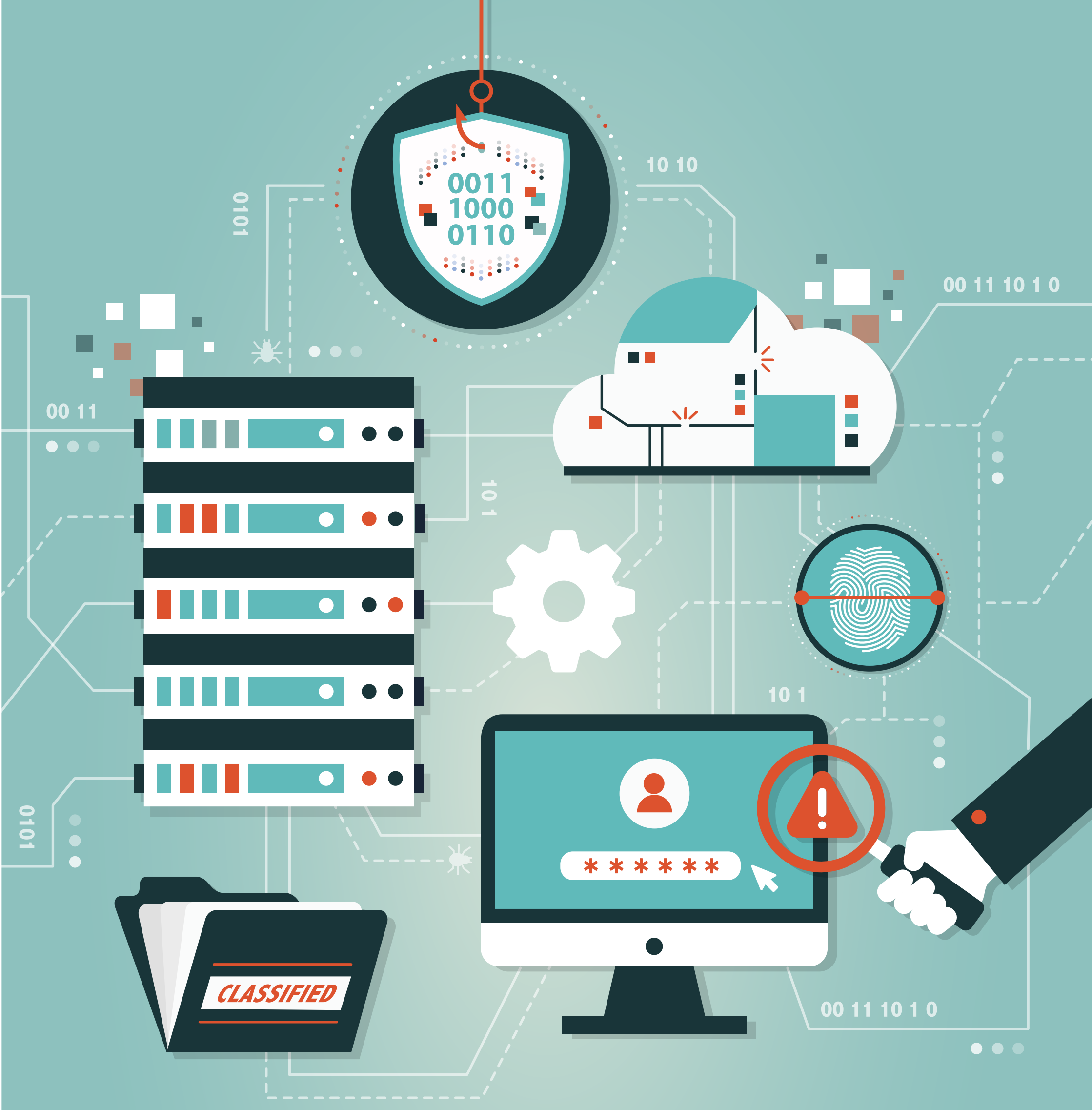
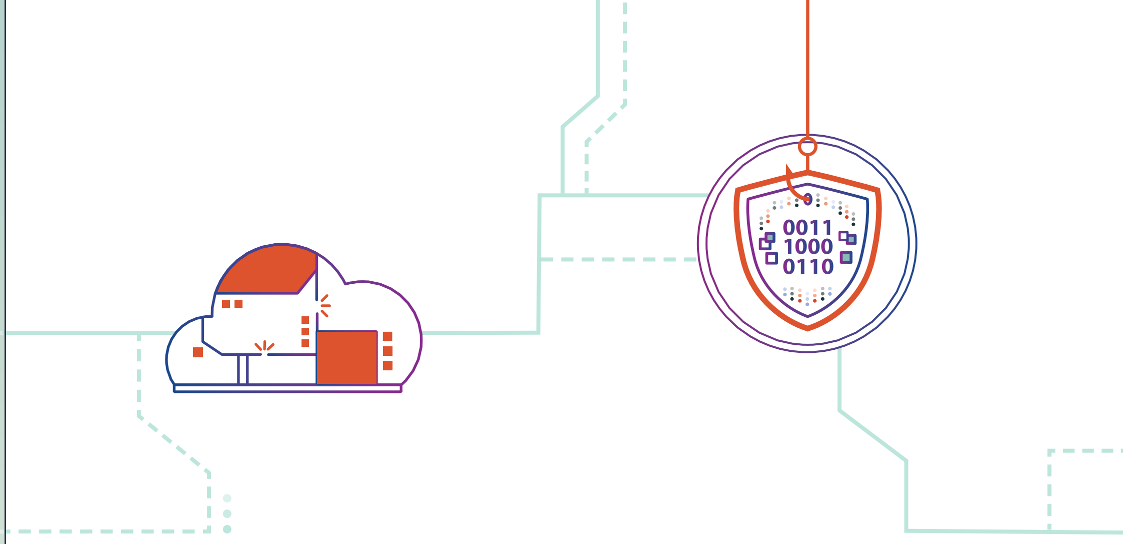


