SOC Report

1.1 Introduction

IIJ maintains the wizSafe security brand and works constantly to create an environment in which the Internet can be used safely. One of its initiatives in this regard is the regular dissemination of information on security issues in blog form via wizSafe Security Signal[*1]. IIJ also uses its position as an ISP to perform data analysis on its Data Analytics Platform[*2], which aggregates backbone traffic as well as security appliance logs, focusing its efforts on preventive measures and swift follow-up responses against the increasingly sophisticated cyberattacks out there.

Section 1.2 of this report looks back at security incidents that occurred in 2023 and provides a security summary covering key incidents of note in calendar format. Section 1.3 then discusses our latest efforts to deal with emerging challenges faced when performing data analysis and operating the Data Analytics Platform under the headings "Challenges in operating and performing analysis on the Data Analytics Platform and the deployment of dbt" and "Visualizing C&C communications with a focus on time interval variability". As part of this discussion, Section 1.3.1 describes the events that led us to deploy dbt to address the issues of data freshness and data mart creation and maintenance. And in Section 1.3.2, we share the results of our evaluation of the use of a statistical measure called the moving coefficient of variation to deal with three issues related to the evaluation of communications time interval variability.

1.2 2023 Security Summary

Tables 1 and 2 show the security incidents that the SOC focused on from among those that rose to prominence in 2023.

*1 wizSafe Security Signal (https://wizsafe.iij.ad.jp/).

*2 Internet Infrastructure Review (IIR) Vol. 38 (https://www.iij.ad.jp/en/dev/iir/038.html).

Table 1: Incident Calendar (January–June)

Month	Summary
January	Access to a local government's official website was disrupted, and it was revealed that a group calling itself Anonymous may have been to blame for the disruption (DDoS attack). A user claiming to be Anonymous posted content on Twitter (now X) hinting at the attack.
January	IIJ's SOC observed an increase in attack emails with Microsoft OneNote attachments. Multiple malware infection campaigns using OneNote files were seen from January onward.
February	Multiple organizations reported ransomware campaigns (ESXiArgs) exploiting an OpenSLP heap-overflow vulnerability (CVE-2021-21974) in VMware ESXi products. IIJ's SOC also confirmed there to be an increase in scanning on port 427, which is used by OpenSLP, around the same time.
February	A crypto asset exchange disclosed that its employees had been subject to a social engineering attack. The attacker targeted multiple employees via smishing and attempted to log into the exchange's internal network using stolen employee credentials but was blocked by multi-factor authentication (MFA). The attacker attempted to keep the attack going by calling the employees and posing as an IT staff member, but the exchange's CSIRT was able to intervene and foil the attack.
February	Ransomware group Clop claimed to have stolen data from over 130 organizations by exploiting a zero-day vulnerability in the file transfer tool GoAnywhere MFT. The vulnerability is identified as CVE-2023-0669. Thereafter, several organizations, including Japanese companies, announced that they have been impacted.
March	The JPCERT Coordination Center (JPCERT/CC) issued an alert on mailouts aiming to infect systems with Emotet, which had not been observed since November 2022. In some cases, the attached ZIP files contained a Word file that exceeded 500MB once extracted, which is likely a contrivance designed to avoid detection by security products(Note 1).
March	A foreign company that develops business communications software revealed there to be a risk of information-stealing malware infections from the official installer of software it provides. An investigating security vendor subsequently reported that the company had suffered a supply chain attack, compromising its software development environment, which led to other supply chain attacks.
March	A telecommunications carrier disclosed that customer information associated with personal internet services and video streaming services had been leaked from one of its contractors. A follow-up report revealed that the cause of the breach was not a malware infection but that a former temp employee who worked on the services at the contractor had exfiltrated the information.
March	A company that provides information communications and system integration services disclosed that an incident occurred in which a local government certificate issuing service it provides was issuing certificates to residents other than the certificate applicants. The service was temporarily suspended, but in the subsequent process of fixing the system program that caused the incident, it was found that the patch had not been applied at some local governments, so the company said it was carrying out inspections and moving quickly to apply the patch.
April	An IT company that works on public services as a contractor disclosed that servers used in solution services it provides for local government meetings had been subject to unauthorized access. The unauthorized access prompted several local governments to report a temporary suspension of Internet-based meeting live-streaming services.
May	A major automaker's business strategy firm disclosed that customer information, including vehicle data and footage captured by drive recorders, had been made externally accessible due to a cloud misconfiguration. A subsequent investigation of other cloud environments revealed new cases of some customer information having been made accessible.
May	Progress Software disclosed that the web application of its MOVEit Transfer file transfer service contained an SQL injection vulnerability (CVE-2023-34362), and that the vulnerability had already been exploited. The company initially advised customers to check for indicators of unauthorized access over "at least the past 30 days", but other security organizations subsequently disclosed that exploitation dated back to more than 30 days prior.
June	Fortinet disclosed that the SSL-VPN functionality of FortiOS and FortiProxy contained a heap-based buffer overflow vulnerability (CVE-2023-27997), and that it may have been exploited in some limited cases. Multiple security vendors reported that the attack group Volt Typhoon uses a zero-day vulnerability in Fortinet products to gain initial access, but it was revealed that no evidence of the heap-based buffer overflow vulnerability being used had been found.
June	From June, there were ongoing reports of people who had made bookings through a travel booking site receiving messages directing them to phishing sites. This was traced back to unauthorized access to the accommodation booking information management system, and it was revealed that devices used to manage the system at some accommodation facilities were infected with malware. Multiple security vendors reported that the attacker was posing as a customer and sending enquiry emails and the like to get operators to open attachments that would infect the devices with information-stealing malware.

