



On construction of a network log management system using ELK Stack with Ceph

Chao-Tung Yang¹ · Endah Kristiani^{1,2,3} · Yuan-Ting Wang⁴ · Geyong Min⁵ · Ching-Han Lai¹ · Wei-Je Jiang¹

Published online: 10 May 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

A log management system is essential for the networks administrator. With a log management tool, we can collect, store, analyze, archive, and finally dispose of the log information. In this paper, we propose the architecture model of a log management system using ELK Stack with Ceph to provide a safe network, good Wi-Fi signal strength, and adequate backup data mechanism. In this case, we use our campus data of Wi-Fi log and NetFlow log. First, we collect and store data of our Wi-Fi log using Filebeats tool, and then, we use Elasticsearch, Logstash, and Kibana Stack to visualize the Wi-Fi log data. Second, we collect and store our NetFlow log using NFDUMP, and then, we also use ELK Stack to visualize the NetFlow log data. Third, we integrate the Wi-Fi log and NetFlow log data in one architecture using a distributed storage Ceph file system (CephFS). Moreover, we also compare the performance of RADOS Gateway and CephFS for better storage mechanism.

Keywords Log management system · ELK Stack · Ceph · Alluxio · Elasticsearch · Logstash · Kibana

1 Introduction

In recent years, the cases of network attacks have become increasingly significant. These have led the authority to pay more attention to such issues. Decision makers at all levels of organizations need to obtain safe, accurate, cohesive information, business intelligence collecting for taking further actions, analyze and insight, and make effective tactical and strategic decisions. Our university data center department needs to provide a monitoring or tracking system that can track abnormal NetFlow logs to offer uninterrupted network logs. This service will be used by our department to make the right decision [1–3].

✉ Chao-Tung Yang
ctyang@thu.edu.tw

Extended author information available on the last page of the article

In terms of Wi-Fi data, nowadays, Internet usage has been changed from Internet cable to Wi-Fi connection. The difference factor between the Internet and Wi-Fi is the Internet which allows us to access information from other computers across the world using a specific language (protocol). While Wi-Fi is a way to connect the same way with a network cable, just without the actual cable, when the wireless network has a weak signal, transfer information across the network could be slower, or we could not access certain parts of the network. Related to this issue, our campus also always finds out this Wi-Fi weakness or unavailable signal problem. From here, we want to resolve this problem with the analysis of each location signal strength in the campus area. For this requirement, we need a tool that can solve this problem and we decided to use the ELK Stack as the best decision-maker device. After finding the problem, we will analyze case by case based on log conditional [4].

To provide a secure network, good Wi-Fi signal strength, and adequate backup data mechanism, we propose the architecture model of a log management system using Elasticsearch, Logstash, and Kibana (ELK) Stack with Ceph. In this case, we use our campus data of Wi-Fi log and NetFlow log. First, we collect and store data of our Wi-Fi log using Filebeats tool, and then, we use ELK Stack to visualize the Wi-Fi log data. Second, we collect and store our NetFlow log using NFDUMP, and then, we also use ELK Stack to visualize the NetFlow log data. Third, we integrate the Wi-Fi log and NetFlow log data in one architecture using a distributed storage [5] and Alluxio. Moreover, to verify the better storage mechanism in Ceph, we compare the performance of RADOS Gateway (RADOSGW) and Ceph file system (CephFS).

In particular, this paper aims to propose a log management system using ELK Stack with Ceph, with specific purposes as follows:

- To collect, store, and visualize Wi-Fi log data using Filebeats tool and ELK Stack.
- To collect, store, and visualize NetFlow data using NFDUMP and ELK Stack.
- To integrate the Wi-Fi log and NetFlow log data in one architecture using a distributed storage CephFS.
- To compare the performance of the storage mechanism in Ceph between RADOSGW and CephFS.

The rest of the paper is organized as follows. Section 2 describes the background and related work. Section 3 presents the system architecture. Section 4 shows the experimental results. The last one, Sect. 5, provides a conclusion and the future work of this paper.

2 Background review and related works

In this section, we provide several components that are approaching in this paper: ELK Stack, Filebeat, NFDUMP, Ceph, and Alluxio. The next sections discuss each component in more detail.

2.1 ELK Stack

The ELK Stack [6] consists of three open-source products: Elasticsearch, Logstash, and Kibana; all these three parts are developed, managed, and maintained by Elastic. Elasticsearch is a NoSQL database that is based on the Lucene search engine. Logstash is a log pipeline tool that accepts inputs from various sources, executes different transformations, and exports the data to various targets. Kibana is a visualization layer that works on top of Elasticsearch [7, 8].

Logstash is a free open-source data engine that for receiving, transforming, and outputting data from various sources. Logstash can centralize the data from disparate sources and collect them into one central point. Clean up and transform all log content with advanced downstream visualization and deepen analytic [9, 10]. In a log system, we can have more than one Logstash services. The purpose of Logstash is that our system can collect data with high speed and optimized. This architecture form is known as completed or cluster Logstash. While Logstash accelerates our insights via greater volume, the data can be transferred double or triple times of speed [11].

