A Real-time Network Traffic Anomaly Detection System based on Storm

Gang He,Cheng Tan,Dechen Yu,Xiaochun Wu
Beijing Key Laboratory of Network System Architecture and Convergence
Beijing University of Posts and Telecommunications
Beijing, China
tancheng_bupt@163.com, brainhe@bupt.edu.cn

***Abstract*—In recent years,with more and more people shopping, chatting and video online,the Internet is playing a more and more important role in human's daily life.Since the Internet is so close to our lives,it contains so much personal information that will cause a lot of troubles or even losses when divulged.So it's necessary and urgent to find a efficient way to detect the abnormal network behavior. In this paper,we present a new detection method based on compound session.In contrast to previous methods,our approach is based on the cloud computing platform and the cluster system,using Hadoop Distributed File System (HDFS) to analysis and using Twitter Storm to make real-time network anomaly detection come true.**

Keywords-Compound Session;enterprise user's behavior;host analytics;Hadoop platform;Twitter Storm

I. INTRODUCTION

The last decade has witnessed the explosion of data,and the network traffic flow is getting increasingly complex due to various reasons[1-3],including cyber attacks,network problems,etc. Abnormal flows or malicious cyber attacks were once limited to causing inconvenience for a small group of infected Computer users.Unlike past,Today's anomalous traffic often goes beyond affecting a single user and can create failures in an entire system,causing many losses[4-7], including productivity loss,economic loss,theft, compromise of critical infrastructure and even loss of public well being.

Different approaches have been proposed for detection of abnormal network behavior.Some methods may base on signature of data,and some other methods may base on the source port where the data is coming from.All of the methods has its own uniqueness as well as shortcomings.With the explosion of data in recent years,the traditional ways to detect anomaly network traffic are no longer suitable. Hadoop Distributed File System (HDFS) may be a good solution for dealing with the huge amount of data,but due to lacking of the instantaneity,Hadoop is not so suit for real-time anomaly detection.Luckily,Twitter Storm[8] come up.Apache Storm is a distributed framework that offers streaming integration with time-based in-memory analytics from the live machine data as they come in stream. It generates low latency, real-time results.

In this paper,we provide a scheme that makes it possible to apply Twitter Storm on large data real-time processing and detecting anomaly network traffic.And the work makes the following contributions:

- We collected the data from more than 2000 enterprises in the real network environment,acquired more than 50GB data every day, which ensures the authenticity and the magnanimity of the data.
- We provide a new method named Compound Session to analyze the data set. Based on the Compound Session, we produce three types of tables, including host real-time table and other two tables,through which we can discover periodical changes of network traffic, and even detect the anomaly network traffic.
- The data processing is based on both Hadoop and Storm,using Hadoop to analysis data and get traffic law,and using Twitter Storm to detect anomaly network traffic in real-time.

The remainder of the paper is structured as follows. Section II shows The data set,including data collection, data description.The framework of data processing and the detailed detecting anomaly network traffic is presented in Section IV. Finally, conclusions are summarized in Section V.

II. DATASET

A. Data Capturing

The network traffic was collected by commercial Traffic Monitoring System (TMS) developed by our research team.TMS is deployed in the network gates of the enterprises, covering more than 2000 enterprises in some province in Southern China. The deployment of TMS in the network is shown in Figure 1.

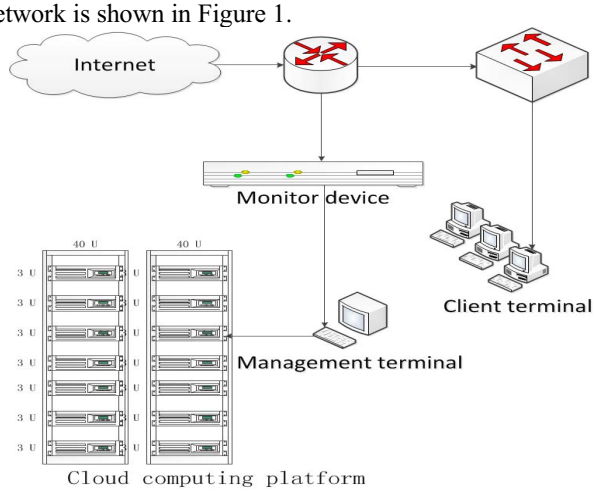

Figure 1. Deployment of the Traffic Monitoring System

978-1-4799-8646-0/15 $31.00 © 2015 IEEE
DOI 10.1109/IHMSC.2015.152

B. Data Description

In this paper,we propose a new concept called Compound Session to explain the process of data collection. We define a four-tuple as a mark of the host. The four-tuple contains a source IP address, a destination IP address, a protocol and a destination port.

