PERSONAL DATA PROTECTION COMMISSION

[2022] SGPDPC 8

Case No. DP-2010-B7266

In the matter of an investigation under section 50(1) of the Personal Data Protection Act 2012

And

RedMart Limited

… Organisation

DECISION

Data Protection *– Protection obligation – Unauthorised disclosure of personal data – Insufficient security arrangements – Failure to implement reasonable access controls*

Data Protection *– Financial penalty – Voluntary admission of liability*

RedMart Limited

[2022] SGPDPC 8

Lew Chuen Hong, Commissioner — Case No. DP-2010-B7266

28 October 2022

Introduction

1 Many organisations rely on web-based application programming interfaces ("**API**") to enable computers or computer programs to communicate and facilitate the sharing of data between them. API keys are in turn used to authenticate users seeking to access APIs. If an organisation fails to implement reasonable security measures to safeguard the security of their API keys, this may allow threat actors unauthorised access to large troves of data stored within multiple interconnected environments.

2 On 29 October 2020, the Personal Data Protection Commission ("**the Commission**") was notified that a database containing personal data of the customers of RedMart Limited (the "**Organisation**") was being offered for sale on an online forum (the "**Incident**"). Subsequently, the Commission commenced investigations to determine whether the circumstances relating to the Incident disclosed any breaches by the Organisations of the Personal Data Protection Act 2012 ("**PDPA**").

Facts of the Case

3 The Organisation operated an online platform selling groceries and fresh produce to consumers. In 2016, the Organisation was acquired by Lazada Group ("**Lazada**"). Thereafter,

the Organisation began to integrate its platform with Lazada's online platform. The customer-facing website and mobile application ceased operations on 15 March 2019. However, on the back end, the migration and integration of the Organisation's system into Lazada's system was not completed by that time. It is worth setting out in some detail the Organisation's information technology architecture to understand the backdrop against which the Incident occurred.

4 From March 2012 until its acquisition by Lazada, the Organisation's business applications (including its customer facing website, mobile application, warehouse and delivery back-end applications) were stored in RedMart's Amazon Web Services Virtual Public Cloud (the "**AWS Environment**"). The personal data of its customers and sellers were in turn, always stored in a MongoDB database within RedMart's Alibaba Virtual Public Cloud. Whilst the MongoDB database is stored within cloud infrastructure that belongs to the Alibaba Group, the Organisation remained responsible for managing its cloud environment (hereinafter, the "**RedMart Cloud**"). The Organisation did not encrypt the MongoDB database, or implement any password authentication requirement to access the MongoDB database.

5 After its acquisition, the Organisation's intention was to re-design and migrate all the relevant databases and applications from the AWS Environment into the RedMart Cloud to facilitate its integration with Lazada's systems and environment by March 2021. However, given the substantial time and resources required to complete the re-design and migration, the Organisation opted to carry out the migration in stages. Following the acquisition and as a matter of priority, the Organisation migrated its front end, customer-facing systems to the RedMart Cloud to enable a seamless integration into Lazada's environment and platform for its customers. This was completed on 15 March 2019, after which the Organisation shut down its public-facing consumer application and website. Additionally, the MongoDB database

residing in the RedMart Cloud containing the affected customer and seller data was disconnected from the application and website (the "**Affected Database**"), and thereafter access by the Organisation's customers was disabled.

6 In contrast, the Organisation's back-end business applications and systems (such as order fulfilment, inventory, transport and warehousing) and its seller portal continued to be hosted on the AWS Environment and linked to the Affected Database contained in the RedMart Cloud.

7 Concomitantly, the Organisation's back-end systems and IT infrastructure during the interim period was structured in the following manner:

(a) The Organisation's GitHub Enterprise repositories (the "**GitHub Repositories**"), were accessible by the Organisation's employees possessing GitHub user and administrative accounts. The Organisation used the GitHub Repositories to store, amongst other things, sensitive source codes including a Chef[1] key that functioned as a high privilege access key to the AWS Environment. The Organisation implemented the following access control measures vis-à-vis the GitHub Repositories:

i. For GitHub user accounts, the Organisation followed GitHub's password policy, which required the use of passwords which were either eight characters long if it included a number and a lowercase letter, or 16 characters long with any combination of characters; and

ii. For GitHub administrator accounts, two-factor authentication ("**2FA**") was required on top of the password requirements above.

[1] "Chef" is an orchestration tool used for automated provisioning and management purposes within an AWS environment.