Kibana makes it convenient to understand large volumes of data with any visual charts. It is easy to learn, and a web browser-based interface enables us to quickly create and share dynamic dashboard that displays changes to Elastic queries directly. Because of Kibana only available on the localhost, we use Nginx [12] to provide access to a web browser. Therefore, we can submit searching, filtering queries, and data exporting to a file or document.

2.2 Filebeat

Filebeat [13] is a free open-source shipper for log file data. As the next generation for Logstash Forwarder, Filebeat tails log and quickly send this information to Logstash for further parsing and enrichment or to Elasticsearch for centralized storage and analysis system.

2.3 NFDUMP

NFDUMP [14] is a tool to collect and process NetFlow data on the command line as a part of the NfSen project. NFDUMP aims to analyze NetFlow data from history as well as to track interesting traffic patterns continuously. The data are organized in a time series based which update typically every 5 min.

2.4 Ceph

Ceph [15] is an open-source, massively scalable, and software-defined storage system. It provides file system storage, data blocks, and user objects on a single platform. Running it on the required system allows it to save costs and increase flexibility. Under the same platform, Ceph also provides three important services

of object storage, block storage, and file system. It also has strong mine development capabilities and is not limited to any hardware specifications. It is a popular open source for a large-scale engine [16, 17].

Ceph storage clusters are designed to run on commodity hardware, using an algorithm called CRUSH (Controlled Replication Under Scalable Hashing) to ensure data are evenly distributed across the cluster and that all cluster nodes can retrieve data quickly without any centralized bottlenecks. Ceph storage system is designed to be self-healing and self-managing and strives to save the budget to operate the service. Ceph uniquely delivers object, block, and file storage in one unified system. Ceph object storage is accessible through like Amazon S3, Open Stack Swift and also provides a native API for integration with software applications. Ceph block storage makes use of a Ceph block device, which is a virtual disk that can be attached to bare-metal Linux-based servers or virtual machines. The Ceph Reliable Autonomic Distributed Object Store (RADOS) provides block storage capabilities. The Ceph RADOS block device is integrated to work as a back end with Open Stack block storage. Ceph file storage makes use of the Portable Operating System Interface (POSIX)-compliant CephFS to store data in a Ceph storage cluster. Figure 1 describes Ceph Architecture.

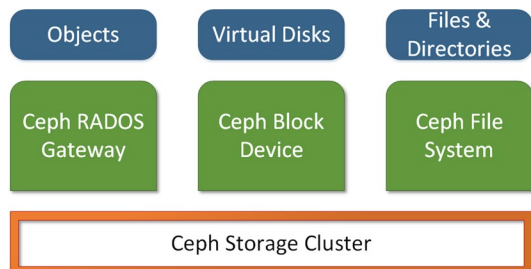
Ceph consists of three component services, which are RADOS Gateway (Ceph Object Gateway Daemon), block device, and CephFS.

1. RADOSGW is a bucket-based REST gateway, compatible with S3 and Swift.
2. RBD is a reliable and fully distributed block device, with a Linux kernel client and QEMU/KVM driver.
3. CephFS is a POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE.

2.5 Alluxio

Alluxio [18] is an open-source memory-speed virtual distributed storage system. It unifies data access and bridges computation frameworks and the underlying storage system. Applications only need to connect with Alluxio to access data stored in any underlying storage systems. Moreover, Alluxio memory-centric architecture enables data access orders of magnitude faster than existing solutions.

Fig. 1 CEPH architecture



In the big data ecosystem, Alluxio lies between computation frameworks or jobs, such as Apache Spark, Apache Map Reduce, or Apache Flink, and various kinds of storage systems, such as Amazon S3, Google Cloud Storage, Open Stack Swift, GlusterFS, HDFS, and Ceph. Alluxio brings significant performance improvement to the big data ecosystem. Alluxio users can work with frameworks of their choice. Additionally, Alluxio enables new workloads across different storage systems. Various frameworks can share data efficiently with each other. Alluxio's unified namespace enables applications to interact with data in any storage system in memory speed. This future proven architecture enables users to extract value from data in any storage faster. Figure 2 shows Alluxio Architecture.

2.6 Related works

C.T. Yang et al. [16] presented their research in applying Ceph storage with big data performance testing to solve the best reading and write speed performance and data backup. The usage of Ceph storage aims to reduce high-risk data corrupt and improve the reading and writing storage performance.

Nguyen et al. [19] presented eLMS as an efficient and scalable log management system. They collected the log file from multiple servers, indexed, and analyzed based on ELK Stack.

Kumar et al. [20] presented a security ecosystem of fatal attacks and threats. Their work helps in evolving a technique for log aggregation and analysis in real time through the dashboard and terminal display, and alerts are generated based on various malicious conditions. Their work aims to increase knowledge about traffic patterns and trends, besides to perform authentic time decision on maleficent traffic using ELK Stack.

Anastopoulos et al. [21] demonstrated a real application of log management infrastructure from a large number of various devices. In their work, they utilize social network analysis to propose a novel methodology for risk-based asset prioritization that leads to the dynamic design of a log management infrastructure.

