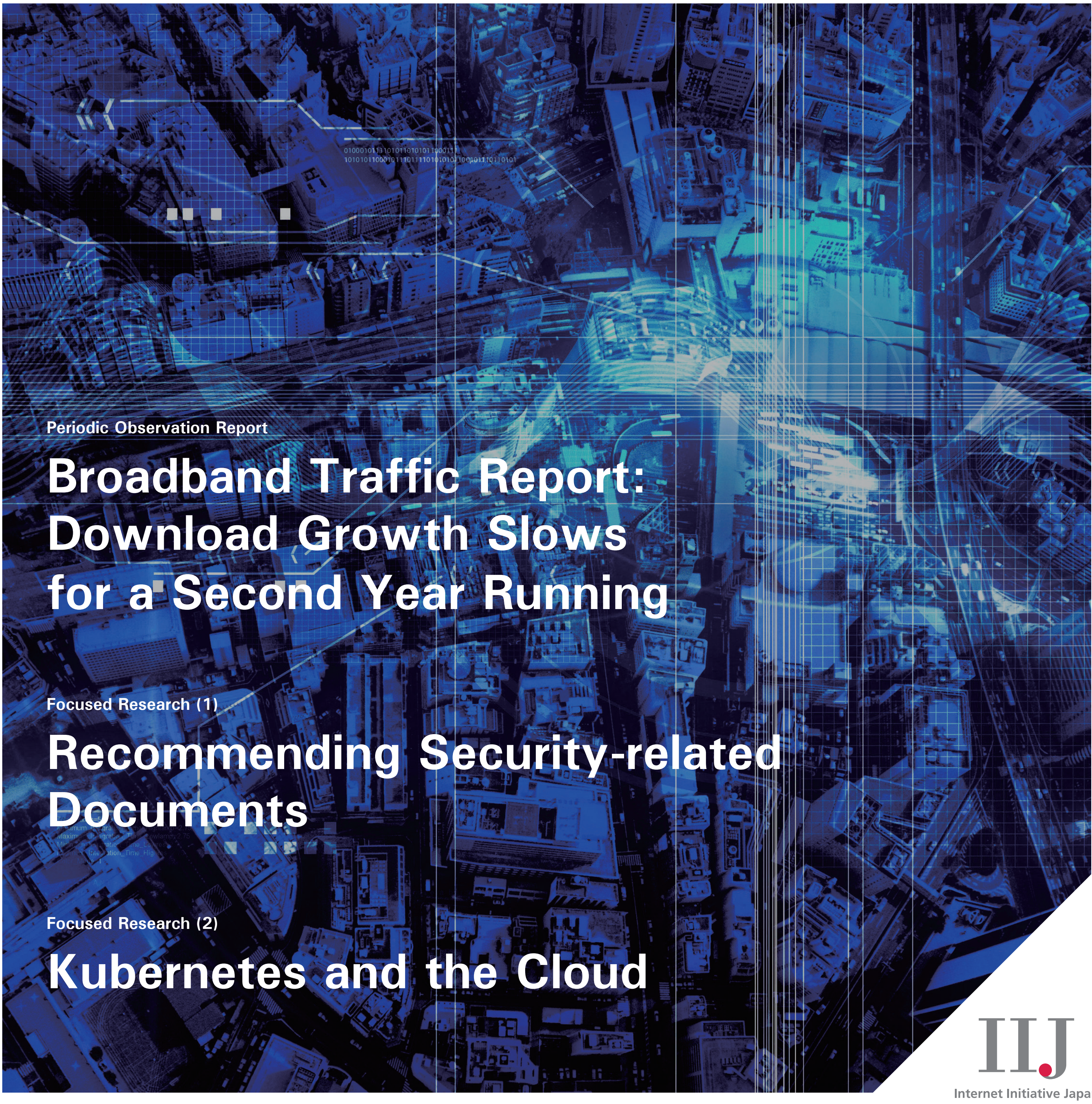
IIR

Internet Infrastructure Review

Nov.2018
Vol. 40



Periodic Observation Report

Broadband Traffic Report: Download Growth Slows for a Second Year Running

Focused Research (1)

Recommending Security-related Documents

Focused Research (2)

Kubernetes and the Cloud

IIJ
Internet Initiative Japan

Internet Infrastructure Review
November 2018 Vol.40

Executive Summary

The Telecommunications Business Act, which govern the telecommunications business in Japan, was revised in 2015. It was stipulated that three years from when the revisions took effect, a review of post-revision implementation status would be carried out and measures taken if necessary. Accordingly, this past August 23, Japan's Ministry of Internal Affairs and Communications consulted the Information and Communications Council regarding a “Comprehensive Examination of Competition Rules and Other Considerations for the Telecommunications Business Field”, and work on this with a view out to around 2030 will now begin in earnest.

The scope of this work is wide-ranging and includes the vision for communications networks as a whole, the state of communications infrastructure development, the state of network neutrality, the state of efforts to deal with issues related to platform services, the state of efforts to ensure a competitive environment in the mobile communications market, and the state of consumer protection rules.

The development of technologies and services by a diverse array of players is ongoing in the Internet arena, driven in part by the advance of technologies such as network virtualization and software control. And with the use of AI, IoT, and 5G technologies set to ramp up ahead, not only will technology development bear close watching, so too will the development of legal frameworks and other considerations.

The IIR introduces the wide range of technology that IIJ researches and develops, comprising periodic observation reports that provide an outline of various data IIJ obtains through the daily operation of services, as well as focused research examining specific areas of technology.

The periodic observation report in Chapter 1 is a broadband traffic report. This report, which we have provided every year since 2009, presents an analysis of traffic over the broadband access services operated by IIJ. The Ministry of Internal Affairs and Communications' “Summary/Estimate of Internet Traffic in Japan” (in Japanese) provides an overall tally of traffic, but in our report, we present an analysis of the distribution of daily traffic volume and the volume of traffic by port. Our results indicate that although traffic growth is slowing, traffic itself continues to rise, and that HTTPS, which has expanded considerably since around four years ago, is also on the rise. With web browsers displaying messages saying that HTTP is unsafe and HTTP-only sites being pushed down in search engine rankings, traffic looks likely to increasingly shift from HTTP to HTTPS ahead.

Chapter 2 is our first focused research report, in which we describe natural language processing techniques for dealing with unstructured information and our experiments with topic modelling. In the area of information security, structured information that is amenable to automated processing, such as IP address blacklists and SCAPs, is widely used in support systems, but challenges remain when it comes to making use of information that does not easily lend itself to automated processing, such as images and documents written in natural languages. With that in mind, we developed a prototype recommender system to make use of this sort of unstructured information in security tasks. Although our results were less than satisfying, we did come away with a real sense that unstructured information could be used under certain conditions.

Chapter 3, our second focused research report, looks at Kubernetes. Two names that increasingly come up when collecting information related to cloud computing are Docker and Kubernetes. Both are at the core of the latest container technologies. In this volume, after going over the functions and roles of Docker and Kubernetes and explaining the rationale for using Kubernetes with IaaS and hybrid cloud services, we then introduce the IKE (IIJ Container Engine for Kubernetes) system built by IIJ. We explain what sort of environment a Kubernetes container cluster actually is and what it aims to achieve, so the discussion is likely to be useful for anyone looking to work with one.

Through activities such as these, IIJ strives to improve and develop its services on a daily basis while maintaining the stability of the Internet. We will continue to provide a variety of services and solutions that our customers can take full advantage of as infrastructure for their corporate activities.

Junichi Shimagami

Mr. Shimagami is a Senior Executive Officer and the CTO of IIJ. His interest in the Internet led to him joining IIJ in September 1996. After engaging in the design and construction of the A-Bone Asia region network spearheaded by IIJ, as well as IIJ's backbone network, he was put in charge of IIJ network services. Since 2015, he has been responsible for network, cloud, and security technology across the board as CTO. In April 2017, he became chairman of the Telecom Services Association of Japan MVNO Council..

Broadband Traffic Report: Download Growth Slows for a Second Year Running

1.1 Overview

In this report, we analyze traffic over the broadband access services operated by IIJ and present the results each year[*1][*2][*3][*4][*5][*6][*7][*8][*9]. Here, we again report on changes in traffic trends over the past year, based on daily user traffic and usage by port.

Figure 1 shows the overall average monthly traffic trends for IIJ's broadband services and mobile services. IN/OUT indicates the direction from the ISP perspective. IN represents uploads from users, and OUT represents user downloads. Because we cannot disclose specific traffic numbers, we have normalized the data, setting the latest OUT observation in each dataset to 1. Starting with this edition of the report, the broadband data include IPv6 IPoE traffic. The thin line labeled "broadband-IPoE" excludes IPv6 IPoE traffic. IPv6 traffic on IIJ's broadband services comprises both IPoE and PPPoE traffic[*10], but IPoE traffic does not pass directly through IIJ's network as we use Internet Multifeed Co.'s transix service for IPoE, and IPoE is therefore excluded from the analysis that follows here. As of June 2018, IPoE accounted for 12% of IN and 8% of OUT broadband traffic overall.

