Hadoop Environment for the Analysis of Large Network Packets

Keisuke Kato
The University of Aizu
Software Engineering Laboratory
Ikki-machi, Aizu-Wakamatsu, Fukushima
965-8580
m5191107@u-aizu.ac.jp

Vitaly Klyuev
The University of Aizu
Software Engineering Laboratory
Ikki-machi, Aizu-Wakamatsu, Fukushima
965-8580
vkluev@u-aizu.ac.jp

ABSTRACT

Nowadays, companies, governments, etc need to defend private information and data such as company's and government institution's data or latest research achievements. However, the crackers are maliciously attacking to steal the important data from them and sell the information such password, credit card number, etc. As the results of these attacks, many companies are getting seriously damages and they are loosing the trust from the users, customers, citizens, etc. Therefore, researches in computer security are required to develop intelligent defense systems. In network security, analyzing huge size of network packet capture (pcap) files is very important to monitor the behavior of networks and/or develop an intrusion detection system (IDS), intrusion prevention system (IPS), web application firewall (WAF), etc. In this paper, we developed an environment by using Hadoop. We executed SQLs to analyze about 85GB of network packet dataset provided by the University of New Brunswick. We presented how to analyze the huge size of pcap files on Hadoop and visualize the analysis results on the web browser by using Hadoop User Experience (Hue). Our result can be helpful for many people.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Computer Networks; D.2.8 [**Software Engineering**]: Metrics—*big-data analysis, distributed systems*

General Terms

Computer Networks, Analysis of Network Packet, Network Security

Keywords

Computer Networks, Packet Data Analysis, Distributed Systems, Network Security

ICAIT '16, Oct. 6 – 8, 2016, Aizu-Wakamatsu, Japan.

1. INTRODUCTION

According to report [5] by International Telecommunication Union (ITU), seven billion people live in an area that is covered by a mobile-cellular network. It means that the Internet is one of most necessary things in human beings' lives. There are many services on the Web such as social network services. As the results, the companies, governments, etc hold many types of private information on servers connected to the Internet. This information include: password, credit card numbers, individual numbers, etc. They should defend the information from crackers who are maliciously attack to steal it. However, there are many news reporting stolen such information. Therefore, analyzing the network packets is one of most important tasks to defend against such cyber attacks.

Our previous works [11, 12] studied the development of an intelligent detection systems against distributed denial of service (DDoS) attacks by analyzing large network packet data that is about 44GB. The results showed that the detection accuracy is high and the results of network packet analysis are very useful to develop detection systems against network attacks such as DDoS attacks.

We analyzed the provided packet capture (pcap) files by using Wireshark [14]. We extracted some features including source IP address, destination IP address, time interval in seconds between packets, and packet size in bytes from the dataset. After that, we analyzed the extracted data. However, it took very long time to finish this process. Therefore, the techniques that can analyze the pcap file directly and visualize the results are required. In this paper, we stored all pcap files to Hive table [2], analyzed the data by executing Hive SQLs, and visualized the results by using Hadoop [1] and Hadoop User Experience (Hue) [4].

The remainder of this paper is organized as follows. In Section 2, we shortly review publications in the area of this research. In Section 3, we explain our experimental environment such as Hadoop, Hive, etc. In Section 4, we present how we can analyze pcap data in the environment and how to visualize the results. In Section 5, we highlight our significant findings.

2. RELATED WORK

Wireshark [14] is open source software and one of most famous analyzing tools for network packets, but there are some limitations. Mistry et al [13] presented four different network monitoring tools that can monitor and analyze the network traffic. They discussed the disadvantage of Wireshark. According to them, it will not detect malicious activities on the network and it means that Wireshark may not be useful to study network security. In addition, Wireshark cannot handle the large packet data. According to Wireshark wiki [8], if the user has a large capture file more than 100MB, Wireshark will become slow while loading, filtering and alike actions.

Asrodia and Patel [9] studied the basics of packet sniffer that is used for network traffic analysis. To carry out this analysis, they introduced several tools. There are many tools to capture network traffic and analyze the data. However, there are some limitations for each tool. Some tools may only capture network traffic without analysis or require large memory.

Bachupally et al [10] presented an approach to analyze the network packet data and detect anomalous connections to the network by using the Hadoop Distributed File System (HDFS). They extracted some features of pcap data that size is 131MB by using Wireshark, exported them to the csv file, and uploaded to the HDFS environment. However, they handled small data and also used Wireshark to extract data into the csv file. It means that their approach requires long time for getting results of the analysis.

From these research reviews, we detected that one of the most difficult problem is to handle the large packet data and analyze it. Thus, in this study, we designed the analysis environment to handle the large packet data by using Hadoop, Hive, and Tez.