Miao et al. [22] proposed a measurement study of Wi-Fi network management from multiple dimension, i.e., server, temporal, spatial, and traffics analysis. Their work aims to improve and manage a large scale of Wi-Fi network based on their multiple dimension analysis.

Qu et al. [23] discussed a hybrid storage solution for a real environment using a distributed storage system based on Ceph. They stated that Ceph has an attractive solution as back-end storage. Dubey et al. [24] stated that data visualization makes



Fig. 2 Alluxio architecture

data have more meaning by storytelling. They used the GitHub repository for data visualization using Elasticsearch and Kibana.

Kumar et al. [25] compared and analyzed Elasticsearch, MongoDB, and Hadoop in a big data processing. Based on their comparison study, they stated that Elasticsearch is a search engine which provides a way to organize data so that it can be easily accessed, also, a tool for querying the word written.

Almohannadi et al. [26] proposed a new threat intelligence technique which is evaluated by analyzing honeypot log data to identify the behavior of attackers and find attack patterns. The log data were collected from honeypot and then analyzed by using ELK Stack.

Our work manages to build a monitoring system using ELK Stack to handle weakness or unavailable Wi-Fi signal problems and to monitor the network attacks in a real-time environment. Also, we quantify and provide a backup mechanism using Ceph to handle a vast and evolving network and Wi-Fi log data.

3 System architecture

In this section, we present our system architecture in three stages. First, we collect and store data of our Wi-Fi log using Filebeats tool, and then, we use ELK Stack to visualize the Wi-Fi log data. Second, we collect and store our NetFlow log using NFDUMP, and then, we also use ELK Stack to visualize the NetFlow log data. Third, we integrate the Wi-Fi log and NetFlow log data in one architecture using a distributed storage Ceph and Alluxio.

3.1 Wi-Fi log system architecture

In this section, we present a system architecture of Wi-Fi log data. We provided two servers as our experimental environment. The first server is used to install Filebeat tool. Filebeat is a log data shipper for local files. The second server is used to set up ELK Stack environment [27]. Servers will communicate with each other constantly

Fig. 3 Wi-Fi log system architecture



if the new data are coming. Figure 3 shows our architecture system that consists of Logstash, Elasticsearch, Kibana, and Filebeat.

This system is divided into two sides as shown in Fig. 4. The left side is our Wi-Fi log located and Filebeat installed. The right side is our ELK Stack Server that the Wi-Fi log will, and the endpoint of data will be showed out on Kibana platform.

3.2 NetFlow log system architecture

In this section, we present our system architecture and data flow. At first, we download the file from the server with a shell script, which collected the NetFlow data to the local computer. Then, we use ELK Stack to perform preliminary collation and analysis. Logstash will collect and filter log data constantly. Elasticsearch will store data input by Logstash. Finally, we will use Kibana to visualize the data on the Web site. Figure 5 describes the process.

The NetFlow data from multiple APs in the areas of the campus exported via Nfdump in CSV format (Fig. 6).

3.3 The performance comparison of RADOSGW and CephFS system architecture

Figure 7 describes comparison mechanism architecture that we deployed on our system. In this figure, we present rough sketch system that consists of CEPH, Alluxio, and Hadoop.

First, we set three kinds of sample data size: 5 GB, 10 GB, and 15 GB. Through map <key, value>, sort by key, and merge [key, [value-1, value-2, value-n]] algorithm, the data are sent into Alluxio memory-speed virtual storage system. We also activate S3 API (one of CEPH components) in the Alluxio file configuration; through S3 and RADOS Gateway, the data are also stored into the Object Storage Daemon (OSDs). Second, we use the same data load as the first one system, but in