Note 1: Alert Regarding Re-emergence of Emotet Malware Infection Activities (https://www.jpcert.or.jp/english/at/2022/at220006.html).

Table 2: Incident Calendar (July–December)

Month	Summary
July	A port transportation group announced that its terminal system was down due to a ransomware infection and that it was working to restore it. After the outage, media outlets reported that the group had received threats from an attack group going by the name Lockbit. Operations were resumed in short order, over the course of about two and a half days.
July	Microsoft disclosed that the attack group Storm-0558 had gained unauthorized access to Exchange Online and Outlook.com and was able to access emails and personal accounts of around 25 organizations, including government agencies. It revealed that the group had used a Microsoft Service Account (MSA) signing key to forge authentication tokens that enabled it to access the services. While it did not have any concrete evidence as to how the group obtained the signing key, Microsoft issued a subsequent report that described likely methods.
August	Media outlets reported that an Indonesian man had been arrested for using a phishing kit called 16shop to steal and fraudulently use other people's credit card information. This was the first case in which Japan's National Police Agency was involved in a joint cyber investigation with another country: it worked with INTERPOL and Indonesian police to arrest the suspect.
August	There were multiple incidents of login screen tampering on mobile network-compatible IoT routers used in Japan. IIJ's SOC confirmed that the affected login screens contained content protesting the discharge of treated water from nuclear power plants. An account on X thought to be involved in the tampering posted an explanation of the vulnerability exploited.
September	A security company revealed that when AI researchers published open-source AI models on GitHub, a misconfiguration of the tokens that grant access to the public data meant that 38TB of data, including researchers' computer backups, was also exposed. It also revealed that the exposed data included passwords to the AI researchers' organization's services, secret keys, and employees' internal messages.
September	It was revealed that a domain used by a remittance and payment service had been listed on a domain registrar's auction site, causing a stir, with people raising concerns of the domain potentially being acquired and misused by a third party. After speaking to the company that owned the domain, some media outlets reported that the incident was due to internal mishandling.
September	The Zero Day Initiative (ZDI), a vulnerability discovery community, disclosed a vulnerability (CVE-2023-42115) in the mail transfer agent (MTA) Exim that could allow remote code execution. The vulnerability had been reported by a security researcher in June of the previous year, and in September, ZDI informed the vendor that it intended to publish it as a zero-day advisory, and subsequently did so(Note 2).
October	Google, Cloudflare, and Amazon Web Services all announced that a large-scale DDoS attack exploiting a vulnerability in the HTTP/2 protocol (CVE-2023-44487) had been observed in August. The vulnerability exploits the HTTP/2 protocol's stream multiplexing, which enables the concurrent processing of multiple HTTP requests and responses within a single TCP connection. The attacking client sends a high volume of HEADERS frames and RST_STREAM frames to drain server resources. This is called an HTTP/2 Rapid Reset attack.
October	Cisco Systems disclosed a vulnerability (CVE-2023-20198) in the web UI feature of Cisco IOS XE. By exploiting this vulnerability, an attacker can obtain the hightest level of access without authentication and create new local users. A few days later, it disclosed another vulnerability (CVE-2023-20273) in the web UI feature that would allow local users to execute commands with root privileges. Reports of damage from attacks combining these two vulnerabilities were reported in Japan as well, and IIJ's SOC observed traffic related to this vulnerability.
October	A contact center company disclosed that a person involved in the operation and maintenance of the call center system had been exfiltrating customer information and leaking it to third parties. This breach is thought to have been ongoing for around 10 years. The root cause was that it was possible to download customer information on operation and maintenance devices and write it to external storage media, and these operations were not detectable via timely detection measures or routine log checks.
October	A company that provides identity management and authentication services disclosed that its support case management system had been subject to unauthorized access. It initially said that only files uploaded by some customers had been affected, but it subsequently revealed that all users had been affected. An investigation into the account used for unauthorized access revealed that an employee had signed in to a personal Google account via a device managed by the company, and the account information had been synced to the employee's personal device, suggesting that the account may have been stolen because the personal device was compromised.
November	It was disclosed that ownCloud, open-source software for building online storage, contained a vulnerability (CVE-2023-49103) that allows confidential information and settings in containerized environments to be leaked. Some days after this was revealed, several organizations reported that the vulnerability was being exploited.
November	Akamai's SIRT disclosed that a Mirai variant called InfectedSlurs was using a zero-day vulnerability in attacks designed to build out DDoS botnets. A follow-up report revealed that one of the products with this vulnerability is a wall-jack-compatible wireless LAN router sold in Japan, and that it is possible to exploit the authentication details in this device's factory default settings. IIJ has observed DDoS attacks emanating from such botnets as ongoing since November.
December	Microsoft and Arkose Labs announced they had seized infrastructure used by Storm-1152, a group that sells fraudulent Microsoft accounts and CAPTCHA bypass tools, identified the ringleaders, and filed criminal charges with law enforcement. It was also revealed that Storm-1152 had sold around 750 million fraudulent Microsoft accounts, and was providing tutorial videos and support chat services for the tools it was selling.
December	Security consulting firm SEC Consult disclosed an SMTP implementation vulnerability called SMTP smuggling. The SMTP protocol defines an end-of-data sequence, but many products will recognize a sequence as signaling the end of data even when an email message deviates from the standard sequence. So by adding a second email message in the data following the sequence, an attacker can, for example, bypass sender spoofing checks that would normally occur per the SPF sender authentication method.