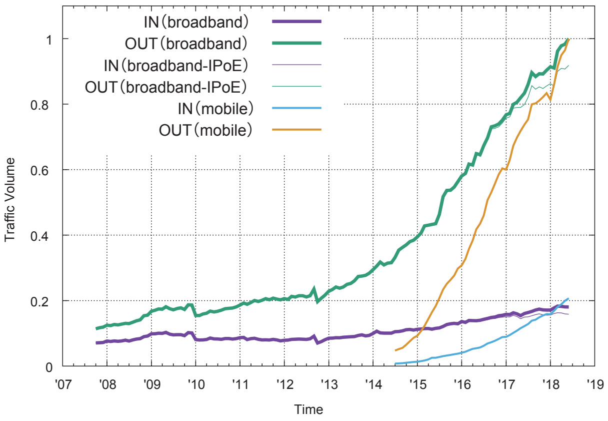


Figure 1: Monthly Broadband and Mobile Traffic over Time

Growth in both broadband and mobile traffic slowed temporarily in the latter half of last year, but that growth has picked up again this year and returned to its original trajectory. Over the past year, broadband IN traffic increased 12% and broadband OUT traffic increased 20%. The respective figures a year earlier were 10% and 25%, and two years earlier 18% and 47%, so growth in download volume has been slowing for two years running. For mobile, we only have data for the past four years. Mobile IN traffic increased 69% and OUT traffic increased 36% over the past year. Although these figures represent a slowing of growth compared with 103% and 70% a year ago, the level of growth remains high. That said, the total volume of mobile traffic remains an order of magnitude lower than broadband traffic.

1.2 About the Data

As with previous reports, for broadband traffic, our analysis uses data sampled using Sampled NetFlow from the routers that accommodate the fiber-optic and DSL broadband customers of our personal and enterprise broadband access services. For mobile traffic, we use access gateway billing information to determine usage volumes for personal and enterprise mobile services, and we use Sampled NetFlow data from the routers used to accommodate these services to determine the ports used.

Because traffic trends differ between weekdays and weekends, we analyze traffic in one-week chunks. In this report, we look at data for the week of May 28 through June 3, 2018, and compare those data with data for the week of May 29 through June 4, 2017, which we analyzed in the previous edition of this report.

Results are aggregated by subscription for broadband traffic, and by phone number for mobile traffic as some subscriptions cover multiple phone numbers. The usage volume for each broadband user was obtained by matching the IP

[*1] Kenjiro Cho. Broadband Traffic Report: Traffic Growth Slows to a Degree. Internet Infrastructure Review. Vol.36. pp4-9. August 2017.

[*2] Kenjiro Cho. Broadband Traffic Report: Traffic Growth is Accelerating. Internet Infrastructure Review. Vol.32. pp28-33. August 2016.

[*3] Kenjiro Cho. Broadband Traffic Report: Comparing Broadband and Mobile Traffic. Internet Infrastructure Review. Vol.28. pp28-33. August 2015.

[*4] Kenjiro Cho. Broadband Traffic Report: Traffic Volumes Rise Steadily Over the Past Year, and HTTPS Use Expands. Internet Infrastructure Review. Vol.24. pp28-33. August 2014.

[*5] Kenjiro Cho. Broadband Traffic Report: The Impact of Criminalization of Illegal Downloads was Limited. Internet Infrastructure Review. Vol.20. pp32-37. August 2013.

[*6] Kenjiro Cho. Broadband Traffic Report: Traffic Trends over the Past Year. Internet Infrastructure Review. Vol.16. pp33-37. August 2012.

[*7] Kenjiro Cho. Broadband Traffic Report: Examining the Impact of the Earthquake on Traffic on a Macro Level. Internet Infrastructure Review. Vol.12. pp25-30. August 2011.

[*8] Kenjiro Cho. Broadband Traffic Report: Traffic Shifting away from P2P File Sharing to Web Services. Vol.8. pp25-30. August 2010.

[*9] Kenjiro Cho. Broadband Traffic Report: Increasing Traffic for General Users. Internet Infrastructure Review. Vol.4. pp18-23. August 2009.

[*10] Akimichi Ogawa. Appendix A.3 "IPv6 PPPoE and IPv6 IPoE" in Professional IPv6 (in Japanese). Lambda Note. July 2018.

address assigned to users with the IP addresses observed. We gathered statistical information by sampling packets using NetFlow. Sampling rates were set between 1/8,192 and 1/16,382, taking into account router performance and load. We estimated overall usage volumes by multiplying observed volumes with the reciprocal of the sampling rate.

IIJ provides both fiber-optic and DSL broadband services, but fiber-optic access now accounts for the vast majority of use. Of users observed in 2018, 97% were using fiber-optic connections and accounted for 99% of overall broadband traffic volume.

1.3 Users' Daily Usage

First, we examine daily usage volumes for broadband and mobile users from several angles. Daily usage indicates the average daily usage calculated from a week's worth of data for each user.

Figure 2 and Figure 3 show the average daily usage distributions (probability density functions) for broadband and mobile users. Each compares data for 2017 and 2018 split into IN (upload) and OUT (download), with user traffic volume plotted along the X-axis and user frequency along the Y-axis. The X-axis shows volumes between 10KB (10^4) and 100GB (10^{11}) using a logarithmic scale. Most users fall within the 100GB (10^{11}) range, with a few exceptions.

The IN and OUT broadband traffic distributions (Figure 2) are close to a log-normal distribution, which looks like a normal distribution on a semi-log plot. A linear plot would show a long-tailed distribution, with the peak close to the left and a slow gradual decrease towards the right. The OUT distribution is further to the right than the IN distribution, indicating that download volume is more than an order of magnitude larger than upload volume. The peaks of both the IN and OUT distributions for 2017 are slightly further to the right than the peaks of the 2016 distributions, indicating that overall user traffic volumes are increasing.

The peak of the OUT distribution, which appears toward the right in the plot, has steadily been moving rightwards over the past few years, but heavy-user usage levels have not increased much, and as a result, the distribution is becoming less symmetric. The IN distribution on the left, meanwhile, is generally symmetric and closer to a log-normal distribution.

The data for mobile traffic (Figure 3) indicate that usage volumes are significantly lower than for broadband. And limits on mobile data usage mean that heavy users, which fall on the right-hand side of the distribution, account for only a small proportion of the total, so the distribution is asymmetric. There are also no extremely heavy users. The variability in each user's daily usage volume is higher for mobile than for broadband owing to those users who only using mobile data when out of the home/office as well as the limits on mobile data. Hence, the daily average for a week's worth of data shows less variability between users than the data for individual days. Plotting the distributions for individual days in the same way results in slightly lower peaks and correspondingly higher tails on both sides, but the basic shape and modal values of the distribution remain largely unchanged.

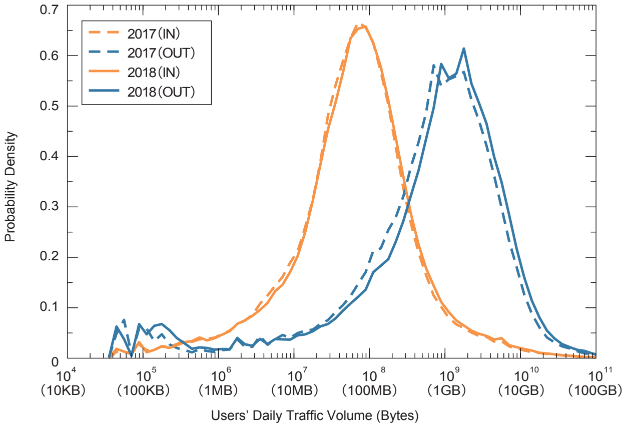


Figure 2: Daily Broadband User Traffic Volume Distribution Comparison of 2017 and 2018

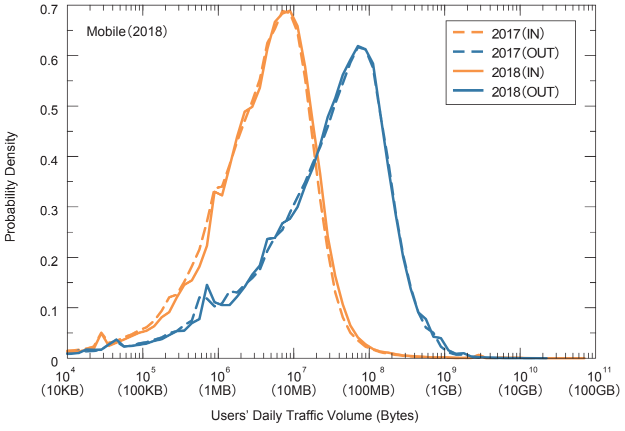


Figure 3: Daily Mobile User Traffic Volume Distribution Comparison of 2017 and 2018

Table 1 shows trends in the mean and median daily traffic values for broadband users as well as the mode (the most frequent value, which represents the peak of the distribution). The peak was slightly off from the center of the distribution, so the distribution was adjusted to bring the mode toward the center.

Comparing the values for 2017 and 2018, the IN mode was unchanged at 79MB, while the OUT mode rose from 1,260MB to 1,413MB, translating into growth factors of 1 and 1.1, respectively. Meanwhile, because the means are influenced by heavy users (on the right-hand side of the distribution), they were significantly higher than the corresponding modes, with the IN mean being 582MB and the OUT mean being 3,139MB in 2018. The 2017 means were 520MB and 2,624MB, respectively. For mobile traffic (Table 2), the mean and modal values are close owing to the lack of heavy users. In 2018, the IN mode was 7MB and the OUT mode was 79MB, while the means were IN 17.0MB and OUT 81.9MB. The modes for both IN and OUT traffic were identical to those for the previous year. The means increased despite there being very little change in the medians and modes, which indicates an increase in heavy users, particularly for IN traffic.