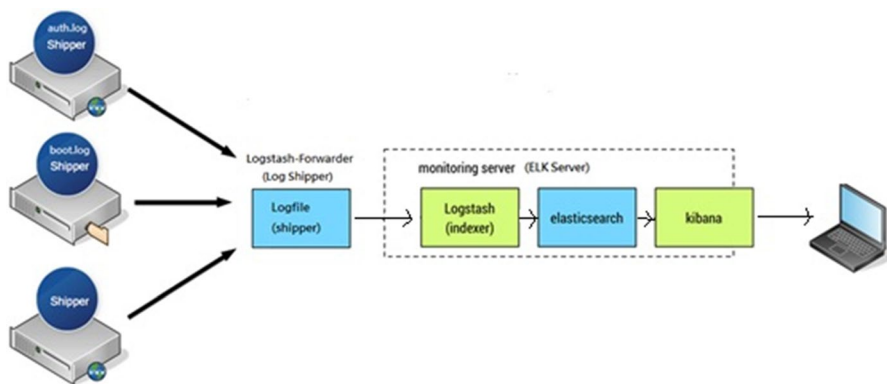


Fig. 4 Wi-Fi log data architecture

Fig. 5 NetFlow log system architecture

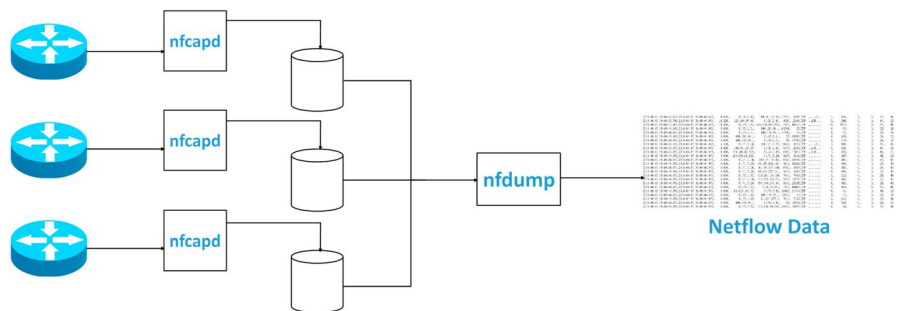
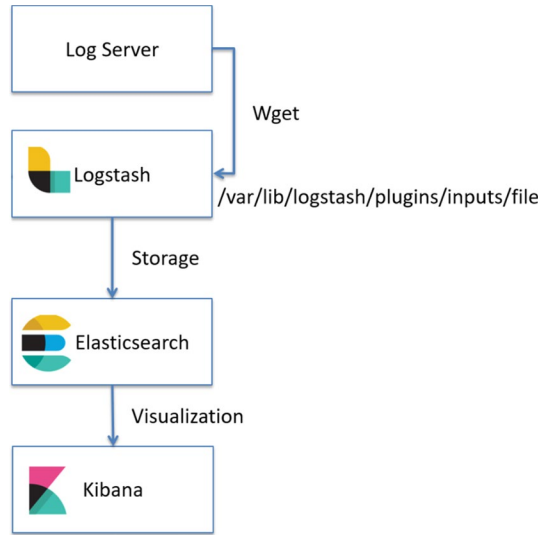


Fig. 6 NetFlow log data architecture

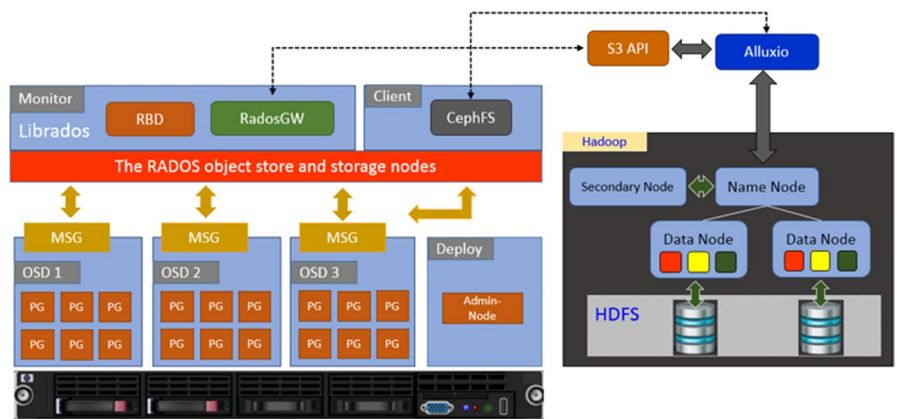


Fig. 7 The performance comparison of RADOSGW and CephFS system architecture

Table 1 Hardware specifications

Item	Disk	CPU	RAM
ESXi	2 TB	10 CPUs x Intel(R) Core(TM) i7-6950X CPU @ 3.00 GHz	128 G
Ceph OSD	10 TB	Intel(R) Core(TM) i7-3970X CPU @ 3.50 GHz	64 G

Table 2 Machine specifications

	Disk	CPU	RAM	Version
ELK Stack	800 GB	6 cores	8G	6.3.1
Ceph-admin	16 GB	1 cores	64G	ceph 10.2.10, ceph-deploy 1.5.39
Ceph-osd1	250 GB	1 cores	1G	ceph 10.2.10
Ceph-osd2	250 GB	1 cores	1G	ceph 10.2.10
Ceph-osd3	250 GB	1 cores	1G	ceph 10.2.10
Ceph-osd4	10 T	8 cores	16G	ceph 10.2.10

this system OSDs directly connect to the Alluxio plug-in. The second experiment reduces the S3 API and RADOSGW levels.

4 Experimental results

In this section, we describe our environmental system for integrating NetFlow log and Wi-Fi log data.

4.1 Hardware and software configuration

The following are the hardware and virtual machine specifications we used when built the experimental environment, the first machine we used to configure all our virtual machines, including ELK Stack [28] and Ceph. Then, we installed VMWare ESXi on a physical machine and create some VM (virtual machine) for deploying environment to make us manage machine easier. Moreover, we chose the machine that needed less resource to deploy on VM. The second machine is used to build Ceph OSD to improve overall storage space (Tables 1, 2).

4.2 Wi-Fi log visualization

Figures 8 and 9 show the Aruba Airwave network management. Airwave uses VisualRF to allow for time lapse of Wi-Fi coverage. This module enables wireless network engineering to record and replay 24 h of RF heat mapping.



Fig. 8 Aruba airwave dashboard



Fig. 9 Aruba airwave dashboard

Dashboard menu is a platform that we can show our chart more than one type. Sometimes we need to compare one with others, so this platform is suitable for us to carry out, for example the top of Wi-Fi used based on certain date range per location, the figure of Wi-Fi used per location tendency, etc.