Note 2: (0Day) Exim AUTH Out-Of-Bounds Write Remote Code Execution Vulnerability (https://www.zerodayinitiative.com/advisories/ZDI-23-1469/).

1.3 Security Topics

This section discusses the IIJ SOC's efforts to tackle some of the challenges it faces.

1.3.1 Challenges in operating and performing analysis on the Data Analytics Platform and the deployment of dbt

At IIJ, we operate a data platform that we call our Data Analytics Platform, with the objective being to implement appropriate preventive measures and post-incident responses to cybersecurity threats. For additional details on these initiatives, refer also to the SOC Report in IIR Vol. 38 (https://www.iij.ad.jp/en/dev/iir/038.html), published March 2018. We continue to operate the Data Analytics Platform, continuously rolling out software updates and expanding data sources. Here, we describe some challenges that have emerged through the operation of the platform and our efforts to introduce open-source software called dbt (data build tool) to address them.

Before delving into the challenges, an explanation of the underlying data platform operations and architecture is in order. Note that the discussion here is not limited to the Data Analytics Platform. Firstly, in general, work on the data platform is performed by team members serving in the following two types of roles.

- Data engineer
- Data analyst

While more fine-grained distinctions may be used at times, here we will say that a data engineer is someone who creates the data platform, and a data analyst is someone who uses the data platform. With IIJ's Data Analytics Platform too, we break the overall staff up into a team responsible for development and a team responsible for analysis, and those teams collaborate with each other.

The current mainstream approach is to manage the data stored on data platforms using the following layered structure, and we manage data on our Data Analytics Platform in a similar manner (Figure 1).