Figure 4 and Figure 5 plot per-user IN/OUT usage volumes for random samples of 5,000 users. The X-axis shows OUT (download volume) and the Y-axis shows IN (upload

Table 1: Trends in Mean and Mode of Broadband Users' Daily Traffic Volume

	IN (MB/day)			OUT(MB/day)		
Year	Mean	Median	Mode	Mean	Median	Mode
2005	430	3	3.5	447	30	32
2007	433	5	4	712	58	66
2008	483	6	5	797	73	94
2009	556	7	6	971	88	114
2010	469	8	7	910	108	145
2011	432	9	8.5	1,001	142	223
2012	410	12	14	1,026	173	282
2013	397	14	18	1,038	203	355
2014	437	22	28	1,287	301	447
2015	467	33	40	1,621	430	708
2016	475	48	56	2,081	697	1,000
2017	520	63	79	2,624	835	1,260
2018	582	67	79	3,139	1021	1,413

Table 2: Trends in Mean and Mode of Mobile Users' Daily Traffic Volume

	IN (MB/day)			OUT(MB/day)		
Year	Mean	Median	Mode	Mean	Median	Mode
2015	6.0	2.7	5.5	46.6	19	40
2016	7.8	3.6	7	63.0	27	63
2017	12.0	4.3	7	77.4	35	79
2018	17.0	4.7	7	81.9	36	79

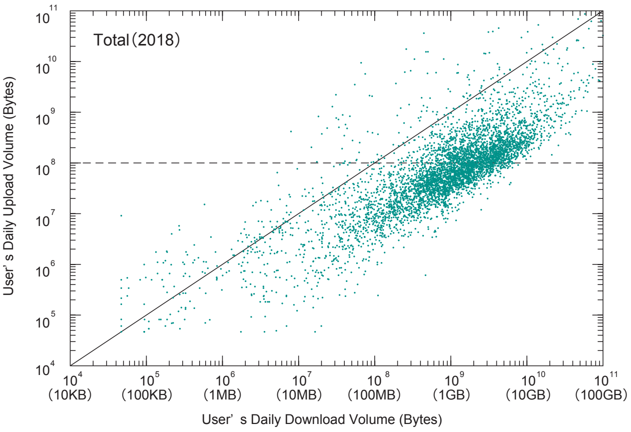


Figure 4: IN/OUT Usage for Each Broadband User

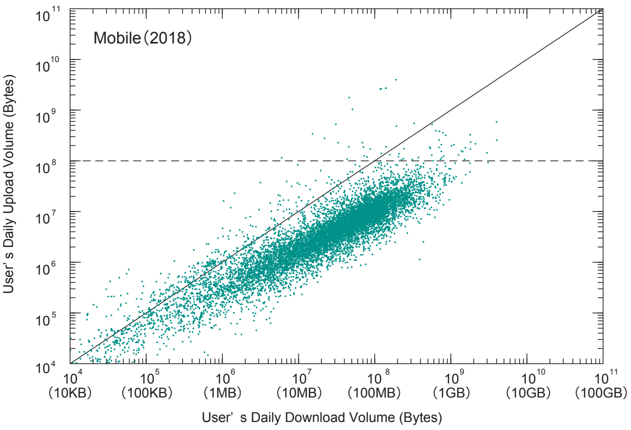


Figure 5: IN/OUT Usage for Each Mobile User

volume), with both using a logarithmic scale. Users with identical IN/OUT values fall on the diagonal.

The cluster spread out below and parallel to the diagonal in each of these plots represents typical users with download volumes an order of magnitude higher than upload volumes. For broadband traffic, there was previously a clearly recognizable cluster of heavy users spread out thinly about the upper right of the diagonal, but this is now no longer discernible. Variability between users in terms of usage levels and IN/OUT ratios is wide, indicating that there is a diverse range of usage styles. Almost no difference can be discerned when these plots are compared with those for 2017.

For mobile traffic, the pattern of OUT being an order of magnitude larger also applies, but usage volumes are lower than for broadband, and there is less variability between IN and OUT. The slope of the mobile cluster is also less steep than the diagonal, indicating that download ratios tend to be higher at higher usage levels.

Figure 6 and Figure 7 show the complementary cumulative distribution of users' daily traffic volume. On these log-log plots, the Y-axis values represent the proportion of users with daily usage levels greater than the corresponding X-axis values. These plots are an effective way of examining the distribution of heavy users. The linear-like decline toward the right-hand side of the plots indicates that the distributions are long-tailed and close to a power-law distribution. Heavy users appear to be distributed statistically and do not appear to constitute a separate, special class of user. On mobile, heavy users appear to be distributed according to a power-law for OUT traffic, but the linear-like slope for IN traffic is more out of shape than it was last year, with a larger proportion of users uploading large volumes of data.

Traffic is heavily skewed across users, such that a small proportion of users accounts for the majority of overall traffic volume. For example, the top 10% of broadband users account for 60% of total OUT and 86% of total IN traffic, while the top 1% of users account for 25% of OUT and 59% of IN traffic. As the proportion of heavy users has declined over the past few years, the skew has also decreased, albeit only slightly. As for mobile, the top 10% of users account for 50% of OUT and 70% of IN traffic, while the top 1% account for 15% of OUT and 52% of IN traffic. The proportion of heavy users has steadily been increasing over the past few years.

1.4 Usage by Port

Next, we look at a breakdown of traffic and examine usage levels by port. Recently, it has become difficult to identify applications by port number. Many P2P applications use dynamic ports on both ends, and a large number of client/server applications use port 80, which is assigned to HTTP, to avoid firewalls. Hence, generally speaking, when both parties are using a dynamic port numbered 1024 or higher, the traffic is likely to be from a P2P application, and when one of the parties is using a well-known port lower than 1024, the traffic is likely to be from a client/server

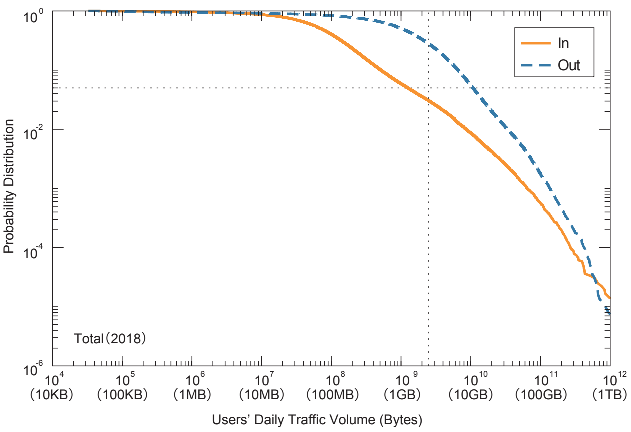


Figure 6: Complementary Cumulative Distribution of Broadband Users' Daily Traffic Volume

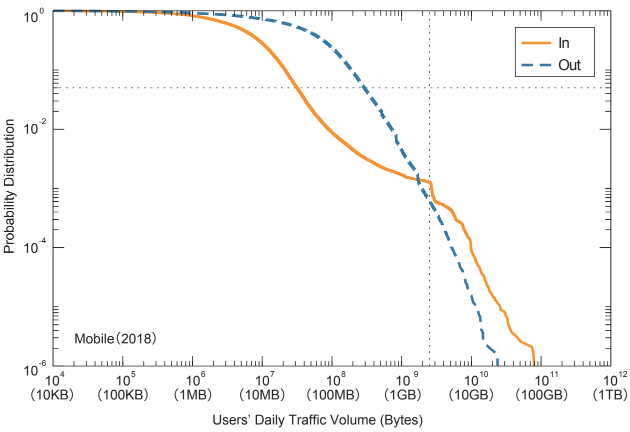


Figure 7: Complementary Cumulative Distribution of Mobile Users' Daily Traffic Volume

application. In light of this, we take the lower of the source and destination port numbers when breaking down TCP and UDP usage volumes by port.

Table 3 shows the percentage breakdown of broadband users' usage by port over the past four years. In 2018, 79% of all traffic was over TCP connections. The proportion of traffic over port 443 (HTTPS) had continued to increase up until the previous edition of this report, but it fell from 2017's 43% to 41% here. The proportion of traffic over port 80 (HTTP) also fell from 2017's 28% to 27% here, while the figure for UDP port 443, which is used by Google's QUIC protocol, rose from 11% to 16%. These figures demonstrate that the ongoing transition from HTTP to HTTPS is now turning toward QUIC. TCP dynamic port traffic, which has been on the decline, fell from 11% in 2017 to 10% in 2018. The proportion accounted for by individual dynamic port numbers is tiny, with port 1935, used by Flash Player, accounting for the largest share at around 0.7%, and the remaining port numbers accounting for less than 0.3%. As for non-TCP traffic, almost all of the traffic over ports other than UDP port 443 is VPN related.

Table 4 shows the percentage breakdown by port for mobile users. HTTPS accounts for a greater proportion of traffic here than with broadband, but the figures are close to those for broadband on the whole, suggesting that mobile users use applications in a manner similar to broadband users.