Wi-Fi log is mapped on our campus area which the larger circle node is, the more users are accessing as shown in Fig. 10.

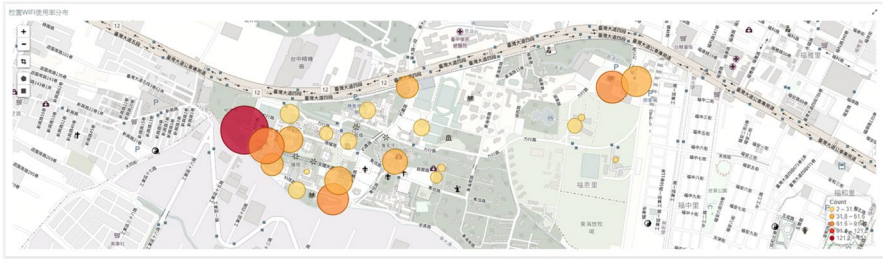


Fig. 10 Wi-Fi log data on the campus map

4.3 NetFlow log visualization

On the NetFlow log data, we set several of cyber-attack like CodeRed, SQL Slammer, Worm Sasser, ICMP, and SYN flooding.

Figure 11 shows the CodeRed attack. The CodeRed worm attempts to connect to randomly generated IP addresses on the IIS server. An abnormally high number of entries with the same source IP address indicates an infected device.

Figure 12 shows the SQL Slammer attack. SQL Slammer is computer worm that caused a denial of service on the host and dramatically slowed down the Internet traffic.

Figure 13 shows the Sasser attack. Sasser is a computer worm that affects computers running vulnerable versions of Windows XP and Windows 2000. Sasser spread through a vulnerable port by exploiting the system.

Figure 14 shows a flooding DoS attack. ICMP flood is a method that relies on sending a large number of ping packets or sending a malformed ping packet. SYN flood is a result of TCP/SYN packets flooding sent by host, mostly with a fake address.

