量、低延迟的特点，每秒可以处理几十万条消息；Hive[11]可以实现海量数据的持久化。在实时处理的框架中，Storm 是一个非常优秀的框架，国内外很多公司都使用它来完成相应的任务：淘宝在生产环境中使用 Storm 做实时日志统计，并从分析的日志中抽取相应的信息；百度公司使用 Storm 来处理用户使用搜索引擎产生的搜索日志，实现用户的 PV 实时统计；Twitter 使用 Storm 来实时处理消息发布者的粉丝的动态行为，实现舆情的动态监控。

3　基于 Storm 计算框架的网络日志分析方法

网络日志中蕴含着丰富的网络安全信息，对网络日志进行分析，识别出具有攻击性的网络 IP，同时对攻击 IP 真人属性进行判定，是网络日志分析的主要内容之一。本文对网络 IP 进行实时统计分析，通过用户的网络动作和网络关系来识别出破坏网络安全的重点 IP，并对识别出的重点 IP 进行细粒度的属性提取分析，工作流程图如图 1 所示。

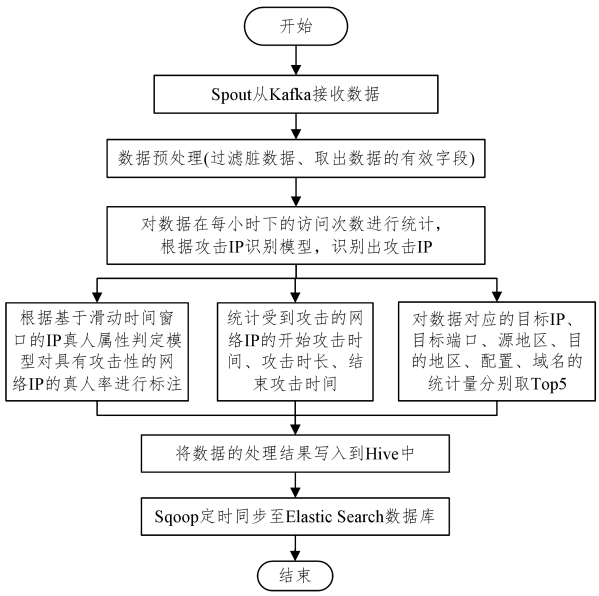


图 1　系统流程图

Fig. 1　System flowchart

3.1　基于访问行为和网络关系的攻击 IP 识别模型

网络攻击 IP 识别模型的构建需要网络 IP 访问行为和网络 IP 的网络关系两个特征，因此本文首先采用 Storm 技术对网络日志数据进行预处理，并以小时为单位划分时间窗口，对源 IP 的访问量和访问关系进行统计。然后基于本文提出的基于访问行为和网络关系的攻击 IP 识别模型准确、实时地对攻击 IP 进行识别，并对其进行分析与监控。网络 IP 访问量及其访问关系属性提取算法如算法 1 所示。

算法 1　网络 IP 访问量及其访问关系属性提取算法

输入：KafkaSpout 数据流中的所有数据字段

输出：发射包含所有输入字段并且增加网络 IP 访问量和目标 IP 访问量的 Tuple

Step1 接入 Spout 预处理后的数据流，取出本算法所需的 Field，进行访问量与访问关系的统计，并定义本算法所需的 keyword。

Step2 判断接入数据流中组件信息的类别。如果组件信息为系统组件，则清除系统内存中的数据，释放存储空间，跳到 Step5；如果为数据处理组件，则跳到 Step3。

Step3 数据处理组件：按照 keyword 对数据流中的数据进行统计，并将统计结果存储于构造的 KeysValuesMap 中，若该 keyword 已经是 KeysValuesMap 的 key，则 value 加 1；否则，将该 keyword 作为 KeysValuesMap 的 key，并把 value 赋值为 1。

Step4 在 keyword 的基础上对数据流中的数据增加访问目标 IP 字段，构建 keyword2 并进行数据统计，将统计结果存储于 KeysValuesMap 中，若该 keyword2 已经是 KeysValuesMap 的 key，则 value 加 1；否则，将该 keyword2 作为 KeysValuesMap 的 key，并将 value 赋值为 1。

Step5 系统组件：数据处理组件运行时间达到预先设定的时长后，若需计算网络攻击的 IP 攻击时长，则转入算法 3；若需计算网络攻击的 IP 各属性 TOP N，则转入算法 4。

Step6 算法结束。

为了保证数据不会被重复计算，本模块将网络 IP 和网络日志采集时间（日期 dd、小时 hh）作为本系统的 keyword，并且采用 Field grouping 对 keyword 进行字段分组，以保证同一个 keyword 被发射到同一个 Bolt 的 task 中进行数据处理与计算。经过计算后，输出原有的字段，并且增加 IP 访问量和 IP 访问目标的集合字段，并根据网络攻击 IP 识别模型判断攻击 IP；同时将识别出的网络攻击 IP 数据发射给下一个 Bolt，进行网络攻击 IP 属性提取与分析，实现对网络攻击 IP 进行监控的社会效益。