- Data lake layer
- Data warehouse layer
- Data mart layer

The data lake layer stores data collected from or transferred by data sources, which generate the data, and to the extent possible, keeps it in the form in which it was received. The data warehouse layer transforms data from the data lake layer into structured data to facilitate analysis. The data mart layer transform data from the data warehouse layer into specific forms tailored to specific use cases.

The data mart layer will be relevant to the challenges discussed below, so an example from the domain of security is in order. Say that data collected from an IPS/IDS on a particular data platform is imported into the

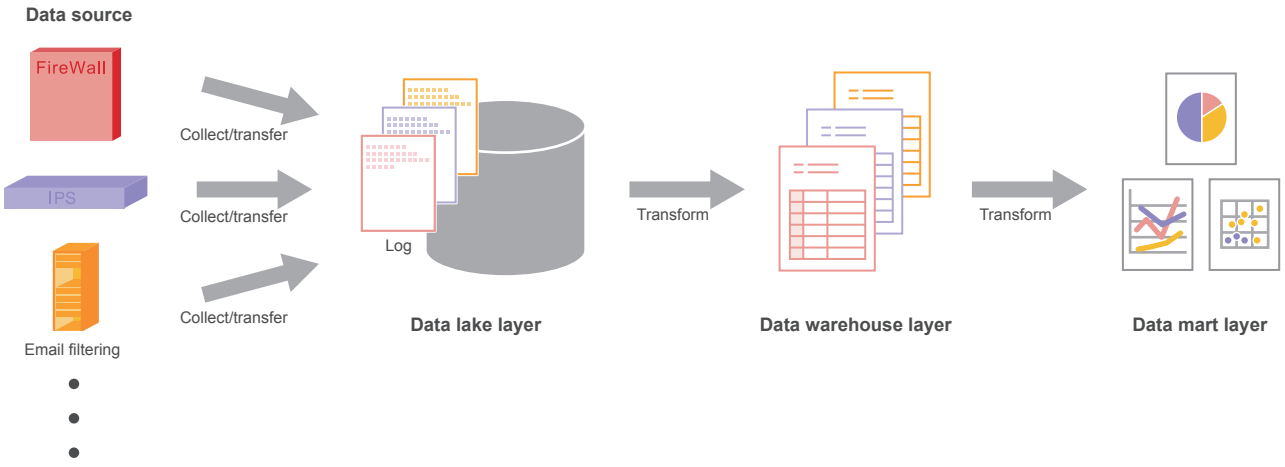

Figure 1: Illustration of How Data is Managed in a Layered Structure on a Data Platform

data warehouse layer. And let's say that a particular data point (record) contains information about the signature of and time at which a threat was detected. If you want to look at daily trends based on this data, you might consider producing daily counts of the type and number of detected signatures. In this scenario, data counts that have been generated in advance, rather than created ad hoc by a data analyst, constitute a data mart. For important information that is frequently looked up, you might also consider adding visualization features using a dashboard or the like.

Next, we look at two challenges that have emerged in the course of operating and performing analysis on the Data Analytics Platform. The first relates to data quality. For data analysts to perform appropriate analyses, data on the platform must remain in a healthy state. Yet there is no such thing as a perfect system, and on our Data Analytics Platform, we experienced the following types of events.

- Decline in freshness of certain data
- Unexpected circumstances befalling certain data

The concept of data freshness concerns whether data is imported into the data platform in a timely manner without any delays. A decline in data freshness occurs when data is imported into the data platform slower than expected or when the data flow temporarily stops. A decline in data freshness may mean that data analysts are unable to perform timely analysis[*3].

Saying that unexpected circumstances can befall data is somewhat vague, but what this means is that, from a data analyst's perspective, something appears to be amiss with the data. Easy-to-visualize examples include values that are supposed to be unique appearing multiple times and unexpected values popping up in the data[*4]. Overlooking these sorts of phenomena can affect the reliability of analysis results.

The second challenge relates to creating and maintaining data marts. Data marts are crucial when it comes to routine tasks that are performed over and over again. This is because the presence or absence of a data mart can greatly impact on work efficiency.

On the other hand, when creating data marts on the Data Analytics Platform, there was a tendency for engineers to develop them individually. With this style of development, significant resources are spent on everything from development through to maintenance. And the lead time required to actually put the data mart into use tends to be long.