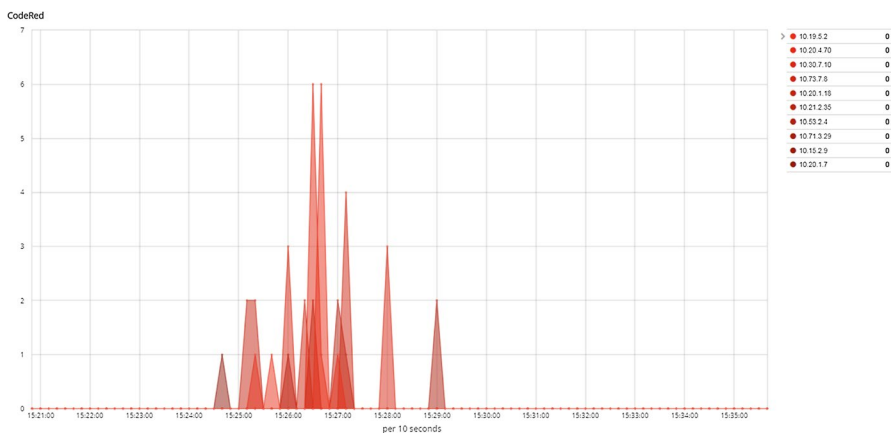


Fig. 11 CodeRed NetFlow attack

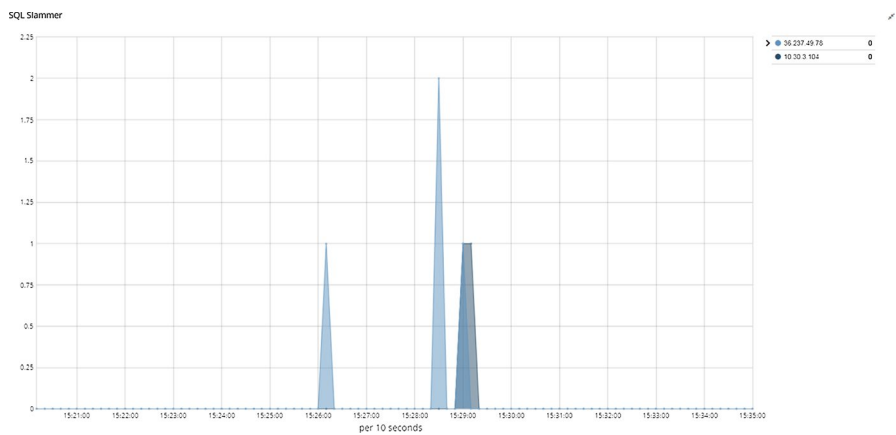


Fig. 12 SQL slammer NetFlow attack

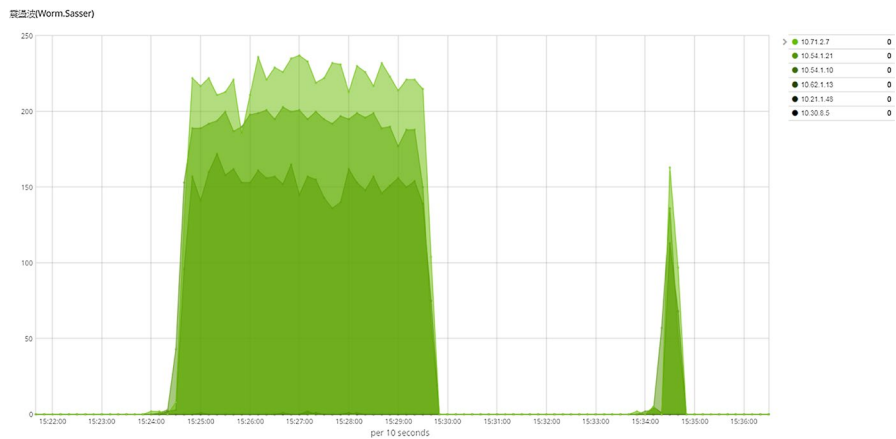


Fig. 13 Worm Sasser NetFlow attack

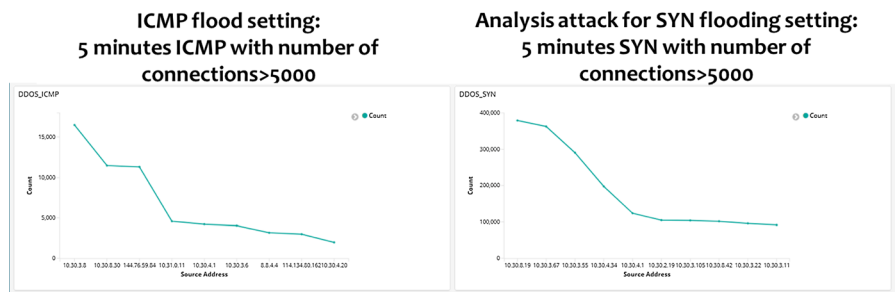


Fig. 14 ICMP and SYN NetFlow flooding

We have some public IP in our network flow. So, we put IP in GeoIP2 database and translate it to latitude, longitude, city, and country. The GeoIP extension allows us to find the location of an IP address. City, state, country, longitude, latitude, and other information, such as ISP and connection type, can be obtained with the help of GeoIP. In Fig. 15, we can find that most flow data are from Taiwan.

4.4 The performance comparison between RADOSGW and CephFS

In this section, we present the performance comparison results of RADOSGW and CephFS using HiBench tool. In this case, we measured IOPS (input/output operations per second). IOPS is a performance measurement used for Ceph storage. The purpose is to give the response time on workload per seconds, which is the higher the value, the better the performance. First, we created three OSDs (osd1, osd2, and osd3) for RADOSGW S3 API. Second, we created one OSD for CephFS. Both mechanisms of speed comparisons in reading and writing are on 1G, 3G, and 5G of networking. According to the experiments, we found that on the RADOSGW, the average of reading speed on OSDs is at 137, 274, and 210 response time per seconds, respectively, while on the CephFS is at 252, 240, and 461 response time per seconds, respectively. In terms of writing speed, the average on OSDs is at 846, 1078, and 1155 response time per seconds, respectively, whereas on the CephFS is at 2099, 2044, and 2009 response time per seconds, respectively. From these experiments, we can conclude that CEPHFS performance is better than through RADOSGW S3 API. Figures 16 and 17 show the IOPS performance test with read, write, and randomize for each OSD.

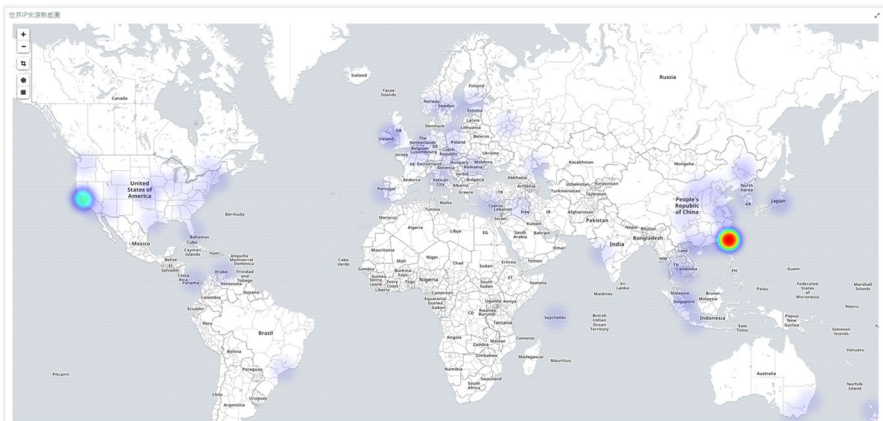


Fig. 15 NetFlow map

	osd1			osd2			osd3			CephFS		
	1G	3G	5G	1G	3G	5G	1G	3G	5G	1G	3G	5G
S.Read	156	383	362	131	306	134	123	133	134	252	240	461
S.Write	645	1103	1198	749	1022	1159	1143	1109	1109	2099	2044	2009
Rand.Read	56	104	104	45	80	71	49	82	152	54	51	53
Rand.Write	98	107	55	88	94	81	92	111	58	37	34	31
S.Read(30%)	121	208	218	110	259	254	126	263	241	304	314	317
S.Write(30%)	50	90	94	46	112	110	53	113	103	130	134	135
Rand.Read(30%)	37	53	60	37	34	52	37	54	54	37	34	24
Rand.write(30%)	15	21	24	15	14	20	15	21	21	15	14	10