网络攻击 IP 的产生会使网络流量在短时间内发生突增现象，并且在短时间内不会削减，趋于一种高峰稳定状态。通过网络 IP 访问流量特征能够识别爆发性增长的源 IP 攻击，而通过 IP 访问关系网络信息熵特征能够对少量但大规模分散访问的攻击 IP 进行识别。因此，本文将用户访问流量特征和用户信息熵[12]特征进行加权，构建基于多特征的网络 IP 攻击行为的辨识模型。

设网络 IP 地址为 x，t 为当前时间，引入滑动时间窗口 T，$df_t(x,y)$为用户 IP 在当前时间 t 下的攻击函数，用以衡量用户 IP 地址在当前时间是否属于攻击行为。$df_t(x,m)$的计算如式(1)所示：

$$df_t(x,m)=\begin{cases}0, & C_t(x)<C_{\text{avg}}(t)+m\times C_{sd}(t)\\ 1, & C_t(x)\geqslant C_{\text{avg}}(t)+m\times C_{sd}(t)\end{cases} \tag{1}$$

其中，$C_t(x)$为网络 IP 地址 x 在当前时间下的访问量，$C_{\text{avg}}(t)$为当前时间 t 下的全部网络 IP 访问量的平均值，$C_{sd}(t)$为当前时间 t 的访问量标准差，m 为攻击流量异常参数，取 $m=3$。攻击函数可以识别出当前时间爆发性的攻击行为，但无法识别高持续性的非爆发性攻击行为。$cf_t(x,n)$为网络 IP 地址在一段时间 T 内的持续流量函数，用于检测那些持续高频访问的用户 IP 地址。$cf_t(x,n)$的计算式如式(2)所示：

$$cf_t(x,n,T)=\begin{cases}0, & \sum_{i=t}^{t-T}df_t(x,n)<T\\ 1, & \sum_{i=t}^{t-T}df_t(x,n)\geqslant T\end{cases} \tag{2}$$

其中，n 为高持续性流量异常参数，取 $n=2$；T 为时间窗口大小。

攻击函数以及持续流量函数值以网络源 IP 地址为分析对象得到，识别出“单”用户的攻击行为。若仅从用户 IP 角度出发，无法识别出“多对一”的攻击流量，即少量但大规模分散访问的恶意行为。因此，引入网络信息熵的概念来针对群体攻击行为进行识别。信息熵的物理意义是体系混乱程度的度量。通过计算单位时间内网络报文目的 IP 的熵值并度量被访问地址的离散程度，可以检测恶意流量。

假设用户 IP 地址为 x，t 为时间，集合 $D=\{mdip_1,mdip_2,\cdots,mdip_n\}$为第 t 时刻用户 IP 地址 x 访问的所有目的 IP 地址集合，集合 $P=\{p_1,p_2,\cdots,p_n\}$为集合 D 中目的 IP 地址在第 t 时刻所处网络的概率分布，p_i 表示目标 IP 地址 i 在当前时刻流量中的占比，则可得到网络 IP 地址对应所有目的 IP 地址的熵特征值，以 $e_t(D)$表示，如式(3)所示：

$$e_t(D)=-\sum_{i=1}^{n}p_i\ln p_i \tag{3}$$

群体攻击流量的产生会导致网络日志中被访问的目的地址比较集中，此时网络关系中目的地址 IP 的信息熵与正常网络流量的熵相比会相对减少。根据式(3)的计算结果，网络群体攻击流量的熵远远小于正常网络流量的熵。因此，将信息熵作为判断 IP 网络攻击的标准之一。同样地，利用类似攻击函数的判断方法构建熵函数 $ef_t(x,m')$，如式(4)所示，用于衡量网络流量的熵是否异常。

$$ef_t(x,m')=\begin{cases}0, & e_t(D)\geqslant e_{\text{avg}}-m\times e_{sd}(D)\\ 1, & e_t(D)<e_{\text{avg}}-m\times e_{sd}(D)\end{cases} \tag{4}$$

综合网络 IP 产生的攻击流量特征和用户信息熵的特性对爆发性和高持续性恶意访问行为进行识别。将 IP 用户的攻击流量特征与用户的信息熵特征进行加权，构建网络用户恶意访问行为的判定模型。

设用户 IP 地址为 x，t 为当前时间，通过计算网络 IP 地址 x 的攻击行为异常值，用 $A_t(x)$表示，可判断用户 IP 地址 x 是否为攻击 IP。构建网络 IP 攻击行为识别函数 $A_t(x)$，如式(5)所示：

$$MA_t(x)=\frac{1}{2}ef_t(x,m)\times cf_t(x,n,T)+\frac{1}{2}ef_t(x,m') \tag{5}$$

其中，为了将攻击行为异常值归一化，将各特征的权重设为 0.5。$A_t(x)$的取值范围为 0 到 1，表示用户 IP 地址的攻击行为概率，该值大于或等于 0.5 时表示网络 IP 产生攻击行为，需要进行重点监测和追踪。上式综合了网络 IP 访问行为特性和 IP 所在网络熵的特性，既避免了单独分析用户行为特性造成的误报，也考虑了单独分析源 IP 可能造成的“多对一”大流量访问异常情况漏报。