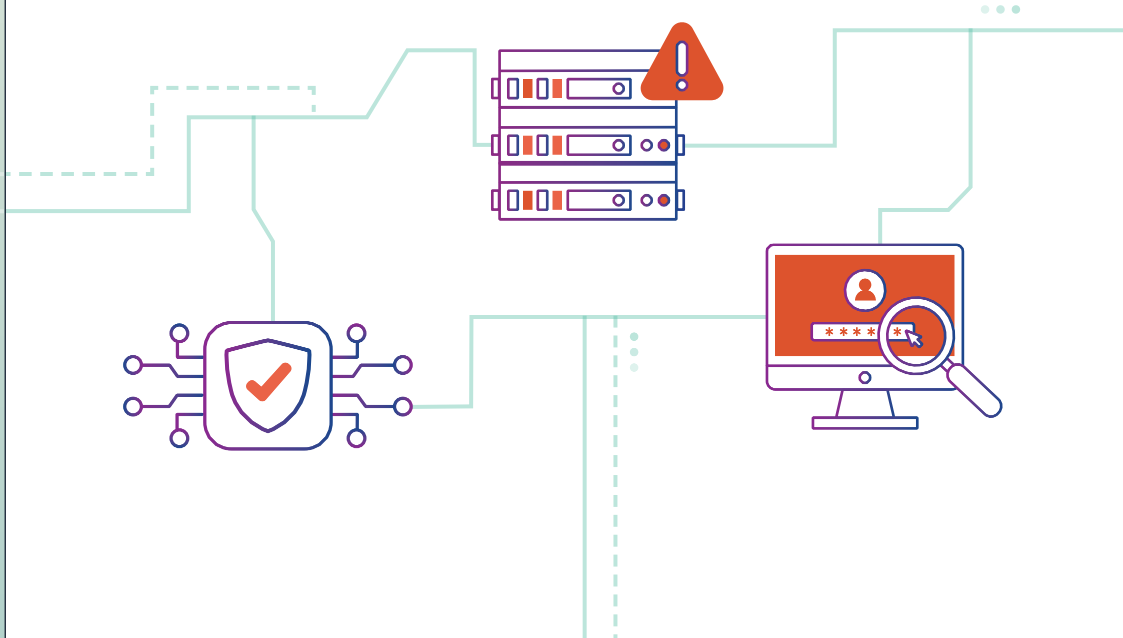
HOW TO GUARD AGAINST COMMON TYPES OF DATA BREACHES?