Fig. 16 The comparison of IOPS performance test with read, write, and randomize for each OSDs

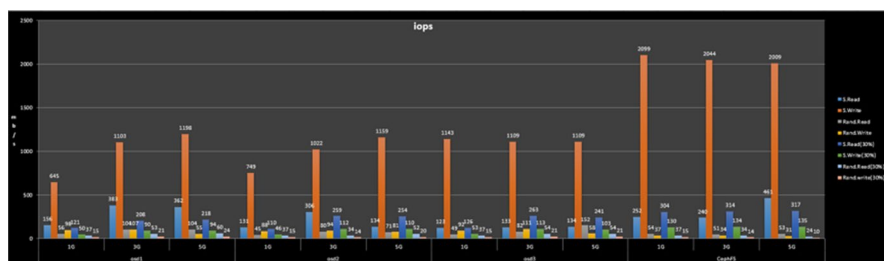


Fig. 17 The graph of IOPS performance test with read, write, and randomize for each OSDs

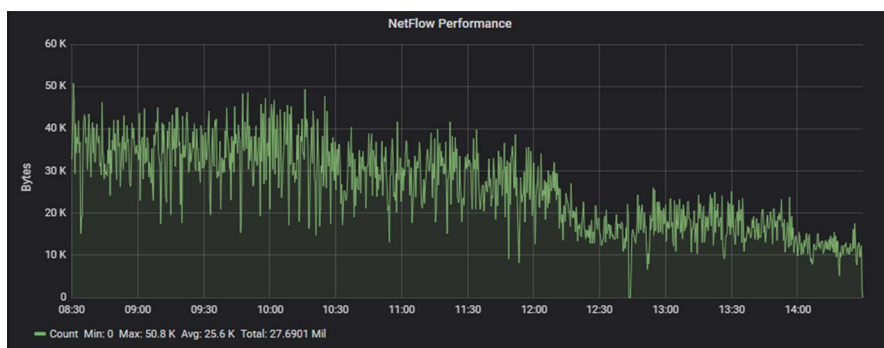


Fig. 18 6 h NetFlow performance test

4.5 NetFlow performance

Figure 18 describes the data traffic graph in the working hours between 08:30 and 14:00 on our campus. It can be said that the size is below 50 K Bytes, and the average is about 25.6 K.

4.6 Discussions

Our findings suggest that in the organizations there are three essential aspects in providing the management of Wi-Fi and network services for user and decision makers.

A secure network, good Wi-Fi signal strength, and adequate backup data mechanism are the significant factors of available network services. By using the ELK Stack, we can monitor, analyze, and store the network data in a real-time monitoring system. However, for advanced analysis and sufficient storage, we should consider the backup data mechanism in a better way. We can use Elasticsearch storage as a short-term repository and use Ceph as a long-term repository. Therefore, there is a process to take into account how long the short-term data will keep in the Elasticsearch and then move the log data to Ceph for long-term storage. Finally, we should consider in disposing of the network log data.

5 Conclusion and future work

In this section, we stated the concluding remarks and the future works of this study.

5.1 Concluding remarks

This paper implements the Ceph storage system to store the vast amounts of data generated each day. By using Wi-Fi and NetFlow log data, we can effectively monitor signal coverage, number of Internet user via Wi-Fi, and analyze the cyber-attacks network. ELK Stack provides related information and analysis. The available graphical displays can help regulators have a better view of analysis results. Through the analysis results, we can detect cyber-attacks to make the appropriate response. This paper presents how to use a large amount of data for attack analysis and use the analysis results to prevent specific IP. Moreover, we found that the CephFS storage mechanism has better performance than RADOS Gateway performance because the data are directly connected from HDFS to OSDs.

5.2 Future work

In the future, we hope to classify malicious attacks through machine learning. We will make an adequate response to different levels, reduce unnecessary personnel costs, and use the different sampling algorithms to increase the accuracy of data analysis to reduce false returns. We hope that through the high accuracy of machine learning and data analysis, it can provide adequate support for previously undiscovered malicious threat events.

Acknowledgements This work was supported in part by the Ministry of Science and Technology, Taiwan (ROC), under Grants Number 107-2221-E-029-008, 107-2218-E-029-003 and 106-3114-E-029-003.

References


1. Rudd J, Sullivan P, King M, Bouchard F, Turner K, Olson M, Schroeder K, Kaplan A (2009) Education for a smarter planet: the future of learning. 2012-09-09. <http://www.redbooks.ibm.com/redpapers/pdfs/redp4564.pdf>. Accessed 29 Nov 2018