Figure 8 compares overall broadband traffic for key port categories across the course of the week from which observations were drawn in 2017 and 2018. We break the data into four port buckets: TCP ports 80 and 443, dynamic ports (1024 and up), and UDP port 443. In this edition, we take out the "well-known ports" bucket, since usage has dwindled, and add in UDP port 443 instead. The data are normalized so that peak overall traffic volume on the plot is 1. The overall peak is between 19:00 and 23:00 hours, with the peak for

Table 3: Broadband Users' Usage by Port

year	2015	2016	2017	2018
protocol port	(%)	(%)	(%)	(%)
TCP	**80.8**	**82.8**	**83.9**	**78.5**
(< 1024)	63.3	69.1	72.9	68.5
443(https)	23.3	30.5	43.3	40.7
80(http)	37.9	37.1	28.4	26.5
182	0.4	0.3	0.3	0.3
993(imaps)	0.1	0.1	0.2	0.2
22(ssh)	0.2	0.2	0.1	0.1
(>= 1024)	17.5	13.7	11.0	10.0
1935(rtmp)	1.8	1.5	1.1	0.7
8080	0.3	0.2	0.3	0.3
UDP	**11.4**	**11.1**	**10.5**	**16.4**
443(https)	0.9	2.4	3.8	10.0
4500(nat-t)	0.2	0.2	0.2	0.2
ESP	**7.4**	**5.8**	**5.1**	**4.8**
IP-ENCAP	**0.2**	**0.2**	**0.3**	**0.2**
GRE	**0.2**	**0.1**	**0.1**	**0.1**
ICMP	**0.0**	**0.0**	**0.0**	**0.0**

Table 4: Mobile Users' Usage by Port

year	2015	2016	2017	2018
protocol port	(%)	(%)	(%)	(%)
TCP	**93.8**	**94.4**	**84.4**	**76.6**
443(https)	37.4	43.7	53.0	52.8
80(http)	52.5	46.8	27.0	16.7
31000	0.0	0.2	1.8	2.9
993(imaps)	0.5	0.5	0.4	0.3
1935(rtmp)	0.5	0.3	0.2	0.1
UDP	**5.2**	**5.0**	**11.4**	**19.4**
443(https)	1.0	1.5	7.5	10.6
4500(nat-t)	0.3	0.2	0.2	4.5
12222	0.0	0.1	0.1	2.3
53(dns)	0.1	0.2	0.1	0.1
ESP	**0.7**	**0.4**	**0.4**	**3.9**
GRE	**0.3**	**0.1**	**0.1**	**0.1**
ICMP	**0.0**	**0.0**	**0.0**	**0.0**

port 443 coming just slightly earlier than the peak for port 80. Traffic increases during the daytime on Saturday and Sunday, reflecting household Internet usage times.

Figure 9 shows the trend for TCP ports 80 and 443 and UDP port 443, which account for the bulk of mobile traffic. When compared with broadband, we note that mobile traffic levels remain high throughout the day, from morning through night. The plot shows that usage times differ from those for broadband, with three separate mobile traffic peaks occurring on weekdays: morning commute, lunch break, and evening from 17:00 to 22:00 hours.

1.5 Conclusion

One identifiable trend in broadband traffic over the past year is that growth slowed somewhat in the latter half of last year but picked up again and returned to its upward trajectory in 2018. Over the past year, download volumes climbed 20% and upload volumes rose 12%, but growth in downloads has been slowing for two years in a row now.

Although the mobile traffic growth rate has fallen slightly, mobile traffic has still grown substantially over the past four years. Differences in comparison with broadband include the paucity of heavy mobile users and notably higher levels of mobile usage on weekdays during commute and lunch-break hours.

The use of HTTPS has expanded greatly since about four years ago, with TCP and UDP port 443 traffic combined accounting for 51% of broadband and 63% of mobile traffic. Given the increasing pressure to transition to HTTPS recently, with web browsers displaying messages saying the HTTP is unsafe and HTTP-only sites being pushed down the search engine rankings, we expect the decline in HTTP traffic to continue ahead.

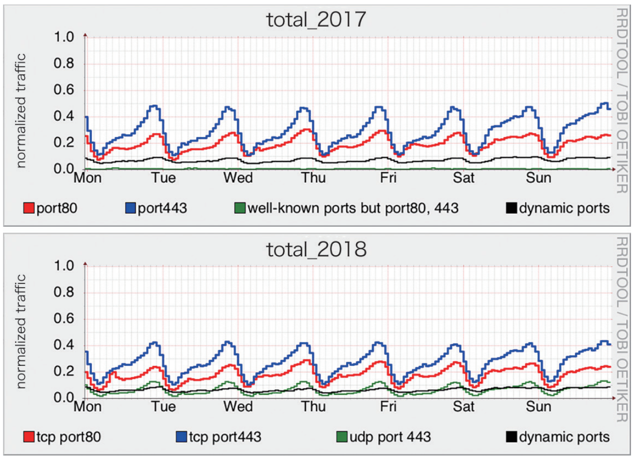


Figure 8: Broadband Users' TCP Port Usage Over a Week 2017 (top) and 2018 (bottom)

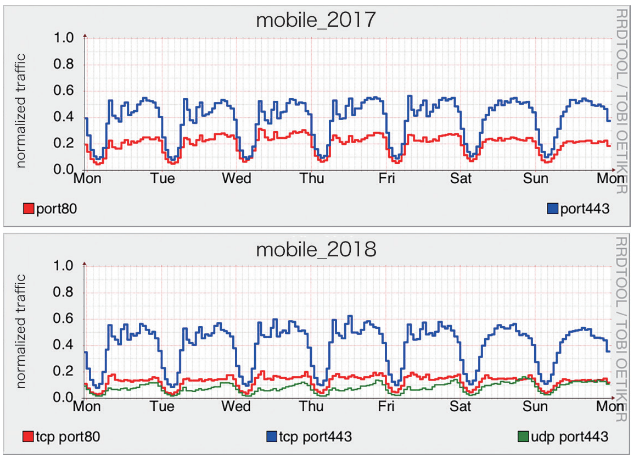


Figure 9: Mobile Users' TCP Port Usage Over a Week 2017 (top) and 2018 (bottom)

Kenjiro Cho
Research Director, Research Laboratory, IIJ Innovation Institute Inc.

Recommending Security-related Documents

2.1 Information Handled by Security Teams

When running security teams, such as SOCs or CSIRTs, you inevitably have to deal with reams of information on a daily basis. The term "information" is a single word, but the concept is broad—the nature of information and how it is handled varies. In many organizations, personnel responsible for security collect, compile, and create the information needed to fulfil their respective roles.

From a systems development perspective, this information falls into two broad categories: structured information that is amenable to automated processing, and unstructured information.

Examples of structured information that can be used by security teams include IP address blacklists, TCP/IP port databases, and SCAPs formulated to automate security operations. Semantic, structured information is amenable to automated processing and is thus widely used in systems that support security.

Unstructured information, meanwhile, is also necessary in security operations but does not lend itself well to automated processing. It includes, for example, documents written in natural languages and images. Even in support systems, it has been difficult to do anything with this sort of information other than presenting it in document form as part of reference information. Unstructured documents and images are often simply accumulated within reports and the like created at the time the information was relevant, meaning that any subsequent use of that information comes down to manual effort.

How then should unstructured information be pulled up when necessary and referenced when relevant during, for example, the performance of security tasks?

2.2 Dealing with Unstructured Information

A number of technologies aim to solve this sort of problem. The first that comes to mind is full-text search systems. Anyone can appreciate the convenience to be had in collecting and storing documents that have been created in a full-text search system so that users can find information via keyword searches when required.

Another type of system is one that presents information of potential importance to the user without the user having to search for anything. These systems include those that provide product recommendations on shopping sites and list related articles on news sites. These types of systems are called recommender systems.

When looking to use these sorts of convenient, proven technologies with natural language documents that you have on hand, implementing all-text search systems was fairly easy, but the task of implementing systems that use collective knowledge, such as recommender systems, has posed difficulties. Much of the information necessary for security operations within companies is confidential, with its use being restricted to certain authorized personnel within the organization, meaning that there are few users to begin with and not enough data can be amassed to make use of collective knowledge.

With this background in mind, here we test out an approach to deciding on recommendations based solely on the information in unstructured documents and without using the user's action history. An advantage of the approach we took and describe in this article is that user actions and documents can be handled without being externally exposed. Let's look at the task of recommending documents that are related to one that the user has selected.

2.3 Natural Language Processing and Topic Models

What sort of technology is used to deal with unstructured documents written in natural languages? This sort of technology is called natural language processing (NLP), a field that has been studied for many years and comprises various component technologies. One of those is topic modelling, which uses machine learning to analyze textual data (Figure 1).

2.3.1 Topic Models

Documents come in various types. Even in the limited context of what we read every day, technical documents and news articles can be thought of as different types of documents. News articles also come in a number of types, such as those reporting on world affairs and those reporting on sports results. The type and frequency of language used may also vary according to the type of document. Moreover, any single document does not necessarily belong to only one type. A document about internet-based attacks sparked by international conflict, for example, would belong to both world affairs and information security.

In the field of topic models, the types to which a document belongs are referred to as topics, and documents are assumed to be generated in the following manner.

First, the document's topic mixture (the degree to which a document's topics are mixed) is determined according to some probability distribution. The document is then filled with words from each topic using the probability of the word's occurrence within that topic until the final document has been generated. If these topic and word probability distributions are calculated from actual textual data, they can be used to investigate what topics are currently being focused on and to categorize documents based on topics.

Methods based on this conceptual approach are collectively called topic models. Latent Dirichlet Allocation (LDA) is the archetypal model, with a range of variations having also been created and studied.

We can expect the topic distributions of closely related documents to be close to one another, so we will use this observation and take the approach of finding documents that have topic distributions close to that of the document the user has selected and displaying those as the recommendations. Here, we use LDA to compute the topic distributions.

2.3.2 Preprocessing with Natural Language Processing Techniques

You cannot simply pass a document list into the LDA algorithm. It also needs the frequencies of the words that appear in the text. In other words, documents require some preprocessing to make them suitable for the algorithm. At the same time, we also pare information considered unnecessary with the aim of both enhancing accuracy when generating the LDA model and to reduce the volume of data required.

The following preprocessing steps are needed.