3.2 基于滑动时间窗口的 IP 真人属性判定模型

网络日志 IP 真人属性提取采用基于滑动时间窗口的 NHT 识别技术[13]，将网络 IP 真人属性的二值改进为基于时间窗口的真人概率判断，使 IP 真人属性的描述更加准确。

对网络日志的访问量进行统计分析，设 t 为当前时间，网络 IP 地址设为 x，提取第 t 天的网络日志，根据以下 3 条规则判断 x 是否为真人 IP。

Rule(1)　IP 为知名公司的爬虫 IP、CDN 服务 IP、云服务 IP、数据中心 IP。

Rule(2)　每小时的 IP 访问量超过 α 次。

Rule(3)　日活跃时长超过 β 小时。

判定第 t 天某个 IP 的真人值如式(6)所示:

$$h_t(x)=\begin{cases}0, & \text{满足上述规则之一,判定为机器}\\ 1, & \text{不满足上述规则,判定为真人}\end{cases} \tag{6}$$

在本次实验中,设置 α 的值为 3600,即每小时的访问量不超过 3600 次,设置 β 的值为 20,即日活跃时长不超过 20 h。满足上述规则之一则判定该 IP 为机器,不满足上述规则则判定其为真人。然后引进滑动时间窗口 T,本次实验设置的滑动时间窗口大小为 30(即最近 30 天的历史数据),将历史数据与当前数据进行加权融合,以提升 IP 真人属性标注的准确率。

设当前 IP 地址为 x,t 为当前时间,时间窗口为 T,d 为时间窗口 T 的元素值,则基于滑动时间窗口的 IP 属性真人率的计算式如式(7)所示:

$$H_t(x)=\sum_{d=1}^{T}\frac{1}{2^d}h_{t-d+1}(x) \tag{7}$$

基于滑动时间窗口的 IP 真人属性提取算法如算法 2 所示。

算法 2　基于滑动时间窗口的 IP 真人属性提取算法

输入:KafkaSpout 数据流中的所有数据字段

输出:发射包含所有有效字段并且增加真人值字段的 Tuple

Step1　根据构建的恶意访问 IP 知识库判断该 IP 是否满足 Rule(1),若满足 Rule(1)则将真人值字段标记为 0,跳转到 Step5,否则跳转到 Step2;

Step2　统计当前日期的小时级 IP 访问量,如果小时级的 IP 访问量超过 α,则将真人值标记为 0,跳转到 Step5,否则跳转到 Step3;

Step3　统计当前日期的日活跃时长,如果日活跃时长超过 β,则将真人值标记为 0,跳转到 Step5,否则跳转到 Step4;

Step4　将真人值标记为 1,跳转到 Step5;

Step5　引入滑动时间窗口 T(如最近 30 天),计算当前 IP 基于时间窗口的概率性 IP 真人率,如式(7)所示;

Step6　算法结束。

3.3　网络攻击 IP 的统计类属性提取与分析

本文构建的网络攻击 IP 识别模型,能够高精度地识别出具有攻击性的网络 IP。为了更好地对网络安全进行监控,本文对识别出的网络攻击 IP 的攻击时长以及网络攻击 IP 的端口、地区、配置和域名进行细粒度的属性提取刻画。

3.3.1　网络攻击 IP 攻击时长属性的提取

接入网络攻击 IP 识别 Bolt 的数据流,同时构建该模块的 keyword,数据流分组策略采用 Field grouping。通过分析网络日志,我们无法得到网络攻击 IP 的结束攻击时间。因此,本文提取网络攻击 IP 最早的一次访问时间将其确定为网络攻击 IP 的开始攻击时间;提取出网络攻击 IP 最晚的一次访问时间并将其确定为网络攻击 IP 的结束攻击时间,两者之差即为网络攻击 IP 的攻击时长。提取 Tuple 中的时间字段,其中提取到的时间为 UNIX 时间,需要对时间字段进行转换以及格式化,然后以构建的 keyword 为主键存入自定义的 Map 中。自定义的 Map 有两个 Value 值,Value1 为开始攻击时间,Value2 为结束攻击时间。网络攻击 IP 攻击时长属性的提取算法如算法 3 所示。

算法 3　网络攻击 IP 攻击时长属性提取算法

输入:网络攻击 IP 识别 Bolt 数据流中的日期(dd)、时间(hh)、IP

输出:发射包含网络攻击 IP 开始攻击时间,攻击时长和结束攻击时间的 Tuple

Step1　接入网络攻击 IP 识别 Bolt 处理后的数据流,取出本算法所需的 Field,计算网络攻击 IP 攻击的时长,并且定义本算法的 keyword。

Step2　判断接入数据流中组件信息的类别,如果组件信息为系统组件,则清除系统内存中的数据,释放存储空间,跳到 Step5;如果组件信息为数据处理组件,则跳到 Step3。

Step3　数据处理组件:将数据流中取出的 UNIX 时间戳转换为 JAVA 时间。按照 keyword 对数据流中的数据进行统计,并将统计结果存放在构造的 KeysValuesMap 中。若 keyword 不在 KeysValuesMap 中,则将该 keyword 作为 KeysValuesMap 的 key,并把该网络攻击 IP 的最早攻击时间和最晚攻击时间定义为转换的 JAVA 时间;如果该 keyword 已经在 KeysValuesMap 中存在,则将转换的 JAVA 时间与 KeysValuesMap 中的最值进行比较,若小于最小值则替换最小值,若大于最大值则替换最大值,否则不操作。