To address this challenge on our Data Analytics Platform, we deployed the open-source version of dbt, called dbt Core. A SaaS version called dbt Cloud is also available. dbt is specifically geared to the T (data transformation) in ETL (extract, transform, load), the basic functionality of a data platform. One of its key traits is that most features are available simply via SQL and YAML configurations[*5]. The fact that SQL and YAML configurations are all that is needed is quite a boon for data analysts, who tend to use SQL extensively when performing analyses.

The first challenge related to data quality is addressed by testing the data using dbt. Automated testing is a best practice in software engineering. And this is also a valid approach in data engineering.

dbt includes functionality for testing data, and the tests can also be written in SQL and YAML. Tests in dbt are SELECT statements that return 0 records in a normally functioning system. In other words, if the SELECT statement for a particular test returns one or more records, the test fails. You can of course write your own SELECT statements from scratch, or you can use built-in test macros and test macros implemented by third-party plugins. Additionally, the source freshness feature can be used to detect when data freshness has declined.

*3 The degree to which a decline in freshness can be tolerated depends on the nature of the work performed by the data analyst.

*4 Situations that actually occur range from abnormalities that are obvious from the data definition to cases that are difficult to detect, such as changes in the distribution of data when viewed through summary statistics.

*5 Differences between databases (SQL dialect etc.) are handled by adapters tailored to each database.

In the past, if something unexpected had happened to data on the Data Analytics Platform, data analysts would typically notice this during the course of performing analyses. But when data analysts perform analyses, they do this with an objective in mind. That is, they usually need that data right at that moment. So discovering something unexpected with the data at that point tended to heavily impact on workflow. Deploying dbt, however, made it possible to efficiently discover and address such occurrences by incorporating data definitions and data analysts' experience-based rules into automated tests.

The second challenge relating to creating and maintaining data marts is addressed by using dbt to transform data with SQL. In dbt, objects called models can be defined using SQL SELECT statements. Results transformed by the model's SELECT statement can be accessed in the form of views and tables. The Jinja templating language can also be used to define models. So it is possible to group frequently used content into macros, and to perform looping and branching that is difficult to accomplish in pure SQL.

Deploying dbt on the Data Analytics Platform has made it possible, in many cases, to create data marts simply by writing SQL. In some cases, we have been able to create data marts for quickly searching for security IoCs (Indicators of Compromise) within a short period of time, making certain tasks dozens of times more efficient. Some processes are difficult to implement using SQL alone, however, so individual development has not been rendered entirely unnecessary. Meanwhile, it has become easier than before to divide labor between data engineers and data analysts in some respects. Specifically, data engineers can implement processes that are difficult to perform using SQL alone as SQL user-defined functions. Data analysts can then write routines for transforming the data in dbt using SQL that includes those user-defined functions. With this workflow, tasks follow an existing framework, so we can expect to reduce resource requirements and lead times from what they were before. And laying out a more concrete path for creating data marts has streamlined our operations and made it easier to come up with new ideas for analysis.

1.3.2 Visualizing C&C communications with a focus on time interval variability

When infecting a system, some types of malware use an Internet-connected command and control (C&C) server to receive instructions from an attacker. The basic behavior of the communications that take place (the C&C communications) is predetermined according to what malware program is running. For this reason, certain patterns are more likely to emerge than those seen in, for example, communications that occur when humans use a browser or other such application in an ad hoc fashion.

One key characteristic that tends to reflect particular patterns is the communication time interval. A typical example that should make this easy to see is polling that occurs at regular intervals, such as a heartbeat used in system health monitoring. Time interval variability tends to be low with these sorts of communications. To enable security analysts to efficiently capture C&C communication patterns, we therefore tried out the use of a statistical measure that focuses on time interval variability to generate visualizations of communications.

It must be noted, however, that small time interval variations can occur even with legitimate, non-malware applications. If you detect small communication time interval variations in a production environment, it's best to assume that most of that is attributable to legitimate applications. The method described here is thus intended to be used to help analysts visually understand what's going on when breaches occur.

First, let's go over the problem and the tools that can help us solve it. C&C communications can be viewed as events that occur over the passage of time. Each event

belongs to a specific session[*6], typically identified by the below elements in combination. The source is the system infected with malware, and the destination is a C&C server set up by the attacker. Note that the source port number is not included in the session identifier because it may be an ephemeral port that changes from connection to connection.

.