3. EXPERIMENTAL ENVIRONMENT

3.1 Apache Hadoop

Apache Hadoop is the open source software for reliable, scalable, distributed computing. The Apache Hadoop software library is a framework that allows the distributed processing of large datasets across a cluster of computers using simple programming models. It is designed to scale up from a single server to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures [1]. On Hadoop, the main execution engine is MapReduce. It s the core of Hadoop and it allows massive scalability for a huge Hadoop cluster.

3.2 Apache Hive

Apache Hive is the open source data warehouse software. It facilitates reading, writing, and managing large datasets residing on distributed storage using SQL. A command line tool and JDBC driver are provided to connect users to Hive [2]. Hive supports analysis of large datasets stored in HDFS.

In this study, we stored the provided dataset to the Hive table. However, there is no default option to handle the pcap files in Hive. So, we utilized the Hadoop PCAP library developed by Réseaux IP Européens Network Coordination Centre (RIPE-NCC) [6].

3.3 Hive on Apache Tez

Apache Tez [3] is a new application framework that can execute complex directed acyclic graphs of general data processing tasks and it can be a flexible and powerful successor of the MapReduce framework. MapReduce was initially created for processing and generating large data sets with parallel distributed algorithms on a cluster.

3.4 Hadoop User Experience (Hue)

Hadoop User Experience (Hue) is the open source software and web interface for analyzing data with Apache Hadoop [4]. Hue has editors for Hive, HBase, Spark, etc. In this study, we utilized Hue to execute hive SQL and visualize the results on the web browser.

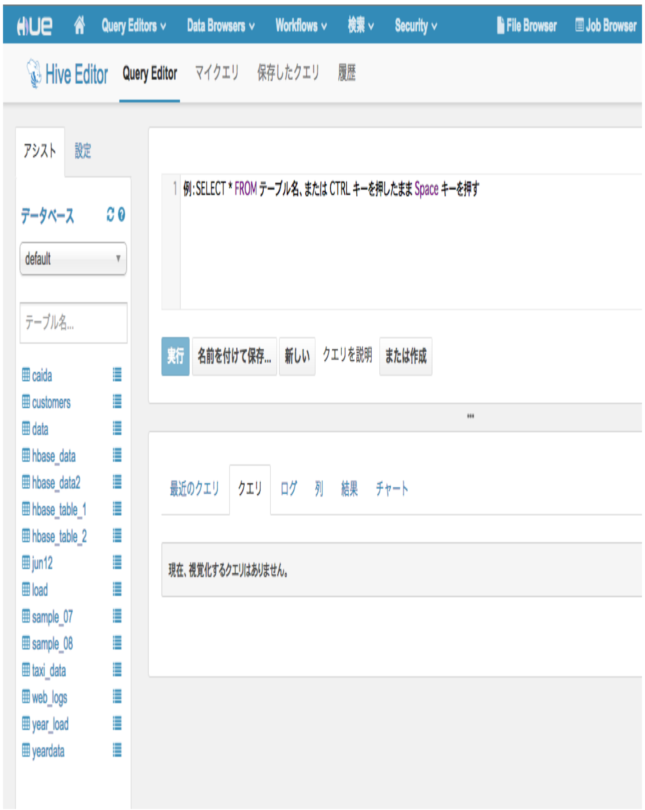
3.5 Dataset

The data that we utilized in this study were provided by Information Security Centre of Excellence at the University of New Brunswick [7]. This dataset consists of labeled network traces, including full packet payloads in the pcap format for seven days of network activity. The size is about 85GB.

4. TOOLS TO ANALYZE AND VISUALIZE THE DATA

In this study, we set up the standalone mode in Hadoop and executed SQLs on a local machine.

Figure 1 shows the Hive page on the Hue. On the left side of the page, we can select the database and show the tables in the database. In this case, we selected the database named *"default"* and showed the all tables. In the middle of the page, we can write and execute on an Hive SQL and the SQL results are displayed. On this page, we can see the submitted query and the result.