Step4　经过数据统计分析之后,得到网络攻击 IP 在当前小时粒度下的最早开始攻击时间和最晚攻击时间,将得到的两个时间进行时间格式化,并把差值作为该网络攻击 IP 在当前小时粒度下的攻击时长。

Step5　系统组件:数据处理组件的运行时间达到预先设定的时长后,转入算法 4,即网络攻击 IP 各属性 TOP N 统计算法。

Step6　算法结束。

3.3.2　网络攻击 IP 各属性 TOP N 的统计

识别出网络攻击 IP 后,需要对攻击 IP 进行属性提取与分析,本文分别对网络攻击 IP 的端口、地区、配置以及域名进行 TOP N 求解。网络攻击 IP 各属性 TOP N 求值的 Bolt 对上个数据流中的数据按照 keyword 进行 Field grouping,并在内存中维护一个 TOP N 的列表。由于具有相同 keyword 的数据流会被发送到同一个节点进行处理,因此这种方式对于大的数据量显然没有可伸缩性。一种更好的方式是并行计算数据流中各属性的 TOP N,然后由一个 Merge 的 Bolt 将局部的 TOP N 合并在一起,计算一个全局的 TOP N,最后将计算结果发送给下一个 Bolt 进行处理或者数据持久化。网络攻击 IP 各属性 TOP N 的统计算法如算法 4 所示。

算法 4　网络攻击 IP 各属性 TOP N 的统计算法

输入:网络攻击 IP 识别 Bolt 数据流中的所有字段

输出:发射网络攻击 IP,网络 IP 攻击时间,IP 统计属性字段以及统计量的 Tuple

Step1　接入网络攻击 IP 识别 Bolt 处理后的数据流,取出本算法所需的 Field,进行网络攻击 IP 各属性 TOP N 的统计,并定义本算法的 keyword。

Step2　判断接入数据流中组件信息的类别,如果组件信息为系统组件,则清除系统内存中的数据,释放存储空间,跳到 Step4;如果组件信息为数据处理组件,则跳转至 Step3。

Step3　数据处理组件:对数据流中的数据按照不同的属性构建不同的 keyword 并统计对应属性的 TOP N,并将统计结果存储于构造的 KeysValuesMap 中,若该 keyword 已经是 KeysValuesMap 的 key,则 value 加 1;否则将该 keyword 作为

KeysValuesMap 的 key，并将 value 赋值为 1。

Step4 MergeBolt 接入网络攻击 IP 各属性 TOP N 统计的数据流，对各属性的 TOP N 统计结果进行排序，并且取出前 N 条数据作为数据统计结果。

Step5 对 IP 各属性 TOP N 统计结果进行数据持久化，算法结束。

4 分布式网络日志实时计算与分析平台的构建

为了对构建的攻击 IP 识别模型和 IP 真人属性判定模型进行有效性验证，本文构建了基于 Storm 流式计算框架的分布式网络日志计算与分析平台。该平台体系架构包括分布式数据采集、分布式数据缓存、分布式实时数据消费和分布式数据结果存储，系统架构如图 2 所示。

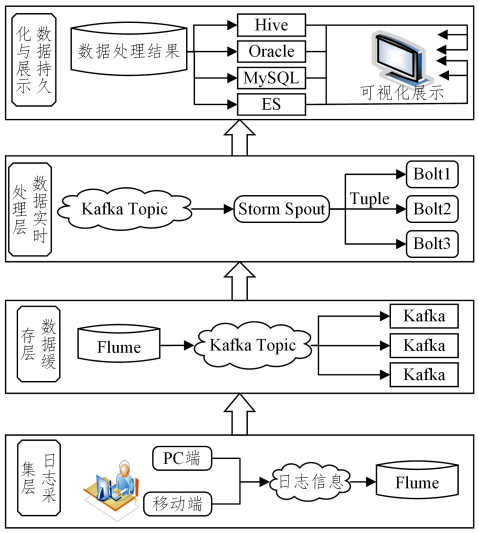


图 2 系统架构图

Fig. 2 System architecture diagram

4.1 网络日志分析与计算框架的选型

对网络日志进行分析处理主要有以下要求：1）实时性；2）高可靠性；3）资源低耗性；4）高准确性。依据上述 4 种要求的相关研究现状，本文采用 Storm 流式处理技术作为数据处理计算框架。与其他大数据处理技术相比，使用 Strom 处理网络日志具有以下优势：Storm 采用流式计算框架处理数据，拓扑结构一旦创建，便会持续处理数据流中新到达的数据，数据处理的实时性强；同时，Storm 对数据处理的关注点在于多次处理、一次写入。本文对 Spark，Hadoop，Storm 大数据处理技术之间的特性[14]进行了总结，如表 1 所列。