SG:D
EMPOWERING POSSIBILITIES

pdpc PERSONAL DATA PROTECTION COMMISSION SINGAPORE



Based on past data breach cases handled by the Personal Data Protection Commission (PDPC), this handbook identifies the **five most common gaps** in ICT system management and processes. It also identifies the corresponding ICT good practices that organisations should put in place to prevent data breaches.



MOST COMMON ISSUE

CODING ISSUES

Mistakes made during the programming phase of software development can lead to application errors that result in disclosure of personal data.

CASE EXAMPLES

A Organisation A needed new codes to interact with an existing Application Programming Interface (API). Organisation A's poor documentation on the API caused the developer to make incorrect assumptions when writing the new code. This resulted in the wrong usage of the API and led to the application unintendedly displaying customer information.

B Organisation B developed new code to fix an existing error, but inserted the code segment into the wrong section of the application because many parts of the application's code appeared similar. Organisation B did not detect this error, and it led to the retrieval of information about the wrong customers during data processing.

C Organisation C used the wrong data field while creating a lookup query, as several data fields appeared similar. The correct data field contained unique values of each person, but the wrong data field that was used contained non-unique values of the persons. Consequently, the lookup query retrieved incorrect records of multiple persons instead of the unique record of the intended person. This resulted in the retrieval and disclosure of incorrect data.

D Organisation D's developer unintentionally removed a line of code from a webpage in Organisation D's web application. The purpose of the removed code was to authenticate users. Without proper authentication, and coupled with URL manipulation vulnerability on the webpage, the affected webpage became publicly accessible and personal data of some customers could be viewed by unauthorised users.

The letters marked at each recommendation indicate the case examples where it may have helped the respective organisation prevent the data breach incident.

OBSERVATIONS

- Clear business requirements translated into **clear technical implementation and adequate planning of testing scenarios** can help detect and rectify such programming errors.
- **Careful code reviews** could have detected most, if not all, of the programming errors in the incidents from this category.
- **Poor documentation** is also often a factor; errors are made when there is a lack of clear knowledge of how other components or modules of the ICT system work. This is especially so when the organisation manages a multitude of ICT systems, development teams, and third-party vendors.

RECOMMENDATIONS

A C

Design before coding. Practise designing before coding and perform thorough impact analysis (e.g. traceability and dependency analysis) of any software or code changes to identify the potential effects of these changes. Organisations should also systematically document their code design, changes, and analysis for proper assessment, review, and verification.

A C

Invest effort to document all software functional and technical specifications (e.g. program specifications, system specifications and database specifications). The usefulness of this documentation will become even more apparent over time as the original developers move on from the project and new developers take over the software maintenance and upgrading. Without proper documentation, developers often have no references to fall back on, and may end up making assumptions about code logic that could produce incorrect results.

A B C D

Ensure that the application is thoroughly tested with comprehensive testing such as unit testing, regression testing, security testing, and User Acceptance Testing (UAT). Most organisations fail to recognise that proper testing can help them to identify defects in programming before a system is launched. Sufficient resources should be allocated for testing, and a comprehensive UAT should ensure good test coverage of scenarios including possible user journeys and exception handling. Organisations should also ensure that the planned UAT scenarios match real-world usage. This can be done through a comprehensive gathering of business requirements and identification of relevant usage scenarios by potential users. These should be driven by the business owner.