1. Extract the body text from document data
   This entails removing parts of the document data that are not part of the main body text.
2. Split the text into words (tokenization)
   With English text, this can largely be accomplished by splitting text strings on spaces and line breaks. But a number of points also need to be addressed, such as line-breaking hyphens in text with line-length limitations. If you want to treat inflectional forms of a word as the same word[*1], NLP techniques called stemming

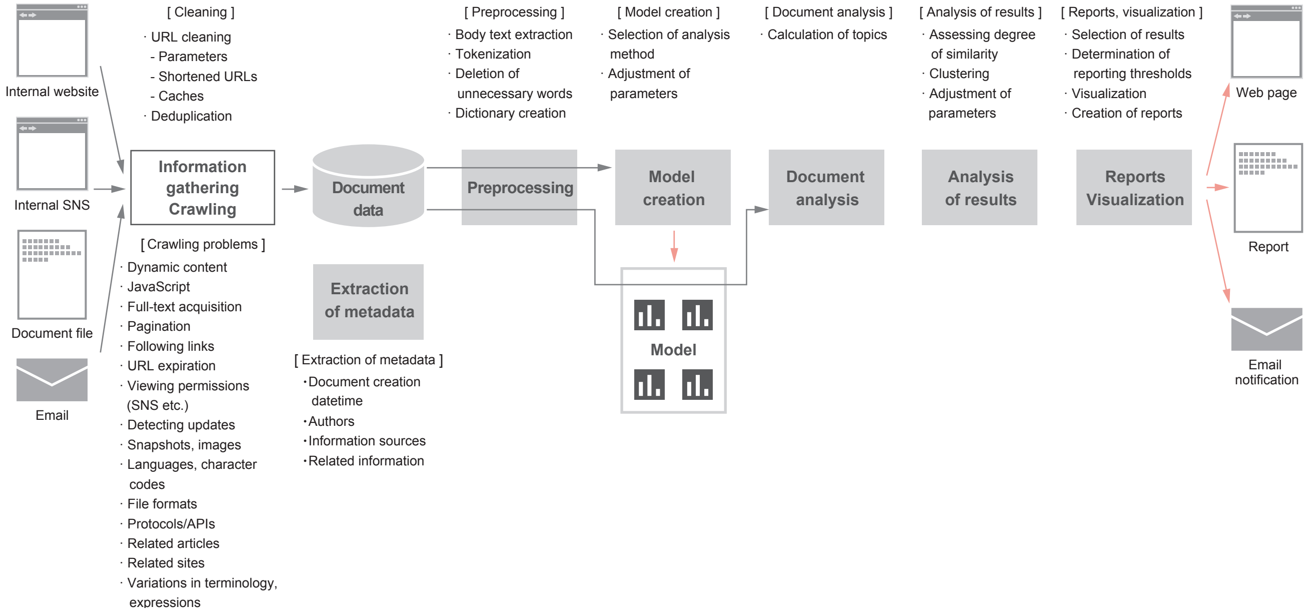


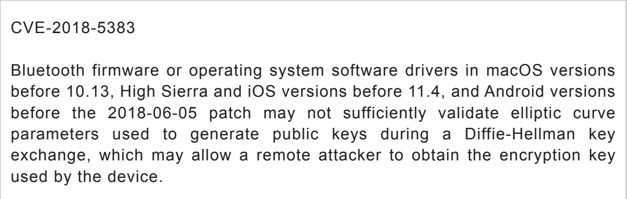
Figure 1: Overview of Natural Language Processing Using Topic Models

[*1] E.g., word-words and write-wrote-written

and lemmatization are available. Japanese text does not contain any clear word barriers, so the tokenization process involves using morphological analysis.

3. Delete unnecessary words
   Words such as "a", "an", and "the", for example, appear frequently in English text but have little to do with the overall meaning. Having the dataset filled with such words, which do not really seem relevant to the objective, is unlikely to improve the accuracy of analysis, so the usual procedure is to delete these words during preprocessing.
   One approach is to use a predefined list of stop words to delete. Other approaches include deleting words that appear frequently in the input text, and also deleting sparse words that appear only infrequently. The task of deciding which words it will be effective to delete is one of trial and error based on your objective and past examples.
4. Create a word frequency table for each document
   This involves counting the number of times words appear in the document. The word frequency table produced is called a bag of words. A separate frequency table is created for each individual document. The list of frequency tables is what you actually pass into the LDA algorithm.

As you will notice from this description, the bag of words does not reflect the order in which words appear in the documents, which means that LDA does not take word order or context into account.



CVE-2018-5383

Bluetooth firmware or operating system software drivers in macOS versions before 10.13, High Sierra and iOS versions before 11.4, and Android versions before the 2018-06-05 patch may not sufficiently validate elliptic curve parameters used to generate public keys during a Diffie-Hellman key exchange, which may allow a remote attacker to obtain the encryption key used by the device.

Figure 2: Example of a text file named by CVE ID and containing the vulnerability summary

2.4 Prototyping

We created prototypes to validate these technologies. Here, we use vulnerability summaries from the CVE[*2] database of vulnerabilities published by MITRE[*3], a US-based not-for-profit organization. Many natural language processing technologies for English text are available as part of libraries, so much of the process can be accomplished simply by using those libraries.

First, let's prepare the source data. In our prototypes, we use the 7,692 CVE vulnerabilities released to date in 2018. We download the CVE data from the NVD Data Feeds[*4] page provided by the NIST in the United States. A beta release of the data in JSON format is available, but we use the XML feed on this occasion, as we have in the past. The dataset includes fields that are easily handled by computers—including CVE ID, publication datetime, datetime of last modification, and links to reference information—but we extract only the unstructured natural language description (found in the vuln:summary tag) for each vulnerability and save these in separate files named according to the CVE IDs (Figure 2). We now have the source text files ready.

2.4.1 Text Preprocessing

We use the Python libraries gensim[*5] and nltk[*6] to perform the processing steps required for creating the LDA model.

First, we perform the necessary preprocessing on the source documents to enable us to create the LDA model. Although not present in the CVE data we used in our prototypes, preprocessing steps commonly performed on English text lifted from websites include:

- Removing HTML tags and processing special HTML characters
- Joining words split by hyphens at line ends

Next, we tokenize, lemmatize, and remove stop words to create word data from the documents. Although we simply pass a string into the lemmatize function, internally the function performs a lot of complicated processing for us, including extracting

[*2] CVE (https://cve.mitre.org/).

[*3] MITRE (https://www.mitre.org/).

[*4] NVD Data Feeds (https://nvd.nist.gov/vuln/data-feeds).

[*5] gensim (https://radimrehurek.com/gensim/).

[*6] nltk (http://www.nltk.org/).

only the nouns, verbs, adjectives, and adverbs and converting inflectional forms of each word to a common base form.

We perform the above steps on each of the documents, and create a list containing the extracted word data.

```
def normalize(txt):
   # De-hyphenation of words across a line-break
   txt = re.sub(r'-\n', '', txt)
   # Concatenate lines
   txt = re.sub(r'\n', ' ', txt)
   # Tokenization and lemmatization
   tokens = [ re.sub(r'/[A-Z]+$', '', x.decode('utf-8'))
                  for x in gensim.utils.lemmatize(txt) ]
   # Remove stop-words
   stopwords = nltk.corpus.stopwords.words('english')
   tokens = [ token for token in tokens if token not in stopwords ]
   return tokens

docs = []
for path in files:
   with open(path, encoding='utf-8') as f:
      txt = ''.join(f.readlines())
   tokens = normalize(txt)
   docs.append(tokens)
```

The gensim.utils.lemmatize() function that we invoke in the normalize function here appends part-of-speech information to the end of words, so we use a regular expression to remove this.

```
[b'cve/VB',
 b'high/JJ',
 b'rate/NN',
 b'vlan/NN',
 ...
```

2.4.2 Creating the Dictionary and Bag of Words

The gensim library that we use here assigns an ID to each word. It assigns word IDs based on the list of word data that we created and then counts the words in each document. This data structure is called a dictionary.

We then use gensim to analyze and filter the content of the dictionary. Specifically, we removed:

- Words that appear in many documents
  (Example: Words that appear in 20% or more of the documents)
- Words that seldom appear
  (Example: Words that appear in only one document)

We experimented by changing the parameters and omitting this processing step altogether. We observed a noticeable decline in categorization accuracy with models that we created without removing words that appear in many documents. When it came to the removal of words that only seldom appear, we did not observe any noticeable effect to the extent that we experimented.

Using the filtered dictionary, we create a bag-of-words (BoW) vector for each document. These are vector representations of the number of times words in the dictionary appear in the document.

```
dic = gensim.corpora.Dictionary(docs)
dic.filter_extremes(no_above=0.2, no_below=1)
bow = [ dic.doc2bow(doc) for doc in docs ]
```

2.4.3 Creating the LDA Model

We create the LDA model from the dictionary and bag of words, and then save the model and the associated data that we have created in a file.

When creating an LDA model, you need to specify the number of topics. The appropriate number of topics to use apparently changes depending on the source documents as well as the volume of words that appear and the way they are distributed. Various approaches to deciding on the value exist. We experimented by changing which documents, and how many, we fed into the model. With our data, we found that we were often able to create models that did relatively well by using values in the range of 30–50, and we therefore use 50 here.

```
lda = gensim.models.ldamodel.LdaModel(bow, id2word=dic, num_topics=50)

lda.save(filename_model)
dic.save(filename_dic)
gensim.corpora.MmCorpus.serialize(filename_corpus, bow)
```

2.5 Analyzing Documents Using the Model

We now analyze the documents using this model. The results allow us to see what sort of topics each document contains.

```
results = []
for doc in docs:
   bow = lda.id2word.doc2bow(doc)
   doc_topics = lda.get_document_topics(bow)
   results.append(doc_topics)
```

2.5.1 Pulling Up Similar Documents

Using the results of this document-wise analysis of topics and calculating the cosine similarity between vectors allows us to select documents that have closely similar topic components from among the set of all documents. So we selected a number of CVE vulnerabilities that had been in the news and such recently to verify whether we could actually pull up similar vulnerabilities.