表 1 大数据处理技术的对比

Table 1 Comparison of big data processing technology

对比点	Spark	Storm	Hadoop
计算模式	分布式批处理计算	分布式实时计算	分布式批处理计算
计算位置	内存	内存	磁盘
实时性	低实时性	准实时性	无实时性
计算框架	迭代计算框架	流式计算框架	分布式计算框架
适用场景	(1)多次操作特定数据集的应用场合 (2)粗粒度更新状态的应用	(1)在线机器学习 (2)持续计算 (3)实时计算系统	(1)海量数据的离线分析处理 (2)大规模 Web 信息搜索 (3)数据密集型并行计算
吞吐量	高	低	较低

在对大数据处理技术进行综合分析的基础上，结合网络日志分析处理的需求，本文采用 Flume 技术完成网络日志的采集，并将数据采集结果存储于 Kafka 消息队列中；采用 Storm 流式计算框架完成网络日志的实时分析处理；采用 Elastic Search 作为实时搜索引擎，完成实时响应处理。

4.2 平台构建中遇到的技术问题及解决方案

在构建分布式网络日志实时计算与分析平台的过程中会遇到许多技术问题，本文在对相关技术问题进行深入研究后，给出了翔实的解决方案。

4.2.1 Kafka 数据丢失与重复消费

在项目的实际开发过程中，Spout 接入 Kafka 消息队列中的数据作为本系统中的数据源，但是在采用 Kafka 读取消息进行处理时，会出现 consumer 重复消费 Kafka 队列中的数据，导致计算结果明显大于准确数据指标的问题；除此之外，在 Storm 流式处理计算流程中，数据量骤降，大量的数据在计算过程中丢失。以上两个问题都导致数据计算结果异常，在定位问题之后，经过调研与研究，我们发现了导致以上问题的原因并提出了解决方案。

Kafka 的组件 consumer 消费数据时，首先会从 broker 里读取一批消息数据进行处理，处理完成后再提交 offset。而本文项目中的 consumer 的消费能力比较低，并且我们设置的消费拉取尺寸和消费时长不合理，导致取出的一批数据在 session. timeout. ms 时间内没有被处理完成，自动提交 offset 失败；然后，Kafka 会重新分配 partition 给其他 consumer，consumer 又重新消费之前的一批数据，又出现消费超时，因此会造成死循环，导致一直消费相同的数据。拉取出的数据未经消费会被系统抛弃，导致数据丢失，从而出现计算结果异常的问题[15]。

找出问题的原因之后，本文给出了关于 Kafka 数据丢失与数据重复消费的解决方案。在项目实施过程中，我们使用了 Spring-Kafka，因此需要把 Kafka 消费者的配置 enable. auto. commit 设为 false，以禁止 Kafka 自动提交 offset。当 consumer fetch 的数据被处理后，手动执行 consumer. commitSync()，在消费数据成功后再手动提交 offset；同时依据数据量的大小，灵活设置 consumer. poll() 和 session. timeout. ms() 的参数，保证拉取出的数据能够在 heartbeat. interval. ms 间隔内消费成功，避免经过 reblance 后再次出现数据重复消费的问题。采用这种策略能够完全解决数据丢失与数据重复消费的问题。

4.2.2 Storm 集群机器节点重启

Storm 作为大数据流式处理技术，具有大吞吐量的实时计算能力。数据到达系统时会立即在内存中进入数据处理流程，能够在很短的时间内处理完成。由于 Storm 是基于内存的计算模式，因此 Storm 对大数据分析结果的储存能力相对较差。我们在系统开发过程中通过 Storm UI 发现，topology 在运行一段时间之后，Storm 集群的机器节点会不定时重启，导致最终的数据计算结果偏移预期值。

经过调研与系统排查之后发现，Storm 集群机器节点重启是由于数据计算结果超过内存存储的上限而溢出，导致 Worker 挂掉。而 Storm 自身具有容错机制，当 Worker 失败

后其会尝试在本机重启它，如果它在启动时连续失败了一定的次数，无法发送心跳信息到 Nimbus，Nimbus 将在另一台主机上重新分配 Worker[16]。内存溢出触发了 Storm 的重启机制，导致最终的计算结果不准确。

采用 Oracle 数据库对中间数据分析结果进行缓存，从而解决 Storm 机器节点重启的问题。在很多场合下，Bolt 在处理数据落地时如果一条数据到来就进行数据存储或者读写是极耗性能的，特别是在数据量大的情况下，因此我们在对中间数据结果进行缓存时采用定时＋定量的批量写入（读入）策略。同时，构建数据定时清理算子，对处理完成的数据定时清除以释放存储空间。采用 Oracle 数据库对中间数据分析结果进行暂时缓存的方案，能够解决因内存溢出而导致机器节点重启的问题。

4.2.3　topology 运行时间超长及不运行

在发布 topology 到 Storm 集群上时，通过 Storm UI 会发现 Spout 以及 Bolt 的各个工作节点长时间都没有进行数据处理，导致该问题的主要原因是资源不足[17]。在项目开发工程中，需要根据实际数据量的大小设置合理的 Worker 数目[18]。本项目中的设置如下：

```
conf. setNumWorkers(8);
```