According to the characteristic of network traffic,in a compound session,We define a busy period counter and an idle period counter.As the name suggests,when the probe capture a packet,the session goes in busy period,and the busy-period counter increases by one.when the probe capture nothing,the session goes in idle period and the idle-period counter adds one as well.Usually we set 10 minutes to generate a record file which contains 31 statistical parameters.These 31 statistical parameters were measured from three aspects including Macrocosm level, Intermediate level and Microcosm level.Table I shows these statistical parameters in detail.

TABLE I. STATISTICAL PARAMETERS

CLASSIFICATION	NAME	DESCRIPTION
Macrocosm level	TUP/TDP	the number of uplink/downlink packets
	TUB/TDB	the number uplink/downlink bytes
	TSE/TSR/ TAS	the total number of new/end/active connections
Intermediate level	BT/IDT	the number of busy/idle periods
	BE/IDE	average duration of busy and idle period
	V_BD/V_ ID	standard deviation of duration of busy and idle period
Microcosm level	E_TS	average exchange times between uplink traffic and downlink traffic in busy period
	E_DT/E_ UT	average number of of uplink and downlink speaking packets in busy period
	E_DTP/E _UTP	averge number of of uplink and downlink speaking packets
	E_DTB/E _UTB	averge number of of uplink and downlink speaking bytes
	V_DTP/V _UTP	standard deviation coefficient of uplink and downlink speaking packets
	V_DTB/V _UTB	standard deviation coefficient of uplink and downlink speaking bytes

We also set that every 3 hours as the reporting time, all the files would be submitted to the central server, stored as a new file.The total procedure and detailed steps are described in the Figure 2.Figure 3 shows the definition Schematic diagram of Composite Session.

It is worth mentioning that,when we consider a extreme case such as a session lasts forever, theoretically this record file will be too large for the probe to bear.To fix problems like that,we set a threshold of 65535 for both the busy period counter and the idle period counter,that either of them reaches the threshold,the statistics about the busy period and idle period will be reset.Moreover,when capture nothing for more than 2 hours,the probe will go into hibernation until the session is captured again.

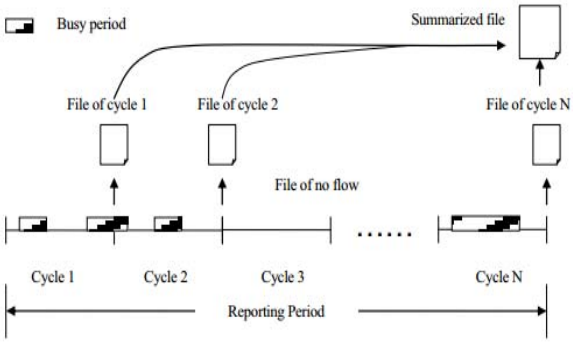

Figure 2. Report of the Statistical Parameters

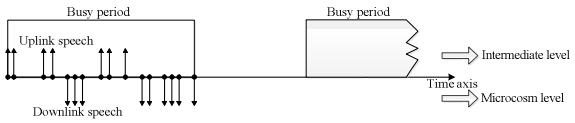

Figure 3. Schematic Diagram of Composite Session

C. Data Processing

Storm is a distributed, reliable, fault tolerant data stream processing system. It provides a series of basic element to compute: Topology, Stream, Spout, Bolt and so on. Data Stream is the abstraction of the data in the Storm, it is unbounded tuple sequence on time. In a Topology, Spout is the source of Stream, responsible for the Topology lunch Stream from a specific data source. Bolt can receive any number of Stream as input, and then data processing process, if need be, Bolt can also launch a new Stream to the lower Bolt for processing.

Twitter Storm is a real-time processing system, which provides a computational model for real-time processing type applications, can be easily programmed processing.

Usually,we would like several real-time data sources unite as input,and then transfer data through the analysis system to processing.Because the system processing speed and the data source producing data speed may not be consistent,We can consider the data source is directly connected to a message oriented middleware,such as Kafka[9], to perform scheduling.Data source, as the Producer of news,submit what it produces(log data,business request data,etc.) to Kafka, and then through the way of subscription,using the Storm Topology as message Consumer,analysis data in real-time in the Storm cluster[10].

Storm module is composed of the following parts:

- Data Preprocessing Modular

Data Preprocessing Modular is responsible for doing Principal Component Analysis on composite session.It extract keywords such as CycleEndTime, IP, UpBytes, DownByetes, etc. for Anomaly Detecting Modular.

- Anomaly Detecting Modular

Anomaly Detecting Modular contains two bolts, Count Bolt and Monitor Bolt.Figure 4 shows its structure.

Count Bolt is the first bolt in this modular.It's main job is to sum UpBytes and DownBytes.