(b) The AWS Environment was accessible through the Chef key, and contained various private Simple Storage Service ("**S3**") buckets[2], one of which contained another sensitive API key – the Hubot[3] key.

(c) Lastly, the AWS Environment was connected to the RedMart Cloud through an OpenVPN[4] connection. The Affected Database was stored within the RedMart Cloud.

The Incident

8 Sometime in September 2020, an unidentified threat actor ("**TA**") gained unauthorised access to the GitHub user account of a member of the Organisation's DevOps (i.e. software development and IT operations) team. The TA utilised the compromised GitHub user account to search through the GitHub Repositories and found the Chef key. Thereafter the TA used the Chef key to access the AWS Environment, whereupon he scanned through the Organisation's private S3 buckets and located the Hubot key.

9 Using the Hubot key, the TA created a rogue Amazon Elastic Compute Cloud ("**EC2**") instance in the AWS Environment. The TA also created a new firewall rule to enable a connection between the rogue EC2 instance and another part of the Organisation's network hosted by the RedMart Cloud. The TA then traversed the OpenVPN connection (that linked the AWS Environment to the RedMart Cloud) to access the RedMart Cloud.

10 Once the TA had gained access to the RedMart Cloud, he proceeded to exfiltrate the Affected Database on 6 September 2020. Subsequently, the Affected Database was found on an online forum being offered for sale.

---

[2] S3 buckets are public cloud storage resources in AWS which are similar to file folders.
[3] Hubot is a chat software that can be scripted to interact with an IT environment.
[4] OpenVPN is a virtual private network system that implements techniques to create secure point-to-point or site-to-site connections in routed or bridged configurations and remote access facilities.

11 The Affected Database contained the following personal data:

	Customer accounts	**Seller business accounts**
Number of affected individuals	**898,791**	
Types of personal data	a. Name b. Email address c. Contact number d. Residential address e. Partial credit card details comprising: • First 6 and last 4 digits of card number • Card owner's name • Expiry date • Credit card billing phone number and billing address • Hashed account password • URL links of call recordings between customer service agents and the Organisation's customers	a. Partial credit card number comprising first 6 and last 4 digits b. Hashed account password

Remedial actions

12 Following the Incident, the Organisation and Lazada implemented the following remedial measures:

Actions to mitigate the effects of the Incident

(a) Disabled and reset the affected Chef and Hubot keys.

(b) Disabled the compromised GitHub user account.

(c) Reset all other existing AWS API keys, GitHub user credentials and tokens

(d) Deleted the EC2 instance created by the TA.

(e) Logged out and reset of all affected customer and seller accounts on 30 October 2020.

(f) Reviewed all of the Organisation's servers with services connected to Internet. All sensitive services were filtered by a firewall.

(g) Conducted vulnerability scanning of 31 Internet facing IP addresses.

(h) Investigated all other existing MongoDB databases to search for traces of the TA.

(i) Monitored the Lazada login page for brute force attacks.

(j) Informed all affected individuals of the Incident via emails and broadcast on the Lazada online and application platforms on 30 October 2020. A media statement was also issued at the same time.

Actions to prevent recurrence of the Incident or similar incidents

(k) Implemented database authentication for all databases containing personal data. Restricted access to sensitive database from only authorised source IP addresses instead of network ranges.

(l) Implemented 2FA for all GitHub accounts and removed unnecessary GitHub accounts and developer access keys.

(m) Scanned for vulnerabilities in AWS critical Virtual Private Cloud instances.

(n) Performed a security architecture review of the Organisation's multiple cloud network and intra-cloud micro-segmentation.

(o) Reviewed all AWS Identity and Access Management user permissions to ensure no "create instance" permission.

(p) Set up identity access management rules to restrict geographical locations from which API keys could be used to access the Organisation/Lazada cloud environments.

(q) Reviewed all user identities and access privileges for all systems and applications used. Removed access privileges of all inactive accounts.

(r) Implemented network traffic logging between the AWS Environment and RedMart Cloud.

(s) Implemented network Access Control list and endpoint detection and response for windows instances for RedMart Cloud.

(t) Implemented security monitoring to detect creation of any new cloud instance

(u) Enabled Virtual Private Cloud (VPC) logs for security monitoring.

(v) Required all of the Organisation's employees to complete an online training course on privacy management and responsibilities in handling personal data.