For example, when we looked at CVE-2018-8373 (Figure 3), which was fixed by a monthly Microsoft patch in August, we obtained a whole list of similar CVE vulnerabilities that also had to do with a “Scripting Engine Memory Corruption Vulnerability” (CVE-2018-0955, CVE-2018-0996, CVE-2018-1001, CVE-2018-8267, etc.).

```
[('CVE-2018-0955', 1.0),
 ('CVE-2018-0988', 1.0),
 ('CVE-2018-1001', 1.0),
 ('CVE-2018-8267', 1.0),
 ('CVE-2018-8353', 1.0),
 ('CVE-2018-8371', 1.0),
 ('CVE-2018-8373', 1.0),
 ('CVE-2018-8389', 1.0),
 ('CVE-2018-0996', 0.9999999),
 ('CVE-2018-8242', 0.9999997),
 ('CVE-2018-0839', 0.9968462),
 …
 ('CVE-2018-8385', 0.9579928),
 …
 ('CVE-2018-8372', 0.8273082),
 ('CVE-2018-8355', 0.8272891),
 …
 ('CVE-2018-8359', 0.7355896),
```

Upon reading the documents determined to be similar, we did indeed observe many results that were similar enough to say that the documents followed an almost standardized wording.

When we looked at how similar the related CVE vulnerabilities listed in the CVE-2018-8373 summary were, we observed cosine similarities in the range of 0.73–1.0. However, 178 CVE vulnerabilities fell within this range. Although we can indeed say that we are able to find similar documents, it appears we may need some fine-tuning if we are to use these similarity scores to present documents of interest to the user.

Next, let’s look at CVE-2018-3620 (Figure 4), which relates to an exploit known as Foreshadow-NG, a side-channel attack on CPUs that use speculative execution. Related entries in the list of similar CVE vulnerabilities that we calculated included CVE-2018-3646 and CVE-2018-3615. However, the list also included vulnerabilities related to a PHP chatbot program and a web application framework with similar cosine similarity scores. Using the cosine similarity of topics as the only threshold metric may not produce very good results.

```
[('CVE-2018-3620', 0.9999924),
 ('CVE-2018-3646', 0.9803927),
 ('CVE-2018-3640', 0.88989854),
 ('CVE-2018-3693', 0.8448397),
 ('CVE-2018-3615', 0.8389072),
 ('CVE-2018-5954', 0.8318751),
 ('CVE-2018-1000181', 0.80068177),
 …
```

As demonstrated, we were able to determine that it is possible, to an extent, to find CVE entries that are along the same lines as any particular CVE vulnerability of interest by using a topic model and cosine similarity scores to find CVE entries with similar summaries.

That said, we observed many cases in which the model also pulled up a bunch of documents that did not appear to be very similar at all. And in some cases, the model failed to pull up some of the documents we had hoped for.

Take CVE-2018-5390 (Figure 5), a DoS vulnerability in the Linux kernel’s TCP implementation. The documents at the top of the list of those determined to be similar did not include any entries with relevant content. The source documents included, for instance, 105 CVE entries on Linux kernel vulnerabilities, but they did not have high similarity to CVE-2018-5390.

```
[('CVE-2018-5390', 0.99944544),
 ('CVE-2018-1237', 0.81856203),
 ('CVE-2018-1240', 0.8044424),
 ('CVE-2018-1217', 0.80138516),
 …
```

Apart from where we set the cosine similarity threshold, these results can also be influenced by the parameters used

A remote code execution vulnerability exists in the way that the scripting engine handles objects in memory in Internet Explorer, aka "Scripting Engine Memory Corruption Vulnerability." This affects Internet Explorer 9, Internet Explorer 11, Internet Explorer 10. This CVE ID is unique from CVE-2018-8353, CVE-2018-8355, CVE-2018-8359, CVE-2018-8371, CVE-2018-8372, CVE-2018-8385, CVE-2018-8389, CVE-2018-8390.

Figure 3: Summary of CVE-2018-8373

Systems with microprocessors utilizing speculative execution and address translations may allow unauthorized disclosure of information residing in the L1 data cache to an attacker with local user access via a terminal page fault and a side-channel analysis.

Figure 4: Summary of CVE-2018-3620

Linux kernel versions 4.9+ can be forced to make very expensive calls to tcp_collapse_ofo_queue() and tcp_prune_ofo_queue() for every incoming packet which can lead to a denial of service.

Figure 5: Summary of CVE-2018-5390

at each stage in the model creation process. Experiments we performed beyond what we have described here showed that the results can vary substantially depending on where these parameters are set. This also gave us insight into the difficulty of dealing with topic models, inasmuch as it is difficult to adjust each of the parameters before examining the results of the model in full.

2.5.2 Improving Accuracy by Using Peripheral Information

Topic models are one effective way of quickly finding similar documents from within a collection of documents written in unstructured natural language. That said, if the output results are to be judged on the basis of whether the model is able to properly pick out documents of interest to the reader, then some fine-tuning suitable to the application at hand will be needed to improve accuracy. This is because which similar documents will be relevant to the user differs according to the circumstances.

For instance, if you want group similar articles from the daily news, then everything other than the most recent information is likely to simply get in the way. Yet if searching for documents relevant to dealing with a problem that occurs only rarely, then surely the reader would also like older information to be included in the document search results.

To deal with this diversity, some academic research on topic models is directed at adjusting the structure of models in the aim of improving accuracy for specific applications. This includes, for example, models that incorporate analysis of time-series variations in topics and models that incorporate analysis of author names.

For this exercise here targeting CVE and security-related documents, we also experimented with filtering using peripheral information. For internal memos, for example, we were able to improve the accuracy with which we found documents of relevance to the user by giving priority to documents created around the same time as the document in question. When looking at internal documents, we found that refining the results based on keywords appearing in project names and the document path was effective.

We also looked at whether refining the results in a situation-independent manner could improve accuracy. We had some good results when filtering the list based on the results of topic clustering using clustering algorithms available in the Python scikit-learn[*7] library, such as DBSCAN. With this method, however, tuning is crucial because the parameters set when clustering greatly influence the nature of the clusters, which means it may be difficult to use this approach consistently in a situation-independent manner.

As discussed here, we conducted a range of experiments focusing on NLP and topic models in an effort to explore effective ways of dealing with unstructured information. Although we were unable to produce satisfactory results with any one method alone, we did discover that, by combining multiple methods according to the objective, it may be possible to fine-tune the approach and get closer to the desired output. Based on these findings, we will continue to look at applications of these methods to situations involving the use of unstructured documents under certain conditions.

Tadaaki Nagao

Senior Engineer, Office of Emergency Response and Clearinghouse for Security Information, Advanced Security Division, IIJ
Mr. Nagao joined IIJ in April 1998. Having worked on security services development, SDN development, and other roles, he is now engaged in research on information security in general from a theoretical perspective.
He is a member of IIJ Group emergency response team IIJ-SECT.

Yasunari Momoi

Lead Engineer, Office of Emergency Response and Clearinghouse for Security Information, Advanced Security Division, IIJ
Mr. Momoi joined IIJ in January 1999. Having worked on research and development of a range of services, including security services and wireless IC tag systems, he is now engaged in research and development related to information security in general.
As a member of IIJ-SECT, he participates in the activities and running of groups such as Information Security Operation providers Group Japan (ISOG-J) and ICT-ISAC.

Kubernetes and the Cloud

3.1 Introduction

If you have been keeping on eye on the cloud computing scene, you may have noticed that news relating to containerization technology seems to appear daily. You can be certain that news containing keywords like Docker, Kubernetes[*1], and CNCF falls into this category. And as the ecosystem continues to expand, with various products based on Docker and Kubernetes being created, it is not uncommon for stories to actually be about container-related technology even if they don't appear to be at first glance.

Although it is undeniable, to an extent, that industry players, particularly the mega cloud vendors, will be sorting out the cloud industry rules for a while to come, a look at the rise of containerization technology compels me to believe that the industry rules are likely to be rewritten by upcoming trends earlier than we had thought. Perhaps events set to have an even greater impact on the IT industry than the emergence of IaaS are afoot.

At IIJ, we have also started to make wide use of this technology to, for instance, enable rapid business deployments and the efficient, high-quality operation of large-scale systems, develop highly portable software, and optimize costs via the efficient use of infrastructure. I will discuss the IKE (IIJ Container Engine for Kubernetes) system that we use internally further below, but first I would like to explain why containerization technology has been thrust so rapidly into the spotlight and what impact this technology is likely to have on cloud computing.

3.2 Docker and Kubernetes

Products and services that encompass containerization technology are springing up like mushrooms, but only two products lie at the center of all this: Docker and Kubernetes. Catching up on developments surrounding these two products is a good way to familiarize yourself with the key trends in the rapidly advancing containerization space.

The relationship between the two is slightly complicated, but in the simplest terms, Docker is a container engine that starts programs and wraps them in containers, while Kubernetes is a container orchestrator that bundles together and controls multiple container engines. In general terms, an orchestrator is a controller that coordinates between multiple interrelated systems and integrates them into a single overall system. A container orchestrator like Kubernetes bundles together the multiple host nodes started by the container engine (e.g., Docker) and controls them efficiently and autonomously as a single large resource pool. Although Docker, as a product, competes with Kubernetes in some respects, it is less confusing to to think of the relationship between them as being one of a container orchestrator and a container engine (Figure 1).