再依据 Spout/Bolt 处理数据的复杂性设置合理的并发度 Executor，本项目中的设置如下：

```
builder. setSpout (SpoutName, new KafkaSpout (),1)
builder. setBolt(CountSumBoltName, new CountSumBolt (),2);
builder. setBolt (TOPNBoltName, new TOPNBolt (),8)
builder. setBolt (MergeBoltName, new MergeBolt (),1)
```

Spout 接入 Kafka 数据时只进行数据预处理，不进行数据分析与计算，因此设置并发度为 1；MergeBolt 仅对数据统计结果进行汇总，不需要做复杂的运算，因此并行度也设置为 1，以节省 Storm 集群资源。然后再对网络攻击 IP 进行 TOP N 统计，数据处理量比较大，逻辑复杂，因此设置并发度为 8，以提高数据处理能力与效率。这样根据实际情况对 Spout 和 Bolt 设置合理的 Executor，既能够优化 topology，又能节约 Storm 集群资源。依据上述设置，topology 运行时间超长或者不运行的情况就可得到解决。

5　实验数据及结果分析

5.1　实验数据

本文实验数据来源于国内某科研机构提供的全量网络访问日志，日均日志量为 20 亿～30 亿条。通过 Flume 采集数据，数据采集结果存储于消息队列 Kafka 中，数据格式非常规范。由于数据涉及隐私，因此本文对实验数据以及实验结果进行了数据加密处理。本系统对每天的全量网络日志进行统计与分析，及时发现网络隐患并对其进行监控。表 2 列出了 Spout 接入 Kafka 消息队列中的部分用户网络日志的数据示例。

表 2　用户网络日志数据示例

Table 2　User network log data example

2018-05-07 08:01:10.362 b. s. d. executor [INFO] BOLT ack TASK:45 TIME:TUPLE:source:BISTU_AttackIpSpout:189, stream:default,id:{},[421277,5745594,1516600527,1516600521,15.239.211. *,111.223.10. *,80,53121,中国:浙江:杭州::电信,新加坡:新加坡,baidu. com,2018-05-07,13]
2018-05-07 08:01:10.362 b. s. d. executor [INFO] Execute done TUPLE source:BISTU_ AttackIpSpout:189,stream:default, id:{},[421277,5745594,1516600527,1516600521,115.239.211. *,111.223.104. *,80,5311,中国:浙江:杭州::电信,新加坡:新加坡,baidu. com,2018-05-07,13]
2018-05-07 08:01:10.362 b. s. d. executor [INFO] Execute done TUPLE source:BISTU_ AttackIpSpout:189,stream:default, id:{},[421277,5745594,1516600527,1516600521,115.239.211. *,111.223.104. *,80,5311,中国:浙江:杭州::电信,新加坡:新加坡,baidu. com,2018-05-07,13]
2018-05-07 08:01:10.362 b. s. d. executor [INFO] Execute done TUPLE source:BISTU_ AttackIpSpout:189,stream:default, id:{},[421277,5745598,1516600514,1516600409,123.125.115. *,49.127.21. *,80,50998 中国:北京::联通,澳大利亚:维多利,tieba. baidu. com,2018-05-07,13]

5.2　实验环境

本文实验环境如下：由某科研机构提供的 Hadoop 集群服务器 52 台，Kafka 集群服务器 2 台，Storm 集群服务器 2 台，Oracle 数据库服务器 4 台，Elastic Search 分布式数据库服务器 4 台，JDK1.7 开发编译环境，IntelliJ IDEA 开发工具，Hive 数据仓库（Hive-2.3.1 版本），CentOS 6.4 操作系统。

5.3　实验结果

本文采用真实数据集对构建的网络攻击 IP 识别模型的有效性进行验证，实验结果表明，2500 万条网络 IP 识别出的攻击 IP 数量为 341 条，攻击 IP 的标注准确率为 98%；同时，提取时间分区 partition(dd='2018-05-07')内的网络 IP 日志数据进行真人率识别判定。与传统的真人值 0-1 二值判定相比，本文构建的基于滑动时间窗口的 IP 真人属性判定模型对真人属性的描述更加准确，标注结果更具有可信度。图 3 为传统的 NTH 真人值判定与本文构建的真人率判定的结果对比图，横坐标为网络 IP 数量的占比，纵坐标为真人率。可以看到，真人值 0-1 判定的统计结果只集中于 0%和 100%的真人率区间，而本文构建的基于滑动时间窗口的 IP 真人属性判定模型计算出的结果按真人率的百分比区间分配，且标注准确率达到 96%。