Findings and Basis for Determination

The Protection Obligation under section 24 of the PDPA

13 Based on the circumstances of the Incident as set out above, the Commission's investigation focused on whether the Organisation had breached its obligation under section 24 of the PDPA to protect personal data in its possession or under its control by taking reasonable security steps or arrangements to prevent unauthorised access and disclosure ("**the Protection Obligation**").

14 In determining what constitutes reasonable security steps or arrangements, an organisation should have regard to the nature of the personal data in its possession and control, as well as the impact that the disclosure of the data might have on the affected persons. As stated in the Commission's Advisory Guidelines on Key Concepts in the PDPA (the "**Advisory Guidelines**")[5] at [17.2]:

> "There is no 'one size fits all' solution for organisations to comply with the Protection Obligation. Each organisation should consider adopting security arrangements that are reasonable and appropriate in the circumstances, for example, taking into consideration **the nature of the personal data, the form in which the personal data has been collected (e.g. physical or electronic) and the possible impact to the individual concerned if an unauthorised person obtained, modified or disposed of the personal data.** For example, in the employment context, it would be reasonable to expect a greater level of security for highly confidential employee appraisals as compared to more general information about the projects an employee has worked on."
>
> (emphasis added)

[5] https://www.pdpc.gov.sg/guidelines-and-consultation/2020/03/advisory-guidelines-on-key-concepts-in-the-personal-data-protection-act

15 Given the high volume of personal data contained in the Affected Database (approximately 898,791 individuals, encompassing personal data such as names, email addresses, residential addresses, contact numbers and partial credit card details), it was incumbent on the Organisation to implement policies and practices that were commensurate with the Organisation's higher-level security needs to discharge its obligation under the Protection Obligation.

16 For the reasons set out below, it is determined that the Organisation failed to implement reasonable security arrangements to protect the Affected Database from the risk of unauthorised access.

Whether the Organisations had contravened the Protection Obligation

17 Based on the Organisation's IT architecture as detailed above, the Affected Database was placed behind various levels of security controls within RedMart Cloud. This meant that a threat actor had to get through various access points to access the Affected Database. The implementation of network segmentation as part of layered defence is a reasonable strategy for defence-in-depth. However, a complex IT architecture can be defeated by simple errors, especially when the high-value data embedded within such complex systems are so often the target of sophisticated threat actors. The complexity of the Organisation's network architecture does not paper over the cracks in its security arrangements – at every level of defence, the Organisation's systems presented clear vulnerabilities that should have been addressed.

18 First, the Organisation failed to implement reasonable access control on its employers' user GitHub accounts, which allowed the TA access to the GitHub Repositories. As stated in paragraph [7(a)], the access control measures were more stringent for GitHub administrative

accounts than for GitHub user accounts. While this adequately dealt with the differentiation of ability to make configuration and other changes to the Organisation's GitHub Repositories, it did not make any distinction to the type of data stored within the repositories. This disparity in access controls was not sensitive to the fact that both types of accounts had equal abilities to access important files within the GitHub Repositories including the Chef key, which provided access to the AWS Environment. Data with higher security implications (such as the Chef Key) ought to be secured to a higher degree than other types of data. It is, of course, open to the Organisation to secure all data in its GitHub Repositories at the higher level. Indeed, as stated in paragraph 12(l), the Organisation and Lazada, as part of their remedial measures post-Incident, implemented 2FA for all its GitHub accounts.

19 Second, the Organisation did not implement sufficient access controls to protect and limit access to the Chef and Hubot keys, which enabled highly privileged access to various environments within the Organisation's systems. Not all accounts need access to the Chef and Hubot keys; and even for accounts that had access to them, they ought to be periodically reviewed to determine whether continued access were necessary. At an organisation-wide level, investigations revealed that the Organisation did not conduct periodic management reviews to ensure that the access to Chef and Hubot keys were limited to the GitHub accounts that required such access, or to remove access from accounts that no longer needed it (including removing unnecessary GitHub accounts altogether). This is a fundamental data security practice. As suggested in the Commission's Guide to Data Protection by Design for ICT Systems[6]:

"**12. Regularly review user accounts**

[6] https://www.pdpc.gov.sg/-/media/Files/PDPC/PDF-Files/Other-Guides/Guide-to-Data-Protection-by-Design-for-ICT-Systems-(310519).ashx?la=en

> This ensures that all user accounts are legitimate. **There should be processes to update or remove user accounts, for instance, when a user has left the organisation.** Test accounts should also be removed after test activities have been completed.
>
> Separately, there should also be a process to review user accounts regularly. **The review should include ensuring all the rights assigned are indeed necessary.**"
>
> (emphasis added)

20 In this connection, best practice dictates that the principle of least privilege should apply i.e. each employee be given only the minimum level of access rights or privileges necessary for that employee to complete an assigned operation. This would limit the damage in case a vulnerability is exploited, as in this case where the TA gained unauthorised access to a GitHub user account. This principle was also espoused in GitHub's own Secure Design Principles[7]. The Organisation's failure to limit access to the Chef and Hubot keys at the material time allowed the TA to access the Chef key, and consequently the AWS Environment, through a GitHub user account.