- Source IP address
- Destination IP address
- Transport layer protocol
- Destination port number

Next, we turn to the descriptive statistical measures of variance and standard deviation, which is based on variance, to quantify the variability in our figures. However, there are some challenges to address when it comes to assessing variability in communication time intervals with variance and standard deviation using all events in a particular session.

■ (a) Comparing values from different sessions is difficult

Consider the following situation, for example. Say that in session A, the average time interval between events is 100 seconds and the standard deviation (a measure of variability) is 10 seconds. Meanwhile, say that in session B, the average time interval between events is 1,000 seconds and the standard deviation is 50 seconds. If we only compare the absolute values, we will decide that the standard deviation is lower for session A than for session B. But if we consider the magnitude of variability relative to the average, we will find this to be smaller for session B than for session A. Hence, variance and standard deviation alone are insufficient when comparing different sessions.

■ (b) Outliers have a substantial impact

If, for example, a malware-infected computer is used for business purposes, it might only be turned on during weekday daytimes. And naturally, when it's not powered on, no C&C communications will occur. Including the times when the computer was off in the calculations would result in a long interval between events, as if the readings were outliers. The calculated variance and standard deviation are likely to be larger than expected as a result.

■ (c) Event time intervals can change partway through sessions

Some malware shortens the time interval between events when the attacker is actively issuing commands. This is possibly because a lot of malware has autonomy over when it goes to the C&C server to receive commands[*7], so if the polling interval is long, it will take time for operations to be reflected in the malware's behavior. When long and short time intervals are mixed like this, it is difficult to express the variability in terms of a single measure like variance or standard deviation.

To address these challenges, we looked at a statistical measure called the moving coefficient of variation. This takes the ordinary coefficient of variation and introduces the idea of moving it over time. The smaller the moving coefficient of variation of the communications time interval, the more we can say that the communications have low variability and are thus regular.

First, the coefficient of variation is obtained by dividing the standard deviation by the mean. As mentioned in challenge (a) above, the standard deviation alone is not enough to determine variability relative to the mean size of the observation. We therefore obtain a dimensionless

*6 For convenience, we refer to a series of exchanges between any particular nodes as a session. This is different from a TCP session.

*7 This is possibly because if the communications originate from a C&C server on the Internet, they are likely to be hindered or blocked by NAT and firewalls.

measure by dividing the standard deviation by the mean. This makes it possible to compare the variability of different sessions.

Next, in statistics, a moving measure is something that is calculated using a moving set of local (as opposed to global) data points. Such measures are used mainly for time series and other continuous data sets and are based on a predetermined number of data points. The range covered by that predetermined number of data points is called the window. The window is progressively moved along the data set so that all of the data is eventually used.

So the moving coefficient of variation is obtained by moving the calculation of the coefficient of variation along the data set. The moving coefficient of variation is calculated by dividing the moving standard deviation by the moving average. This reduces the impact of the issues discussed in challenges (b) and (c) above. As regards (b), outliers only affect the windows in which they occur. And as for (c), even if the time intervals at which events occur change during the observation period, this should show up as changes in the statistical measure over time.

We will now look at how malware-generated C&C communications from past infection incidents can be visualized using the moving coefficient of variation. The first case deals with malware communicating with multiple C&C servers at the same time. In recent times, malware programs often attempt to communicate with multiple C&C servers at the same time in the manner discussed here.

In Figure 3, time elapsed since the reference time is plotted on the horizontal axis, and the moving coefficient of variation is plotted on the vertical axis. Each line represents a session, all of which constituted communications to a C&C