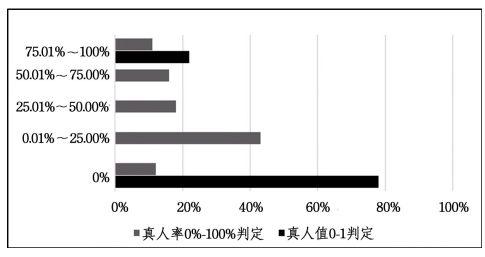


图 3　真人率判定结果对比图

Fig. 3　Contrast diagram of results of life-rate judgment

对识别出的网络攻击 IP 进行细粒度的属性提取分析，包

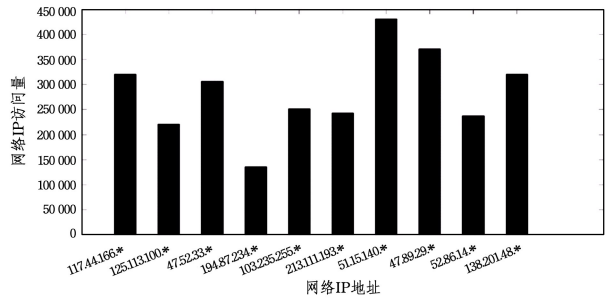
括网络攻击 IP 的攻击时长属性的提取，网络攻击 IP 的端口、地区、配置以及域名属性 TOP N 的计算。数据分析结果采用 Hive 进行持久化，按照时间建立分区索引。本文以 2018 年 5 月 7 日 8 点的数据为例进行分析展示，表 3 列出时间分区 partition(dd='2018-05-07',hh='08')内的 3 条网络攻击 IP 的数据统计信息。

表 3 网络攻击 IP 各属性的统计信息

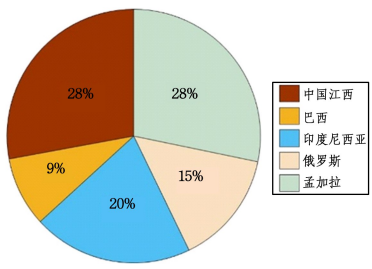
Table 3 Network attack IP attribute statistics

属性/网络 IP	117.44.166.*	125.113.100.*	47.52.33.*
访问量	320503	220774	306242
域名量	1	1	15
目标 IP 量	772	5750	1
目标端口量	50	92	5000
配置量	2	2	10
目标 IP 的 TOP5	93.183.144.*=152 200.49.43.*=150 177.130.132.*=144 190.11.32.*=144 213.6.199.*=144	191.8.80.*=217 177.21.126.*=212 207.38.90.*=199 191.242.173.*=195 89.38.97.*=189	223.13.33.*=66242
目标端口的 TOP5	8080=22447,20183=8108, 80=6750,53281=5868, 8888=1117	8080=103289,3128=55606, 20183=48156,1080=28052, 53281=22850	20403=19,20142=19, 21056=18,20490=18, 21256=18
目的地区的 TOP5	中国江西=50503 巴西巴西=11435 印度尼西亚=3293 俄罗斯=2350 孟加拉=1560	中国浙江=310774 巴西=69002 印度尼西亚=29988 美国=19686 俄罗斯=15021	中国香港=66242 中国山西=66242
配置的 TOP5	5668771=25282 5727861=25221	5668771=155429 5727861=155345	9392324=18543 9321375=14146 9419778=8485 9325917=5429 9421670=4398
域名的 TOP5	jjhgame.com=50503	jjhgame.com=310774	www.69759.com=18543 www.52636.com=14146 www.q85.com.cn=8485 www.90679.com.cn=5429 www.90236.com.cn=4398
开始攻击时间	2018-05-07 08:00:00	2018-05-07 08:00:00	2018-05-07 08:03:39
结束攻击时间	2018-05-07 08:59:59	2018-05-07 08:59:59	2018-05-07 08:59:59
攻击时长	3599	3599	3380

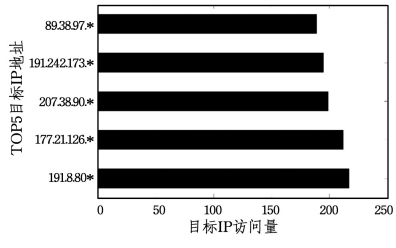
本文依据数据属性对实验结果进行可视化展示，通过对网络攻击 IP 识别模型检测出的攻击 IP 进行统计类属性提取与分析，能够实时监控网络攻击 IP 的访问量分布以及其他属性的分布趋势，便于更加直观、实时地监控国家网络安全。图 4(a)为 2018 年 5 月 7 日 8 点的网络攻击 IP TOP10 的访问量分布图，图 4(b)为网络攻击 IP 地址为 117.44.166.* 的目的地区 TOP5 分布图，图 4(c)为网络攻击 IP 地址为 125.113.100.* 的访问目标 IP 的 TOP5 分布图。



(a)访问量分布图



(b)目的地区 TOP5 分布图



(c)目标 IP 的 TOP5 分布图

图 4 网络攻击 IP 属性分布图

Fig. 4 Network attack IP attribute distribution map

结束语　本文采用 Flume，Kafka，Storm，Elastic Search 等框架技术，开发了基于 Storm 流式计算框架的具有实时收集、分析、存储等功能的用户网络日志管理分析平台。本文提出了基于访问行为与网络关系的攻击 IP 识别模型，提高了攻击 IP 标注的准确性与完善性，同时提出了基于滑动时间窗口的 IP 真人属性判定模型。与传统的真人值 0-1 二值判定模型相比，本文构建的模型对于网络 IP 真人属性的描述更加准确和细致。采用某科研机构提供的用户访问日志进行数据分析的实验结果表明，所构建的分布式网络日志计算与分析平台既能够处理海量的网络日志数据，也能保证数据处理的实时性，发挥实时分析与监控网络安全的作用。

在下一步的工作中，需要对网络日志进行细粒度的分析提取，同时对网络 IP 的安全性进行预测；进一步优化 Storm 处理网络日志的空间复杂度，优化数据处理算子，以提高网络日志的处理能力。

参 考 文 献

[1] WANG X L，CHEN M，XING C Y，et al. A software-defined security network mechanism for defending against DDoS attacks [J]. Journal of Software，2016，27(12)：3104-3119. (in Chinese)
王秀磊，陈鸣，邢长友，等. 一种防御 DDoS 攻击的软件定义安全网络机制[J]. 软件学报，2016，27(12)：3104-3119.