Monitor Bolt is the second bolt and is the key bolt in this system. It uses the KNN-Grubbs algorithm to detect anomaly traffic flow.KNN-Grubbs will be introduced in next section.After detected,both normal and anomaly data will be stored in HBase. Normal ones will update the value of KNN while anomaly ones will not accepted in KNN. Table II shows its stored format in HBase.

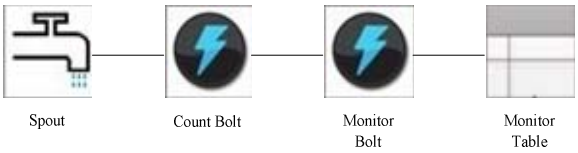

Figure 4. Anomaly Detecting Modular

TABLE II. MONITOR TABLE IN HBASE

Row key	Column: Status
time unit(e.g.1409500800)	0: normal
	1:anomaly

III. ALGORITHM

In this paper,we propose a new algorithm called 'Grubbs-KNN'. It's impossible to put every previous data into calculate model.Actually when we consider whether a traffic flow is anomaly or not, we only take a few flow before this flow into consideration.So here we choose k-NearestNeighbor(KNN) algorithm to select the appropriate data to predict.

Grubbs algorithm is described as follows:

Set n measurements in order of size $x_1 \le x_2 \le \cdots \le x_n$ .Suppose x_n is the data needed to be examined whether abnormal or not. x_n will be consider as anomaly if its residuals satisfy the following formula:

$$|V_n| = |x_n - \bar{x}| > g(n,\alpha) * \sigma(x) \quad (1)$$

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \quad (2)$$

$$\sigma(x) = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}} \quad (3)$$

In formula, $\bar{x}$ is the average value of array x , $\sigma(x)$ is the standard deviation of array x , $g(n,\alpha)$ depends on the number of measurements n and the significance level α (equivalent to committing "abandoning true" probability of error coefficient), α usually take 0.05 or 0.01,which means confidence probability P is 95% and 99%. Table III shows the how the value of $g(n,\alpha)$ changes.Because the anomaly traffic flow is usually much higher than the previous normal ones,the Grubbs algorithm fits the anomaly traffic detection prefectly.Since we can't choose too many nearest neighbor data, and also Grubbs algorithm needs the number of array be no less than 10, here we only choose a few number of k to test the accuracy.

TABLE III. GRUBBS CRITICAL VALUE $g(n,\alpha)$

n / p	0.95	0.99
10	2.176	2.410
11	2.234	2.485
12	2.285	2.550
13	2.331	2.607
14	2.371	2.659
15	2.409	2.705
20	2.557	2.884
25	2.663	3.009

IV. RESULTS AND ANALYSIS

In this section, we show some results to check out our system's accuracy. Experimental data is one day exported flow data form some enterprise in G province, China.TMS submit every ten minutes, so there are 144 data points.To show the result more directly,we randomly choose one statistical parameters each measuring level,which is total bytes for Macrocosm level,standard deviation of duration of busy period(V_BD) for Intermediate level and standard deviation coefficient of downlink speaking bytes(V_DTB) for Microcosm level.

Before the experiment, we should determinate the value of k. A few experiments were did to choose the value of k. The results are showed in Table IV. From the table,we can figure out that when take the value of 0.05,in other way the confidence probability P is 95%,when k beyond 15,there are little improvement in accuracy. Consider the capability of the system,we choose 15 as k.

TABLE IV. THE ACCURACY RATE OF DIFFERENT VALUES OF K

n / p	0.95
10	97.1%
12	98.4%
15	99.3%
20	99.8%

Figure 5 shows the appearance of system in Macrocosm level. From it we can see that, in the absence of anomaly traffic situations, variation curves of network traffic in a continuous time is stable.The front part of the curve just approves that the total traffic is stable. While if the curve appears some obvious changes,it may mean that traffic flow is anomaly.We can see that the traffic jumps from a very low

numerical to a extortionate one and reaches the first crest.Then it comes down.Soon after, it comes to another crest and this one lasts much longer.

The red curve in Figure 5 means the processed results.0 means normal and 1 means abnormal. We can see that the red curve just fits the anomaly traffic curve, which proves that our anomaly detection system can reflect the network traffic anomaly changes in real-time.

Figure 6 shows the results in Intermediate level and Figure 7 shows results in Microcosm level.Compared these three figures,we can see that though total bytes,V_BD and V_DTB are totally different,the system detected result are broadly similar in variation law, but not same.Figure 5 shows the anomaly traffic begins at about 65 in X-axis while Figure 6 begins at about 50 and Figure 7 begins at 55.But they do show that after a few unstable anomaly traffic, there is a anomaly traffic that lasts very long almost until the end of the day. Generally speaking, the system has a good test score, and also we can see that detecting anomaly traffic may not focus on only one statistical parameters,we should detect some more statistical parameters at the same time , make comprehensive consideration to acquire the best result.