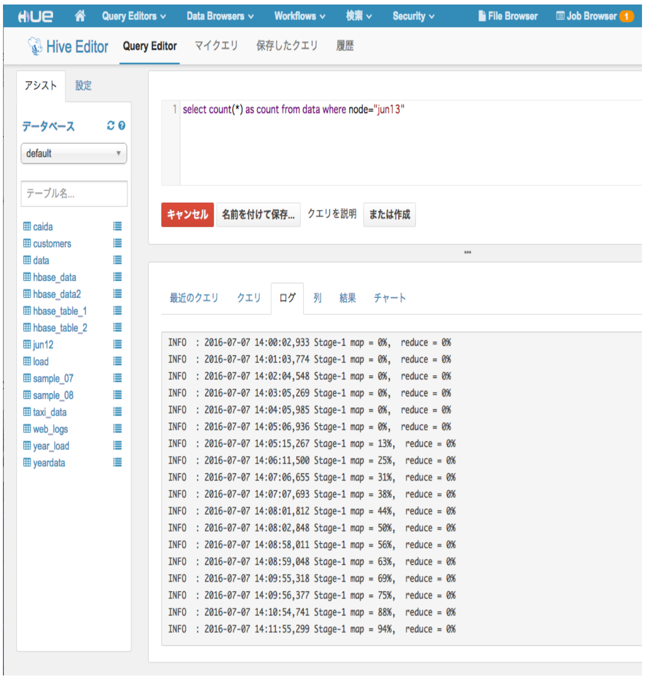


データベース: default

```
例:SELECT * FROM テーブル名, または CTRL キーを押したまま Space キーを押す
```

現在、視覚化するクエリはありません。

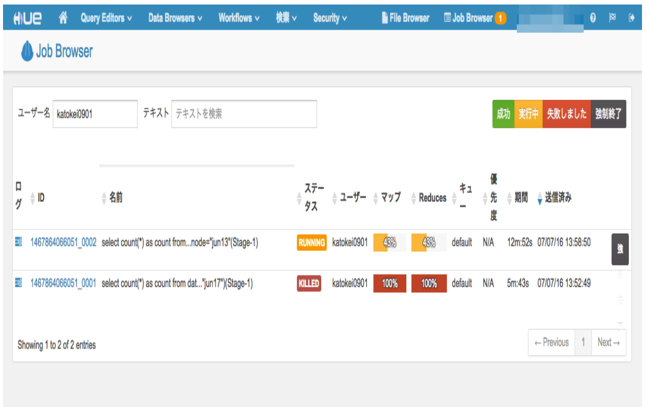
Figure 1: Hive page on Hue



```
select count(*) as count from data where node="jun13"
```

```
INFO  : 2016-07-07 14:00:02,933 Stage-1 map = 0%,  reduce = 0%
INFO  : 2016-07-07 14:01:03,774 Stage-1 map = 0%,  reduce = 0%
INFO  : 2016-07-07 14:02:04,548 Stage-1 map = 0%,  reduce = 0%
INFO  : 2016-07-07 14:03:05,269 Stage-1 map = 0%,  reduce = 0%
INFO  : 2016-07-07 14:04:05,985 Stage-1 map = 0%,  reduce = 0%
INFO  : 2016-07-07 14:05:06,936 Stage-1 map = 0%,  reduce = 0%
INFO  : 2016-07-07 14:05:15,267 Stage-1 map = 13%,  reduce = 0%
INFO  : 2016-07-07 14:06:11,500 Stage-1 map = 25%,  reduce = 0%
INFO  : 2016-07-07 14:07:06,655 Stage-1 map = 31%,  reduce = 0%
INFO  : 2016-07-07 14:07:07,693 Stage-1 map = 38%,  reduce = 0%
INFO  : 2016-07-07 14:08:01,812 Stage-1 map = 44%,  reduce = 0%
INFO  : 2016-07-07 14:08:02,848 Stage-1 map = 50%,  reduce = 0%
INFO  : 2016-07-07 14:08:58,011 Stage-1 map = 56%,  reduce = 0%
INFO  : 2016-07-07 14:08:59,048 Stage-1 map = 63%,  reduce = 0%
INFO  : 2016-07-07 14:09:55,318 Stage-1 map = 69%,  reduce = 0%
INFO  : 2016-07-07 14:09:56,377 Stage-1 map = 75%,  reduce = 0%
INFO  : 2016-07-07 14:10:54,741 Stage-1 map = 88%,  reduce = 0%
INFO  : 2016-07-07 14:11:55,299 Stage-1 map = 94%,  reduce = 0%
```