2. Raghav R, Akash S, Shobha G, Poonam G, Pratiba D, Singh A (2016) Design and development of generic web based framework for log analysis. In: 2016 IEEE Region 10 Conference (TEN-CON). IEEE, pp 232–236
3. Awad M, Menasc DA (2015) Automatic workload characterization using system log analysis. In: Computer Measurement Group Conference on Performance and Capacity, San Antonio, TX
4. Kononenko O, Baysal O, Holmes R, Godfrey MW (2014) Mining modern repositories with elasticsearch. In: Proceedings of the 11th Working Conference on Mining Software Repositories. ACM, pp 328–331
5. Yang C-T, Chen S-T, Den W, Wang Y-T, Kristiani E (2019) Implementation of an intelligent indoor environmental monitoring and management system in cloud. *Fut Gener Comput Syst* 96:731–749
6. ELK Stack (2018) <https://www.elastic.co/elk-stack>
7. The Complete Guide to the ELK Stack 2018, <https://logz.io/learn/complete-guide-elk-stack/intro>
8. Gupta P, Nair S (2014) Survey paper on elastic search. *Int J Sci Res (IJSR)* 5(1):4
9. Sanjappa S, Ahmed M (2017) Analysis of logs by using logstash. In: Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications. Springer, Singapore, pp 579–585
10. Chuvakin A, Schmidt K, Phillips C (2012) Logging and log management: the authoritative guide to understanding the concepts surrounding logging and log management. Newnes
11. Gormley C, Tong Z (2015) Elasticsearch: the definitive guide: a distributed real-time search and analytics engine. O'Reilly Media Inc, Sebastopol
12. Nginx introduction (2018) <http://www.webopedia.com/TERM/N/nginx.html>
13. <https://www.elastic.co/products/beats/filebeat> (2018)
14. <http://nfdump.sourceforge.net/> (2018)
15. CEPH (2018) <https://ceph.com/>
16. Yang C-T, Chen C-J, Chen T-Y (2017) Implementation of Ceph storage with big data for performance comparison. In: International Conference on Information Science and Applications. Springer, Singapore, pp 625–633
17. Zhan K, Piao AH (2016) Optimization of Ceph reads/writes based on multi-threaded algorithms. In: 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, pp 719–725
18. <https://www.alluxio.org/> (2018)
19. Nguyen VN, Tran VC (2016) An efficient log management system. *VNU J Comput Sci Commun Eng* 32(2):43–48
20. Kumar A, Bandyopadhyay A, Bhoomika H, Singhania I, Shah K (2018) Analysis of network traffic and security through log aggregation. *Int J Comput Sci Inf Secur* 16(6)
21. Anastopoulos V, Katsikas S (2018) Design of a dynamic log management infrastructure using risk and affiliation network analysis. In: Proceedings of the 22nd Pan-Hellenic Conference on Informatics. ACM, pp 52–57
22. Miao C, Wang J, Wang H, Zhang J, Zhou W, Liu S (2018) A multi-dimension measurement study of a large scale campus Wi-Fi network. In: 2018 IEEE 43rd Conference on Local Computer Networks (LCN). IEEE, pp 351–359
23. Qu Z, Xie C, Liu C (2018) The study of mixed storage scheme of private cloud platform based on Ceph. In: 2018 International Conference on Network, Communication, Computer Engineering (NCCE 2018). Atlantis Press
24. Dubey S, Balaii B, Rao D, Rao D (2018) Data visualization on GitHub repository parameters using elastic search and Kibana. In: 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI). IEEE, pp 554–558
25. Kumar P, Kumar P, Zaidi N, Vijay SR (2018) Analysis and comparative exploration of elastic search, MongoDB and Hadoop big data processing. In: Soft computing: theories and applications. Springer, Singapore, pp 605–615
26. Almohannadi H, Awan I, Al Hamar J, Cullen A, Disso JP, Armitage L (2018) Cyber threat intelligence from honeypot data using elasticsearch. In: 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA). IEEE, pp 900–906
27. Prakash T, Kakkar M, Patel K (2016) Geo-identification of web users through logs using ELK Stack. In: 2016 6th International Conference on Cloud System and Big Data Engineering (Confluence). IEEE, pp 606–610

28. Elasticsearch guidance page, <https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started.html> (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Chao-Tung Yang¹  · **Endah Kristiani**^{1,2,3} · **Yuan-Ting Wang**⁴ · **Geyong Min**⁵ · **Ching-Han Lai**¹ · **Wei-Je Jiang**¹

Endah Kristiani
endahkristi@gmail.com

Yuan-Ting Wang
yttom@cht.com.tw

Geyong Min
g.min@exeter.ac.uk

Ching-Han Lai
a8901256609@gmail.com

Wei-Je Jiang
s22775605@gmail.com

- ¹ Department of Computer Science, Tunghai University, No. 1727, Sec. 4, Taiwan Boulevard, Xitun District, Taichung 40704, Taiwan, ROC
- ² Department of Industrial Engineering and Enterprise Information, Tunghai University, No. 1727, Sec. 4, Taiwan Boulevard, Xitun District, Taichung 40704, Taiwan, ROC
- ³ Department of Informatics, Faculty of Engineering and Computer Science, Krida Wacana Christian University, Jakarta 11470, Indonesia
- ⁴ Cloud Computing Laboratory, Chunghwa Telecom Laboratories, No. 99, Dianyan Rd., Yangmei Dist., Taoyuan City 326, Taiwan, ROC
- ⁵ Department of Mathematics and Computer Science, College of Engineering, Mathematics and Physical Science, University of Exeter, Exeter EX4 4QF, UK