21 On a more granular level, insufficient security measures were implemented to protect the Chef and Hubot keys, as both were stored in plain text files that were not encrypted or even password-protected. Specifically:

(a) The Chef key was hardcoded in source code and stored in the GitHub repositories, and was accessible to a large group of developers from the Organisation, Lazada and Alibaba for the purpose of knowledge sharing. By allowing so many accounts to access the Chef key in such a manner, the risk of exposure and exploitation was heightened.

[7] https://github.com/OWASP/DevGuide/blob/master/02-Design/01Principles%20of%20Security%20Engineering.md

(b) The Hubot key was stored in plain text in an AWS private S3 bucket. Therefore, any account that accesses the AWS Environment was able to access the Hubot key as a conduit from which to access the RedMart Cloud and the Affected Database stored therein.

22 The manner in which the Chef and Hubot keys were stored in the GitHub Repositories and AWS Environment were manifestly inadequate in view of the circumstances, and this vulnerability was exploited by the TA to access the keys after he gained access to the GitHub Repositories and AWS Environment respectively. Ideally, such keys ought to be stored in a separate location such as Secrets Manager within the AWS Environment (i.e. a dedicated key vault). *The Organisation should not have embedded the Chef and Hubot keys directly in the source code*. This is also reflected in AWS Access Keys best practices[8], which the Organisation should have been aware of given its usage of the AWS Environment since 2012:

> **"**Observe these precautions when using access keys:
>
> - **Don't embed access keys directly into code. The AWS SDKs and the AWS Command Line Tools enable you to put access keys in known locations so that you do not have to keep them in code.**"
>
> (emphasis added)

23 Such best practices applies to of all platforms that use APIs, and are not confined to AWS. They are echoed in Google's best practices for securely using API keys[9], which are meant to apply to Google Cloud but are nevertheless apposite here:

[8] https://docs.aws.amazon.com/general/latest/gr/aws-access-keys-best-practices.html
[9]https://support.google.com/googleapi/answer/6310037?hl=en#:~:text=Delete%20unneeded%20API%20keys%3A%20To,use%20the%20newly%2Dgenerated%20keys

"Best practices for securely using API keys

When you use API keys in your Google Cloud Platform (GCP) applications, take care to keep them secure. Publicly exposing your credentials can result in your account being compromised, which could lead to unexpected charges on your account. To keep your API keys secure, follow these best practices:

- **Do not embed API keys directly in code**: API keys that are embedded in code can be accidentally exposed to the public, for example, if you forget to remove the keys from code that you share. **Instead of embedding your API keys in your applications, store them in environment variables or in files outside of your application's source tree.**
- Do not store API keys in files inside your application's source tree: If you store API keys in files, keep the files outside your application's source tree to help ensure your keys do not end up in your source code control system. This is particularly important if you use a public source code management system such as GitHub.
- Restrict your API keys to be used by only the IP addresses, referrer URLs, and mobile apps that need them: **By restricting the IP addresses, referrer URLs, and mobile apps that can use each key, you can reduce the impact of a compromised API key.** You can specify the hosts and apps that can use each key from the GCP Console Credentials page and then create a new API key with the settings you want, or edit the settings of an existing API key.
- Restrict your API keys to be usable only for certain APIs: If you have multiple APIs enabled in your project and your API key should only be used with some of them, restrict usage of that key to those APIs. You can specify the allowed

APIs for each key from the GCP Console Credentials page and then create a new API key with the settings you want, or edit the settings of an existing API key.

- Delete unneeded API keys: To minimize your exposure to attack, delete any API keys that you no longer need.
- Regenerate your API keys periodically: You can regenerate API keys from the GCP Console Credentials page by clicking Regenerate key for each key. Then, update your applications to use the newly-generated keys. Your old keys will continue to work for 24 hours after you generate replacement keys."