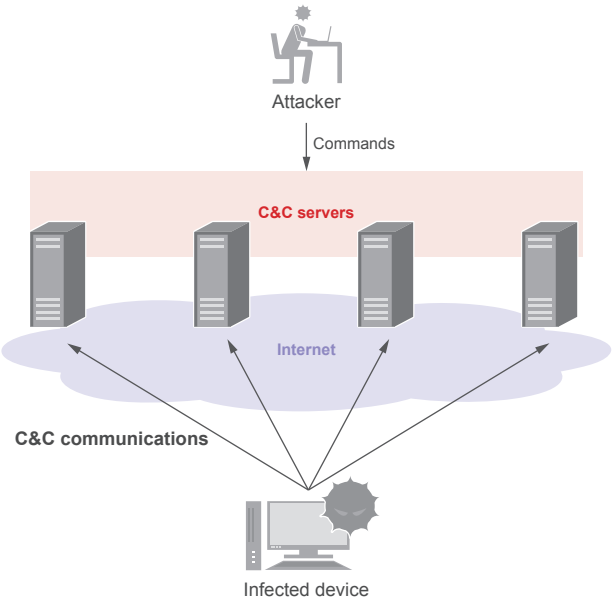

Figure 2: Malware Communicating with Multiple C&C Servers at the Same Time

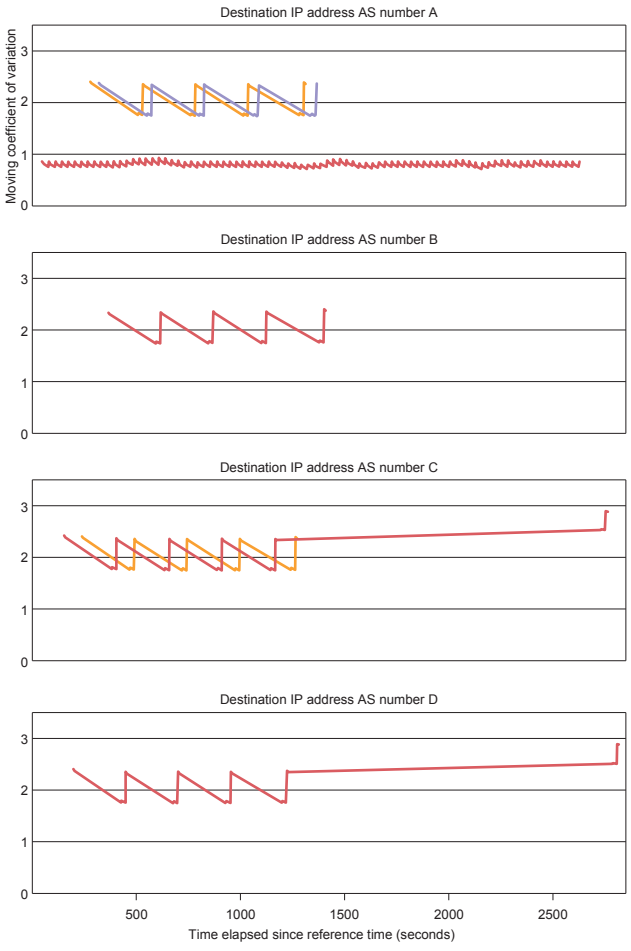

Figure 3: Typical Sawtooth Wave Pattern of Malware Communications to C&C Servers

server. In this case, each session used the same source IP address and transport layer protocol but had different destination IP addresses and port numbers. For ease of viewing, the graph is divided into panels by the AS number of the destination IP address. The window size used to calculate the moving coefficient of variation was 10 data points.

The graphs in each panel of Figure 3 show the typical sawtooth wave pattern, oscillating around a value of 2, in the moving coefficient of variation. The waves are not perfectly aligned with each other, but they are close. This means that attempts to communicate with C&C servers happened at similar times, even across different C&C servers.

The second example shows malware falling back to another C&C server when the C&C server it was originally communicating with becomes unavailable. While this malware only communicates with one C&C server at a time, it likely has the ability to switch to a different C&C server if it loses contact with its current C&C server for some set amount of time.

In Figure 5, as before, time elapsed since the reference time is plotted on the horizontal axis, and the moving coefficient of variation is plotted on the vertical axis. Each line again represents a session, and the window size used to calculate the moving coefficient of variation is 10 data points here also. Note, however, that elapsed time is calculated in minutes here, as opposed to seconds in the previous graphs.

Two sessions are plotted on the graph, and you can see times during which the moving coefficient of variation remained stable at close to zero in both cases. In other words, there are times when communication was occuring at regular intervals such that there was very little variability in the communication time interval. In fact, during these times, the malware was repeatedly communicating with a C&C server at intervals of around 317–320 seconds.