That said, these two products are not always paired with each other. Docker is already widely recognized and used for the convenience it offers as a standalone system, but Kubernetes does not seem to be used quite that extensively in production environments. This is because Kubernetes is a platform designed to control container clusters of a fairly

[*1] Kubernetes is a container orchestrator that controls container engines, such as Docker, and manages container clusters composed of multiple nodes. It was created by Google and is currently an open source software project hosted by the CNCF (Cloud Native Computing Foundation). It is said to have originally been based on an internal Google system called Borg. Kubernetes is a runtime environment for containerized applications and makes possible infrastructure-independent, portable application packaging, provisioning, and operation. Kubernetes is sometimes referred to as an OS for the cloud era and is looked to as a potential unified operations interface for multicloud and hybrid cloud environments.

To use Kubernetes, you need network drivers, storage drivers, traffic managers, and so on to match your infrastructure, but Kubernetes basically does not include implementations of these elements. And to set up a Kubernetes environment properly, portal and monitoring tools for application management and account management are also essential, but these sorts of tools also need to be found elsewhere. This is the reason for the emergence of Kubernetes distributions, which package Kubernetes together with an environment for it to run in as well as an installer and other tools. IKE (IIJ Container Engine for Kubernetes), which I discuss below, is one such Kubernetes distribution.

decent size, so there are certain hurdles to overcome even if you just want to try it out, whereas Docker can also serve as a convenience utility in your local computing environment and is therefore a simple and easy choice even for minor use cases. A lot of engineers are probably making convenient use of Docker as a tool for setting up test environments, and as a means for distributing software. Docker is already on its way to becoming an essential part of the engineer's toolkit.

Yet, of the two, it seems to be Kubernetes that is causing the bigger stir in the IT industry at present. This is because through the use of Kubernetes, containerization technology, rather than merely providing convenient utilities, is expected to significantly change the face of server-side systems.

Why is Kubernetes, which is by no means yet mature, garnering so much attention? Probably because Kubernetes originally came from Borg, which has supported Google's systems for over 10 years. Google's internal systems are almost never explained in any detail, but by revealing some of the ways in which it is used, the book Site Reliability Engineering (the SRE book, as it is commonly known) offered a glimpse into the surprising realities of Borg. No doubt this is what prompted more than a few people to take an interest in containerization technology. The revelation that Google's systems do not contain virtual machines and that all processes are essentially run as containers had a major impact on engineers, particularly those working in the cloud business. Kubernetes is sometimes referred to as the OSS version of Borg, but it is unclear how much the two actually have in common. As a relative newcomer, Kubernetes may appear to have a limited track record, but it is quite conceivable that its design incorporates best practices that have have been in use, and grown mature, at Google over a long period of time.

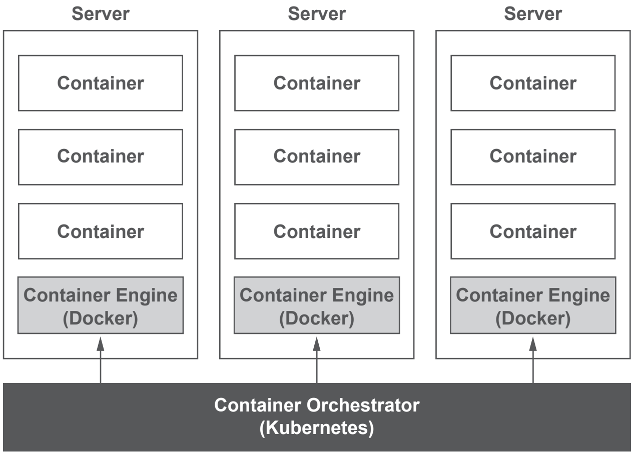

Server
Server
Server
Container
Container
Container
Container
Container
Container
Container
Container
Container
Container Engine (Docker)
Container Engine (Docker)
Container Engine (Docker)
Container Orchestrator (Kubernetes)

Figure 1: Container Engines and the Container Orchestrator

3.3 Best Practices for Harnessing IaaS

So then, how is Kubernetes set to change the face of server-side systems? It will facilitate highly portable, infrastructure-independent deployments, goes the narrative. It will offer large-scale cluster management capabilities and make dynamic use of computing resources to provide excellent scalability. The potential of Kubernetes is described in all sorts of ways, but they tend to be vague and somewhat nebulous. That is in some sense inevitable—the role of Kubernetes is akin to that of a computer OS. Try explaining what an OS does to someone with no idea what an OS is and you're likely to either find yourself telling a rambling, fragmented tale or delving into extremely technical detail.

Kubernetes is actually often referred to metaphorically as an OS for the cloud era. Drivers absorb any differences in infrastructure, the configuration of networks and storage is virtualized, and a unified interface is provided to any systems deployed on Kubernetes. Kubernetes makes it possible to design, build, and operate unified systems without relying on IaaS-specific interfaces. That said, Kubernetes is not an OS and does not provide an applications interface. The applications that run on Kubernetes are simply ordinary Linux and Windows applications.

While they are comparable in some respects, the major difference between an OS and Kubernetes is that an OS controls a single computer housed within a physical box, whereas Kubernetes manages multiple networked computers as one large resource pool. Processes managed by an OS cannot leave the physical box, but processes running on a Kubernetes cluster (i.e., containers) can be on any of the nodes that make up the cluster. So when resources are depleted, all the system has to do is increase the number of nodes and reassign containers (this happens automatically), and if a container stops running because of a fault with a node, the container can be restored simply by restarting it on a different node (this also happens automatically).

Cloud services, in many cases, are regarded as having excellent stability, free from outages. But in reality, cloud services are many and varied, and since redundancy is not built into IaaS resources, in particular, such services can stop when faults occur, and they also experience planned outages for maintenance purposes. The spread of IaaS has made it possible to procure and build system resources and to recover from hardware faults in impressively short amounts of time, but for the most part, not that much has changed in the way systems are operated. Whether virtualized or not, if you're still dealing with servers, storage, and networks, very little changes in terms of operations. IaaS is not difficult to use, but the reality is that considerable effort is required to take advantage of the utility computing benefits that IaaS can offer.

Enter Kubernetes. A characteristic of IaaS systems is that they secure only the necessary resources as and when needed, and users only incur costs for the resources used. This dovetails well with Kubernetes, which makes it easy to dynamically manage and thereby efficiently use resources and to take advantage of scalability. Not only does Kubernetes combine multiple nodes to ensure availability, it also makes it possible to automatically recover, without the need for human intervention, from all but systemwide faults by having the task of restoring faulty nodes delegated to Kubernetes (Figure 2).

An oft-heard explanation is that containers do away with the thick management layer conventionally constituted by hypervisors and virtual machines and thus allow for a thin, lightweight management layer with containerized processes running directly on a single OS. But the effect of containerization technology would be very limited if that were all there was too it. Containerization is indeed more efficient than virtual machines in many cases, but that is simply a matter of means. The real effects are only realized once multiple servers are bundled into a single large resource pool, with operations automated by delegating management of configuration information to Kubernetes. In fact, a lot of containers right now probably run on an IaaS system implemented as a virtual machine. Kubernetes can be thought of as a package imbued with best practices for making better use of IaaS technology.

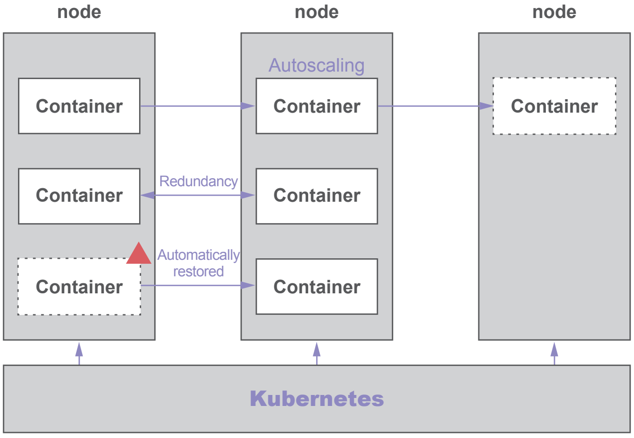


Figure 2: Kubernetes using IaaS

3.4 Realizing Hybrid Clouds with Kubernetes

While it is true that Kubernetes is a platform for better harnessing the advantages of IaaS, it's not the case that Kubernetes can only be used on IaaS. In fact, looking ahead, Kubernetes probably deserves more attention with respect to workloads for which an on-premise environment is key. We have been listening to our customers, and looking at cloud-related reports, for almost 10 years now. With the spread of cloud computing, although one can reason that a 100% on-prem setup is unrealistic, the response from the vast majority of customers has been that a 100% cloud setup would also be difficult to pull off. The fact that this is the majority opinion at a time when many engineers have become highly proficient with IaaS may offer some indication of what lies ahead.

What this means is that we need to think seriously about using hybrid clouds, which make use of both IaaS and on-prem environments as necessary. But, needless to say, this is no mean feat. Since IaaS is a closed system, it cannot be used in on-prem environments. Even if it were possible, however, operating a complex IaaS system in an on-prem environment could end up with you scratching your head and wondering what the whole point of using IaaS was in the first place. There certainly are some workloads for which an IaaS-equivalent setup in a private environment may be justified, but there are many tradeoffs to consider. However, wrapping both an IaaS and an on-prem environment with Kubernetes may be a realistic solution. Broadly, there are two challenges on the road to a hybrid cloud. These are the integrated management of IaaS and on-prem systems, and application and data portability (Figure 3).