(emphasis added)

24 The Organisation claimed that it was constrained from implementing password protection and encryption for the Chef and Hubot keys by technical issues, and that the implementation of password protection or encryption was not feasible or practical given the keys' core administrative and automating technical functions. The mere existence of technical issues does not exculpate the Organisation, or justify the Organisation's decision to embed the Chef and Hubot keys into source code. Technical issues are not insurmountable, and it is open to the Organisation to expend the necessary time, effort and resources to troubleshoot and resolve them. However, investigations revealed that the Organisation did not make sufficient efforts to resolve the technical issues in a timely manner, and were thus responsible for its difficulties in implementing access controls for the Chef and Hubot keys. More broadly, even if the Chef and Hubot keys were not compatible with password protection or encryption, there were a variety of other methods available to safeguard the security of the keys as set out in the best practices set out above, such as removing unnecessary GitHub accounts, restricting access to only certain GitHub administrative accounts or storing the keys separately in a configuration

file or a key vault. Nothing prevented RedMart from adhering to such best practices, as is evident by the fact that all these measures were implemented after the Incident occurred.

25 Lastly, the Organisation also did not implement separate authentication requirements, for the Affected Database. Once the TA accessed the RedMart Cloud, this enabled the TA to access and exfiltrate the Affected Database. Instead, the only measures implemented to safeguard the Affected Database were the access requirements for the RedMart Cloud environment in general, which was circumvented by the TA through his possession of the Hubot key. This reflects a failure of the Organisation's attempt to deploy a reasonable "defence in depth" strategy (despite its multiple layers of access) to safeguard the security of the Affected Database, as it should have implemented separate authentication requirements for the Affected Database to prepare for the contingency of someone gaining unauthorised entry into the RedMart Cloud.

26 At a basic level, this should have included access controls such as password protection. Further, given the high volume of personal data contained in the Affected Database, the Organisation should also have taken steps to implement more stringent access controls, such as limiting access only to certain authorised network connections / interfaces. The Center for Internet Security MongoDB 5 Benchmark[10] recommends requiring authentication of, amongst others, users and servers before allowing access to a MongoDB database on the following basis:

> "**2 Authentication**
>
> This section contains recommendations for requiring authentication before allowing access to the MongoDB database.

[10] https://cissecurity.org/benchmark/mongodb

…

Rationale:

Failure to authenticate clients, users, servers can enable unauthorized access to the MongoDB database and can prevent tracing actions back to their sources.

It's highly recommended that password length and complexity also be in-place. When performing the traditional user/password authentication against MongoDB there is not in-place intrinsic password complexity check and there is no LOCKING mechanism with multiple failure logins. **So, MongoDB is prone to brute force attacks compared to other database systems.**"

(emphasis added)

27 The Organisation, again, cited technical difficulties for why separate authentication requirements were not implemented for the Affected Database. Again, such technical difficulties could have been overcome if the Organisation had expended the effort and resources to do so.

The Commissioner's Decision

28 In determining whether any directions should be imposed on the Organisation under section 48I of the PDPA, and/or whether the Organisation should be required to pay a financial penalty under section 48J of the PDPA, the factors listed at section 48J(6) of the PDPA were considered, with particular emphasis on the following mitigating factors:

(a) the Organisation was cooperative with the Commission's investigations, responding to the Commission's queries in a prompt and forthcoming manner;

(b) the Organisation had put in place some good data de-identification practices for the credit card numbers in its possession by redacting part of the data and storing only partial credit card details. This reduced the sensitivity and harm that might be caused when personal data within its control are disclosed in a data breach; and

(c) the Organisation swiftly implemented effective measures to mitigate the effects of the incident and prevent recurrences.

29 The Organisation was notified of the preliminary decision by way of the Commission's letter dated 2 August 2022 and was invited to make representations.

Representations Made by the Organisation

30 On 23 August 2022, the Organisation made the following representations to the Commission seeking a reduction in the financial penalty:

(a) The Organisation had voluntarily notified the Commission and affected individuals of the Incident even though it was not legally obliged to do so at the material time, as the Data Breach Notification Obligation under Part 6A of the PDPA only came into effect on 1 February 2022;

(b) As of the date of the representations, the Organisation was not aware of any personal data of the affected individuals being misused as a consequence of the Incident. In support, the Organisation stated that the data in the Affected Database relates to data used on its previous application and website which is no longer in use. Moreover, the data was more than 18 months out of date, and was not linked to any current Lazada databases. Although the Organisation's account passwords had been securely encrypted at the time of the Incident, it had conducted a forced logout and

password reset on every affected current Lazada account upon discovery of the Incident to ensure that no third party could misuse the leaked passwords by logging into a current Lazada account; and

(c) The Incident was the Organisation's first breach of the PDPA. To the best of the Organisation's knowledge and belief, the Incident was its first experience of a data breach.