In the session indicated by the orange line, the moving coefficient of variation increases from around the 200-second mark[*8] and the line subsequently cuts out. A little while after this, the purple line appears. This is

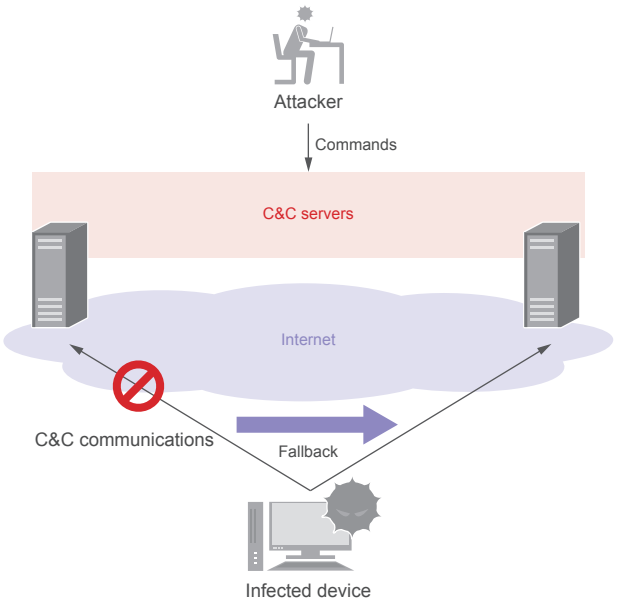

Figure 4: Malware Falling Back to Another C&C Server

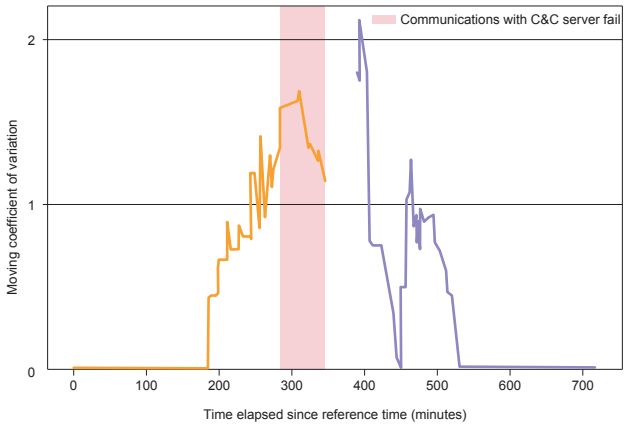

Figure 5: Plot of Communications When Malware Falls Back to Another C&C Server

*8 An increase in the moving coefficient of variation indicates a breakdown in the regularity of communication intervals partway through the session. This corresponds to challenge (c) above.

what it looks like when the malware falls back to another C&C server. In the orange session, communications with the C&C server had failed about an hour before the line cut out. The malware, it seems, therefore switched to sending its communications to a different C&C server.

Note that the examples covered here are typical ones and that there is a whole range of malware communication patterns in the wild. It is not uncommon, for instance, for malware to vary communication intervals depending on which C&C server it is communicating with, or to introduce some degree of randomness. These sorts of behaviors are likely attempts to avoid detection based on communications time interval variability. We have observed that, even in such cases, the shape of the graphs of the moving coefficient of variation exhibit similarities, and that the coefficient will move within a particular range.

Here, we have described our work to enable visualizations of communications traffic using a statistical measure that focuses on time interval variability. IIJ's SOC will continue to study techniques for efficient data-based security analysis.

1.4 Conclusion

In this article, we discussed several incidents that our SOC focused on out of those that we observed in 2023. The annual summary in Section 1.2 indicated that software vulnerabilities continue to come to light, and that ransomware attacks as well as information breaches due to configuration errors and external attacks continue to occur. Section 1.3.1 described how we used dbt to address issues related to data freshness and quality and the creation and maintenance of data marts through automated testing and the use of SQL for data conversion in a way that does not require individual development. Finally, Section 1.3.2 discussed how we arrived at the idea of using the moving coefficient of variation, along with some actual use cases, as a means of addressing challenges that arise when assessing the variability in communications time intervals based on variance and standard deviation—namely, the challenges of comparing different sessions, addressing the impact of outliers, and addressing changes in time intervals.

IIJ will continue to publish information to address the ever-changing array of threats out there. We hope that you will continue to turn to the IIR and wizSafe Security Signal for such information and that it will prove useful to you in your security responses and operations.

Junya Yamaguchi
Data Analytics Section, Security Operation Department, Advanced Security Division, IIJ

Hiroyuki Kamogawa
Data Analytics Section, Security Operation Department, Advanced Security Division, IIJ

Satoshi Kobayashi
Data Analytics Section, Security Operation Department, Advanced Security Division, IIJ