[2] CAO X，FEI J L，ZHU Y F. Anti-identification model of operating system based on network spoofing[J]. Computer Application，2016，36(3)：661-664. (in Chinese)
曹旭，费金龙，祝跃飞. 基于网络欺骗的操作系统抗识别模型[J]. 计算机应用，2016，36(3)：661-664.

[3] MAZZARIELLO C，BIFULCO R，CANONICO R. Integrating a network IDS into an open source Cloud Computing environment [C]// 2010 Sixth International Conference on Information Assurance and Security (IAS). IEEE，2010.

[4] WU J. Research on Network User Access Pattern Mining Algorithm[J]. Computer Engineering and Applications，2016，52(5)：61-64. (in Chinese)
武健. 网络用户访问模式挖掘算法研究[J]. 计算机工程与应用，2016，52(5)：61-64.

[5] ZHENG B H，JUN S，SHIROCHIN V P. An Intelligent Lightweight Intrusion Detection System with Forensics Technique [C]// IEEE Workshop on Intelligent Data Acquisition & Advanced Computing Systems：Technology & Applications. IEEE Xplore，2007.

[6] XU K Y，GONG X R，CHENG M C. Audit Log Association Rule Mining Based on Improved Apriori Algorithm[J]. Computer Application，2016，36(7)：1847-1851. (in Chinese)
徐开勇，龚雪容，成茂才. 基于改进 Apriori 算法的审计日志关联规则挖掘[J]. 计算机应用，2016，36(7)：1847-1851.

[7] HAN H，LU X L，REN L Y. Using data mining to discover signatures in network-based intrusion detection[C]// 2002 International Conference on Machine Learning and Cybernetics. IEEE，2002.

[8] SUN X B，SHI F D. Research and Optimization of Apriori Algorithm Based on Hadoop[J]. Computer Engineering and Design，2018，39(1)：126-133. (in Chinese)
孙学波，石飞达. 基于 Hadoop 的 Apriori 算法研究与优化[J]. 计算机工程与设计，2018，39(1)：126-133.

[9] HU Y P，DING W L，WANG G L. A monitoring and scheduling service for heterogeneous big data computing framework[J]. Computer Science，2018，45(6)：73-77，101. (in Chinese)
胡雅鹏，丁维龙，王桂玲. 一种面向异构大数据计算框架的监控及调度服务[J]. 计算机科学，2018，45(6)：73-77，101.

[10] WANG G，KOSHY J，SUBRAMANIAN S，et al. Building a replicated logging system with Apache Kafka[J]. Proceedings of the Vldb Endowment，2015，8(12)：1654-1655.

[11] CHEN Y，ZHU N，SHI Y. Online analytic processing of big data based on Hive[J]. Computer Era，2018(1)：1-3.

[12] ZHENG K，WANG X. Feature selection method with joint maximal information entropy between features and class[J]. Pattern Recognition，2018，77：20-29.

[13] GAO J X. NAT recognition method based on Network Traffic Features[D]. Chengdu：University of Electronic Science and Technology of China，2012. (in Chinese)
高骥翔. 基于网络流量特征的 NAT 识别方法[D]. 成都：电子科技大学，2012.

[14] WANG C K，MENG X F. Research on Distributed Data Flow Relational Query Technology[J]. Journal of Computer，2016，39(1)：80-96. (in Chinese)
王春凯，孟小峰. 分布式数据流关系查询技术研究[J]. 计算机学报，2016，39(1)：80-96.

[15] WANG Y，WANG C. A reliable Consumer Design Scheme based on Kafka[J]. Software，2016，37(1)：61-66. (in Chinese)
王岩，王纯. 一种基于 Kafka 的可靠的 Consumer 的设计方案[J]. 软件，2016，37(1)：61-66.

[16] CARDELLINI V，GRASSI V，PRESTI F L，et al. Distributed QoS-aware scheduling in storm[C]// ACM International Conference on Distributed Event-Based Systems. ACM，2015：344-347.

[17] GHADERI J，SHAKKOTTAI S，SRIKANT R. Scheduling Storms and Streams in the Cloud[C]// ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems. ACM，2015：439-440.

[18] JIANG Y，LUO Y H，ZHU H W. Topology-based task scheduling strategy under Storm cluster[J]. Computer Engineering and Applications，2018，54(7)：84-88，95. (in Chinese)
蒋溢，罗宇豪，朱恒伟. Storm 集群下一种基于 Topology 的任务调度策略[J]. 计算机工程与应用，2018，54(7)：84-88，95.