Voluntary notification

31 The Commission had already taken into account the Organisation's voluntary notification of the Incident into account in determining the quantum of the financial penalty in the preliminary decision. Hence, this factor does not warrant a further reduction in the financial penalty imposed.

Lack of misuse of the affected personal data

32 The fact that the Organisation was not aware of any misuse of the affected personal data is neither here or there, and does not merit a further reduction in the financial penalty. Whilst evidence of exploitative use may be a relevant factor of harm that may be relevant for enhancing the financial penalty, the inverse is not necessarily true.

33 In support of its representations, the Organisation cited the case of ***Learnaholic Pte Ltd*** [2019] SGPDPC 31 ("***Learnaholic***"), in which the Commissioner had taken into account the fact that *"while there was actual exfiltration of the Personal Data…there was no evidence of*

further exploitation, use or disclosure of the Personal Data by the attacker".[11] It suffices to say that the Commission has explained *Learnaholic* in *Farrer Park Hospital Pte Ltd* [2022] SGPDPC 6 at [35 to 36] as follows:

> "35 In the case of *Learnaholic* the main factors taken into account when deciding to reduce the preliminary financial penalty imposed were:
>
> (a) A reduction in the total number of affected individuals due to a recalculation of figures; and
>
> (b) The benefit of doubt given to Learnaholic as to the period of time the vulnerability in its system existed.
>
> 36 The lack of evidence of further exploitation, use or disclosure is not, *ipso facto*, a factor meriting a reduction of the financial penalty. The Organisation's representations are not accepted as the lack of an aggravating factor (i.e. subsequent exploitation, use or disclosure of personal data) is not in itself a mitigating factor."

34 The explanation in *Farrer Park Hospital* that is cited above is equally applicable in the present case.

Lack of antecedents

35 In calibrating the financial penalty in the preliminary decision, the Commission had already taken into account the fact that this was the Organisation's first non-compliance with the PDPA, and no further reduction is merited. In support of its representations, the Organisation also cited *Aviva Ltd and Toh-Shi Printing Singapore Pte Ltd* [2016] SGPDPC 15 ("***Aviva***"), where the Commission had taken into account the fact that the breach of the Protection Obligation by the organisation was its second breach in approximately one year, and

[11] *Learnaholic* at [34].

both data breach incidents involved similar fact patterns and causes.[12] In *Aviva*, the Commission had considered the organisation's previous non-compliance as a contributory factor that justifies an increase in the financial penalty meted as the organisation had failed to improve its data protection practices. In contrast, the lack of antecedents in this instance means that the Commission did not increase the financial penalty imposed on the Organisation. To be clear, the absence of an antecedent does not, *ipso facto*, merit a reduction in the financial penalty imposed.

Acceptance of the Commission's findings

36 Notwithstanding the foregoing, the Commission notes that the Organisation voluntarily accepted the Commission's findings in the preliminary decision, that it had failed to comply with the Protection Obligation and explicitly indicated that it would not seek to challenge these findings. The Organisation's voluntary acceptance of liability (even at this late stage) ought to be reflected in the financial penalty. Had the Organisation accepted responsibility for the Incident at an earlier stage of the investigation, it could have significantly shortened the time for investigations and resolution of this case through the expedited breach procedure and also benefited from greater mitigatory considerations. Nonetheless, an organisation that voluntarily accepts responsibility for its non-compliance with the PDPA should be recognised as an organisation that demonstrates its commitment to the Accountability Obligation and shows that it can be responsible for the personal data in its possession or under its control.[13]

37 Having considered all the relevant factors in this case, the Commission hereby requires the Organisation to pay a financial penalty of $72,000 within 30 days from the date of the

[12] *Aviva* at [38(c)].
[13] Refer to section 11(2) of the PDPA.

directions, failing which interest at the rate specified in the Rules of Court in respect of judgment debts shall accrue and be payable on the outstanding amount of such financial penalty until the financial penalty is paid in full.

38 In view of the remedial actions already been taken by the Organisation, no further directions need be issued to the Organisation.

YEONG ZEE KIN
DEPUTY COMMISSIONER
FOR COMMISSIONER FOR PERSONAL DATA PROTECTION