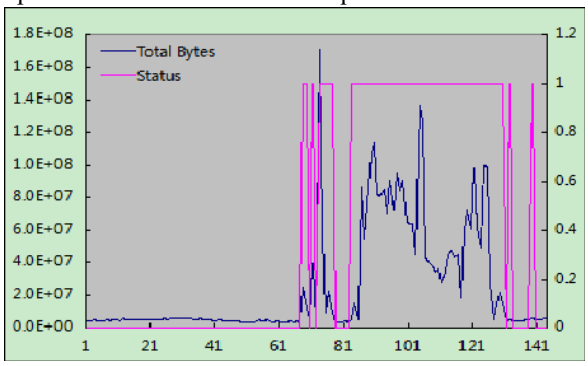

Figure 5. Total Bytes and Processed Results

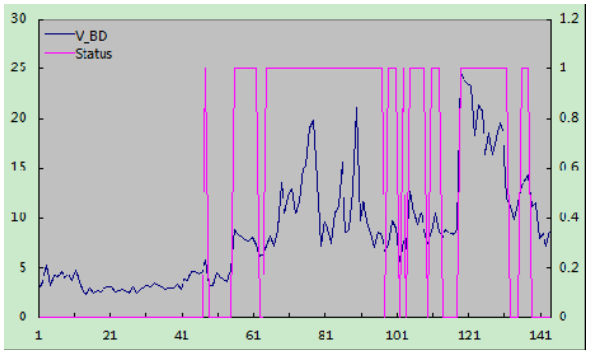

Figure 6. V_BD and Processed Results

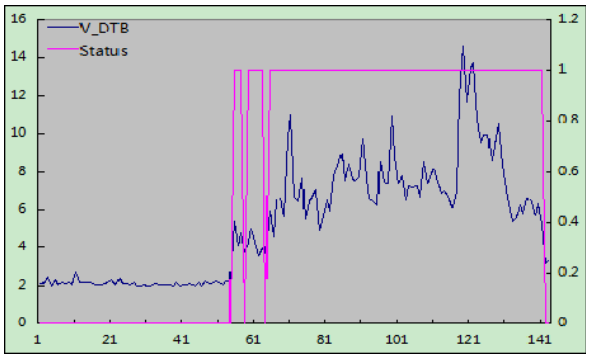

Figure 7. V_DTB and Processed Results

V. CONCLUSION

Due to the variability of anomalies, it's never a easy work to detect anomaly network traffic especially when needed to be real-time.This paper proposes a real-time anomaly detection system based on storm, including all the necessary modules and all the theories used in this system.We design a new algorithm called Grubbs-KNN which is perfectly fitted for real-time data stream detect. Experiment results turn out that this system can detect real-time anomaly traffic flow in a high accuracy. For future work, we will focus on using more parameters to further improve the system's ability to detect anomaly traffic flow and scalability.

REFERENCES

[1] [1]Stevan Novakov and Chung-Horng Lung."Studies in Applying PCA and Wavelet Algorithms for Network Traffic Anomaly Detection". IEEE 14th International Conference on High Performance Switching and Routing,2013.

[2] F.Dressler and G. Munz."Flexible flow aggregation for adaptive network monitoring". IEEE LCN Workshop on Network Measurements,2006.

[3] J.M. and Estevez-Tapiador. "Anomaly detection methods in wired networks: a survey and taxonomy," Computer Commun., 27(16), 2004.

[4] HuangCT and TharejaS. "Wavelet-based real time detection of network traffic anomalies".Securecomm and Workshops, 2006. IEEE, 2006: 1-7.

[5] Gerhard Munz and GeorgCarle. "Real-time Analysis of Flow Data for Network Attack Detection." Integrated Network Management, 2007.IM '07. 10th IFIP/IEEE International Symposium on, 21-25 May 2007, Munich, Germany:100-108.

[6] Xiaofen Wang and Liaojun Pang."A Scheme For Fast Network Traffic Anomaly Detection". International Conference on Computer Application and System Modeling,2010.

[7] Mohiuddin Solaimani and Latifur Khan,"Real-time Anomaly Detection Over VMware Performance Data Using Storm".IEEE IRI Conference Day 1: AUGUST 13, 2014.

[8] Twitter Storm, http://storm.incubator.apache.org/

[9] Apache Kafka Homepage, http://kafka.apache.org/.

[10] Mohiuddin Solaimani and Mohammed Iftekhar."Spark-based Anomaly Detection Over Multi-source VMware Performance Data In Real-time". Computational Intelligence in Cyber Security (CICS), 2014 IEEE Symposium.