You certainly want to keep the following within the on-prem environment: workloads and data that cannot be allocated to an IaaS system for compliance reasons, and workloads that use a fixed set of large-scale resources over an extended period with no increases or decreases in the amount of resources. And you may want to allocate all other workflows to an IaaS system. If it were clear, to an extent, which is the most appropriate from the get go, there would not be much to worry about, but it is not uncommon for the answer to the question of which is most appropriate to change over time, with, say, IaaS being preferred during a business's startup phase and on-prem being a better choice once the business has stabilized. If Kubernetes can solve the two previously mentioned challenges effectively, this market has the potential for sudden expansion.

At present, however, neither hybrid clouds that use Kubernetes nor on-prem environments that use Kubernetes can really be described as mature when compared with comparable IaaS systems. IaaS allows for software-based control of all resources via APIs, whereas on-prem environments

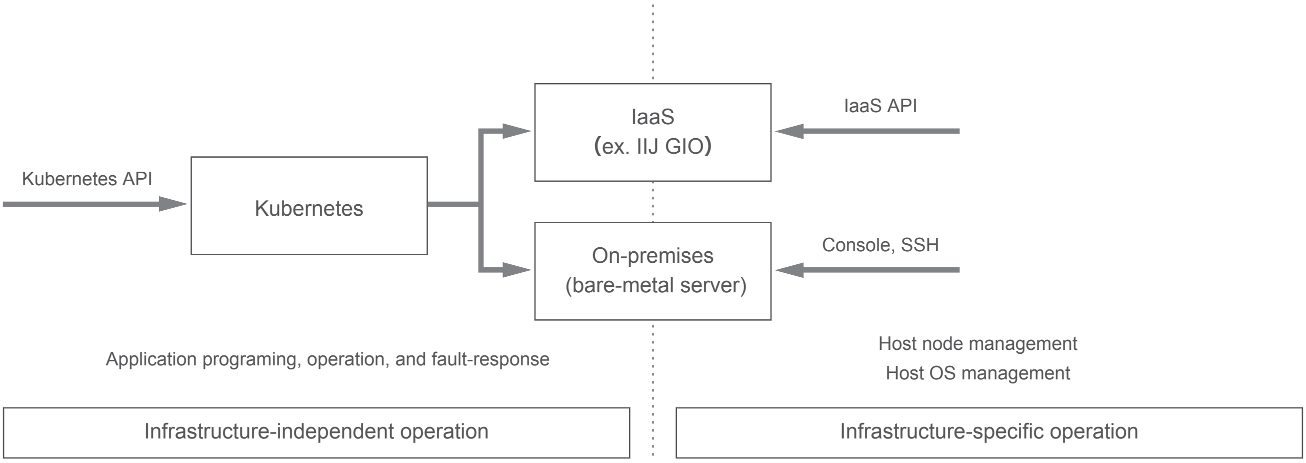


Figure 3: Kubernetes Enabling a Hybrid Cloud

do not allow integrated software control, so compromise and cooperation between both Kubernetes and devices is necessary in order to build the environment. Also, once technologies such as SDN (software-defined networking) and SDS (software-defined storage) become easier to use in on-prem environments, this is likely to spur on the spread of Kubernetes in on-prem environments.

To realize a hybrid cloud, we need a common system for managing IaaS and on-prem. Kubernetes is without doubt a strong candidate for such a system.

3.5 IKE (IIJ Container Engine for Kubernetes)

Kubernetes has the future potential to dramatically improve efficiency by fundamentally changing the design and operation of server side systems and the distribution and provisioning of applications, and at IIJ, we have also developed and begun using a container cluster system. Named IKE (IIJ Container Engine for Kubernetes), this system was created to serve as a common platform for services and as an operating environment for our internal systems.

Packages like IKE that provide a container cluster environment around Kubernetes are called Kubernetes distributions. Above, I likened Kubernetes to an OS, but it is actually like an OS kernel. An OS cannot do much with a kernel alone. It needs the tools and device drivers provided by a distribution (e.g., RedHat or Ubuntu) before it can be of any use. Similarly, simply installing Kubernetes will not get you far. At a minimum, you need network drivers and storage drivers to suit your infrastructure; for a more pleasant container cluster experience, you need management tools to control Kubernetes; and a full support environment that facilitates the monitoring of applications deployed on Kubernetes, the display of alerts, the collections of logs, and so on is also indispensable. Packages that provide the ecosystem for setting up a Kubernetes cluster along with a mechanism for installing that environment as appropriate for the specified infrastructure are called Kubernetes distributions. IKE is one such distribution.

IKE was not developed for the purpose of providing services to customers, so it is only designed to work in a somewhat limited operating environment, but it can be installed on IIJ GIO, our cloud service, as well as in on-prem environments. We also plan to make it installable on other vendors' IaaS platforms and make it possible to provide a common environment on any sort of infrastructure.

IIJ has several reasons for implementing a Kubernetes distribution, and making use of IaaS is not the only objective. Our foremost objective is to speed up business, and our second objective is to enhance operational specialization so as to handle increasingly sophisticated and complex systems. This may all sound a little abstract and leave you with the impression that our objectives are vague, but what we ultimately aim to achieve is an environment in which, for example, teams responsible for developing services can concentrate on development while operations teams can focus on operations. If we can achieve this, I think we will naturally also fulfill those objectives (Figure 4).

In any case, containerization is a fascinating area. Simply enclosing each process in a container, rather than jamming a full server environment into a single container like with a virtual machine, makes it possible to realize infrastructure-independent server-side system distributions and all-purpose operation systems, giving rise to products that influence the entire IT industry. And this is probably only the beginning. It's exciting to think that more surprising and unexpected ideas still lie ahead.

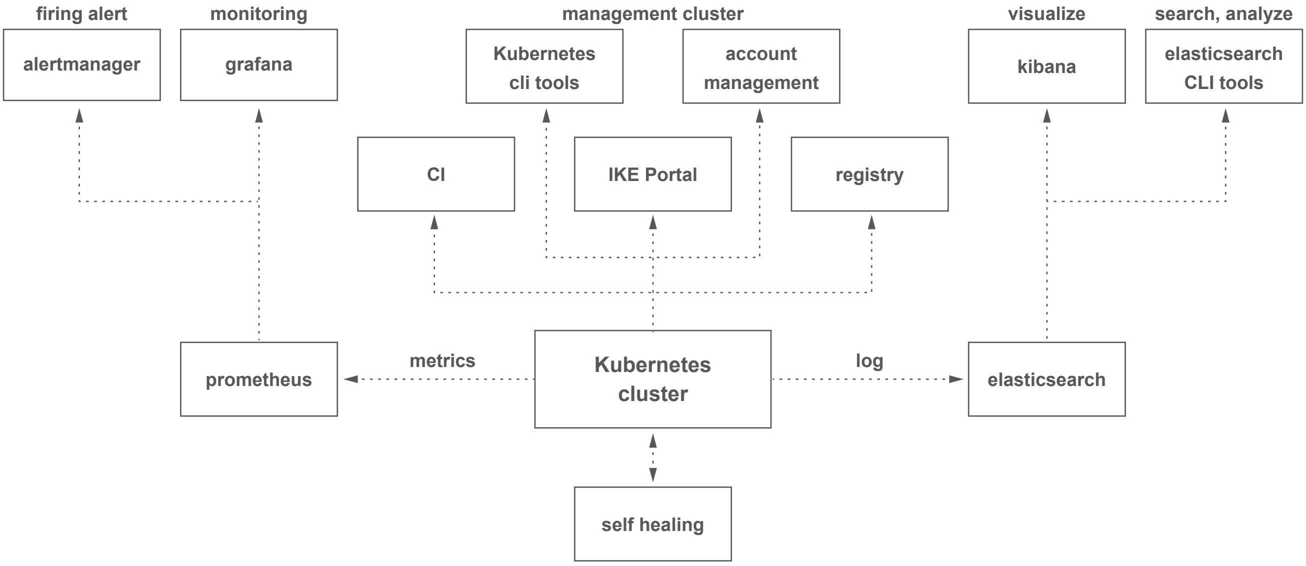


Figure 4: IKE

Keisuke Taguchi

Technology Strategy Office, Service Administration Division, IIJ.

As an engineer, I divide my working life equally between freelancing and IIJ. I started out as an applications engineer, but as soon as I encountered cloud computing, I ended up giving half of myself over to the infrastructure side of things.

I'm currently absorbed in the area of containerization technology, which I could go on and on about, so I'll stop here. Tags that apply to me: container, cloud, road bike, udon.

November 2018 Vol.40

Internet Initiative Japan

About Internet Initiative Japan Inc. (IIJ)

IIJ was established in 1992, mainly by a group of engineers who had been involved in research and development activities related to the Internet, under the concept of promoting the widespread use of the Internet in Japan.

IIJ currently operates one of the largest Internet backbones in Japan, manages Internet infrastructures, and provides comprehensive high-quality system environments (including Internet access, systems integration, and outsourcing services, etc.) to high-end business users including the government and other public offices and financial institutions.

In addition, IIJ actively shares knowledge accumulated through service development and Internet backbone operation, and is making efforts to expand the Internet used as a social infrastructure.

Internet Initiative Japan Inc.

Address: Iidabashi Grand Bloom, 2-10-2 Fujimi, Chiyoda-ku,
Tokyo 102-0071, Japan
Email: info@iij.ad.jp URL: https://www.iij.ad.jp/en/