Figure 2: Execution of Hive SQL on MapReduce



ユーザー名 katokei0901 テキスト テキストを検索

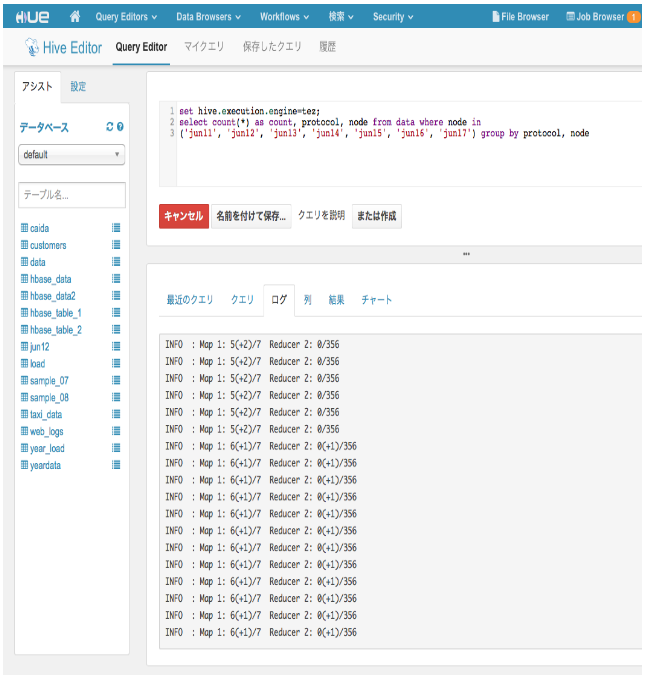
ログ	ID	名前	ステータス	ユーザー	マップ	Reduces	キュー	優先度	期間	送信済み	
☰	1467864066051_0002	select count(*) as count from...node="jun13"(Stage-1)	RUNNING	katokei0901	43%	43%	default	N/A	12m:52s	07/07/16 13:58:50	強
☰	1467864066051_0001	select count(*) as count from dat..."jun17")(Stage-1)	KILLED	katokei0901	100%	100%	default	N/A	5m:43s	07/07/16 13:52:49	

Showing 1 to 2 of 2 entries

Figure 3: Job Browser on Hue

Figure 2 shows an example of the Hive SQL execution to count the number of records in *data* table that is partitioned by *"node"* and node name is *jun13*. In the middle of Figure 2, *INFO: 2016-07-07 14:00:02,933* shows that we executed the SQL at 14:00:02,993 on July 07, 2016 and *Stage-1 map = *%, reduce = *%* shows how percentage of map and reduce job were finished. Figure 3 shows more details on the status of the MapReduce job. On this page, we can see the log by clicking the square, execution ID, the executed SQL, the status of the job, the user name, a percentage of the map and reduce job, a queue, priority (if it sets), the elapsed time, and time when the SQL was submitted from left side. The status is categorized by four colors: Green shows successfully executed, Yellow shows running the SQL, Red shows the failed execution, and Black shows the forced termination.

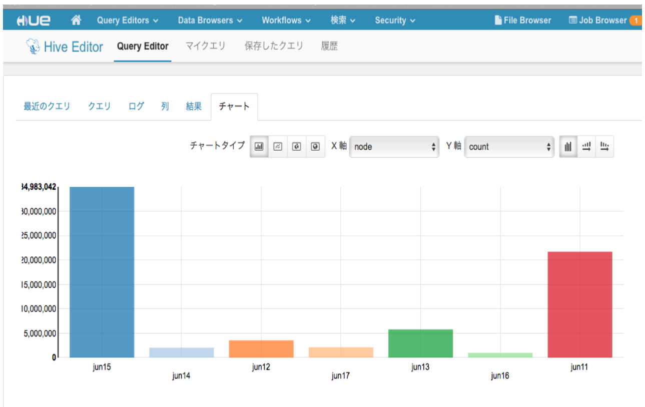
Figure 4 shows an example of executing Hive SQL on Tez. The first line in the SQL sets the execution engine in Hive from MapReduce to Tez. This SQL counts the number of



```
set hive.execution.engine=tez;
select count(*) as count, protocol, node from data where node in
('jun11', 'jun12', 'jun13', 'jun14', 'jun15', 'jun16', 'jun17') group by protocol, node
```

```
INFO  : Map 1: 5(+2)/7  Reducer 2: 0/356
INFO  : Map 1: 5(+2)/7  Reducer 2: 0/356
INFO  : Map 1: 5(+2)/7  Reducer 2: 0/356
INFO  : Map 1: 5(+2)/7  Reducer 2: 0/356
INFO  : Map 1: 5(+2)/7  Reducer 2: 0/356
INFO  : Map 1: 5(+2)/7  Reducer 2: 0/356
INFO  : Map 1: 6(+1)/7  Reducer 2: 0(+1)/356
INFO  : Map 1: 6(+1)/7  Reducer 2: 0(+1)/356
INFO  : Map 1: 6(+1)/7  Reducer 2: 0(+1)/356
INFO  : Map 1: 6(+1)/7  Reducer 2: 0(+1)/356
INFO  : Map 1: 6(+1)/7  Reducer 2: 0(+1)/356
INFO  : Map 1: 6(+1)/7  Reducer 2: 0(+1)/356
INFO  : Map 1: 6(+1)/7  Reducer 2: 0(+1)/356
INFO  : Map 1: 6(+1)/7  Reducer 2: 0(+1)/356
INFO  : Map 1: 6(+1)/7  Reducer 2: 0(+1)/356
INFO  : Map 1: 6(+1)/7  Reducer 2: 0(+1)/356
INFO  : Map 1: 6(+1)/7  Reducer 2: 0(+1)/356
INFO  : Map 1: 6(+1)/7  Reducer 2: 0(+1)/356
```

Figure 4: Execution of Hive SQL on Tez



チャートタイプ X 軸 node Y 軸 count

Figure 5: Bar graph of the total number of records for each day

records for seven partitions in the data table. In comparison, on Figure 2, a past below of the SQLs shows the percentages of the finished map and reduce job. However, in this case, it shows how many jobs of the total job number are finished. In addition, the log shows how many workers are assigned to map and reduce job. In this study, we set that number of workers to two. On the log, *INFO : Map 1: 5(+2)/7 Reducer 2: 0/356* shows that there are total 7 map jobs and 356 reduce jobs, five map jobs are finished and two workers are assigned to finish last two map jobs. *INFO : Map 1: 6(+1)/7 Reducer 2: 0(+1)/356* shows that six map jobs are finished and one worker is assigned to finish last map job and one worker is assigned to finish the reduce job. Using Tez, we can finish executing Hive SQL quickly compared to MapReduce.

To visualize the result of SQLs on Hue, we can select four types of graphs: Bar, Line, Circle, and Map by selecting x and y axis or latitude and longitude. After finishing the SQL (see Figure 4), we can see the result and visualize the result as shown in Figure 5. We can sort the result by ascending or descending order. In addition, we can save the SQL result to a new Hive table in the database or save it as a csv or excel spread sheet (xls) file and visualize it by using other visualization tools.

5. CONCLUSION

Due to the increase influence of the Internet and web services on the user's lives, many people require intelligent defense systems against cyber attacks. In network security, one of most important problem is to develop an intrusion detection system (IDS), intrusion prevention system (IPS), web application firewall (WAF), etc.

In this paper, we introduced Hadoop, Hive, Tez and designed the analysis environment to handle large packet data. We discussed the advantages of the techniques to analyze pcap files directly and how they are powerful to handle the pcap files. In addition to these techniques, there are many frameworks and tools to analyze many types of data and develop different systems. The development of these techniques is also very active and many libraries are dedicated for these purposes. We can execute some machine learning algorithms and visualize the results by using the machine learning libraries in Apache Spark, Apache Mahout, etc.

As our future work, we will utilize these techniques to analyze the large network packet data in near real-time and apply some machine learning algorithms to develop near real-time automatic detection systems against network attacks.

6. REFERENCES

[1] Apache Hadoop. `http://hadoop.apache.org/`.

[2] Apache Hive. `http://hive.apache.org/`.

[3] Apache Tez. `http://tez.apache.org/`.

[4] Hadoop User Experience. `http://gethue.com/`.

[5] ITU ICT Facts and Figures 2016. `http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2016.pdf`.

[6] Réseaux IP Européens Network Coordination Centre. `https://www.ripe.net/`.

[7] UNB ISCX Intrusion Detection Evaluation DataSet. `http://www.unb.ca/research/iscx/dataset/iscx-IDS-dataset.html`.

[8] Wireshark Performance. `https://wiki.wireshark.org/Performance`.

[9] P. Asrodia and H. Patel. Network traffic analysis using packet sniffer. *International Journal of Engineering Research and Applications (IJERA)*, Vol 2(Issues 3):854 – 856, 2012.

[10] Y. R. Bachupally, X. Yuan, and K. Roy. Network security analysis using big data technology. *Proceeding of IEEE SoutheastCon 2016*, pages 1 – 4, 2016.

[11] K. Kato and V. Klyuev. An intelligent ddos attack detection system using packet analysis and support vector machine. *International Journal of Intelligent Computing Research (IJICR)*, Volume 5(Issues 3/4 Sep/Dec 2014):478 – 485, 2014.

[12] K. Kato and V. Klyuev. Large-scale network packet analysis for intelligent ddos attack detection development. *Proceeding of the 9th International Conference for Internet Technology and Secured Transactions*, pages 360 – 365, 2014.

[13] D. Mistry, P. Modi, K. Deokule, A. Patel, H. Patki, and O. Abuzaghleh. Network traffic measurement and analysis. *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pages 1 – 7, 2016.

[14] Wireshark. `https://www.wireshark.org/`.