A B C D

Perform code reviews. In addition to reviewing their own code, code authors can also conduct peer code reviews, which can be effective at catching programming errors and can complement other forms of testing. When performed by an experienced developer, a code review can be very effective.

CONFIGURATION ISSUES

An ICT system often consists of various components (e.g. application/web server, database, operating system, firewall). Many of these components have configurable settings and parameters.

This is a broad category and for the purpose of this handbook, issues in code management and deployment are also classified under this category. Unsecured settings, including leaving settings in their default, can result in unintended disclosure of personal data.

CASE EXAMPLES

E

Organisation E used a WordPress plugin to implement a customer registration facility. Organisation E was not familiar with the plugin and assumed that all data collected was privately accessible to only the administrator. Although the details of the plugin and its settings were described in the product documentation, Organisation E did not configure the plugin appropriately. Hence, the data collected was made publicly accessible and personal data was disclosed.

F

Organisation F discovered an error in their application and in order to fix it, retrieved their application's code to debug. However, Organisation F did not store their code systematically, but in several different code repositories. They subsequently retrieved, debugged and deployed an outdated copy of the code. The outdated code contained errors that led to the disclosure of personal data.

G

Organisation G specified the wrong URL for its antivirus software updates to be downloaded from. As a result, its antivirus software was outdated for an extended period and eventually some of Organisation G's computers became infected by malware.

H

During troubleshooting to fix an application error, Organisation H temporarily configured a web folder to be publicly accessible, but forgot to restore the original setting after troubleshooting. Hence, the web folder was left publicly accessible and this led to a data breach.

The letters marked at each recommendation indicate the case examples where it may have helped the respective organisation prevent the data breach incident.

OBSERVATIONS

- **Configuration issues** can result in vulnerabilities. Examples of common issues include the following:
  - **Website** - not using HTTPS protocol
  - **Firewall** - firewall rules not configured properly to allow only legitimate traffic
  - **Software** - for an antivirus software, not scanning certain file types, or not specifying sufficient follow-up action upon detection of malware
  - **Folder permissions** - not setting restrictions/access control for access to folders with personal data
- **Code management/deployment issues** such as configuration issues in code management/deployment systems can result in wrongly deploying test code to production environment.
- Unlike code, which can be easily stored and managed in a code repository, configuration settings exist in different forms and in the different modules/components of an ICT system. Therefore, they may be more challenging to document, keep track of and deploy.

RECOMMENDATIONS

E

"Harden" system configuration by making appropriate changes to settings instead of relying on default settings to be sufficiently secure. **Some common examples are:**

i. Firewall configuration: Block all traffic by default and allow only specific traffic to identified services. Examples include allowing only permitted types of network traffic to pass through, such as blocking Remote Deskop Protocol (RDP) traffic if not needed, or only allowing external access to certain administrative services from a selected whitelist of IP addresses.

ii. Web server configuration: Turn off services that are not in use (e.g. disabling of directory listing, disabling of banner display, restricting IP access, and turning off unused modules).

F

Automate build and deployment processes to minimise manual steps and hence reduce the likelihood of human error. For example, execute predefined scripts instead of manually typing out commands each time a new build of an application is required. This can eliminate errors in typing and decrease the possibility of accidentally leaving out certain commands or deploying the new build to the wrong environment (e.g. deploying a test build to the production environment).

G H

Manage configuration settings in a systematic way:

i. **Document baseline configuration settings.** Update and review this baseline periodically. This provides a reference point for configuration change and restoration, and is also useful should there be a need to rebuild the server.

ii. **Establish procedures** for configuration management, code management and code deployment. This ensures a systematic way to manage configuration changes.

iii. **When troubleshooting, note down any configuration changes made.** This is useful for review, to update the baseline, or in case of a need to revert to previous settings.

iv. **Conduct security review and testing regularly** to ensure actual configuration settings in use correspond to documented values.

MALWARE AND PHISHING

With employees having unrestricted access to the Internet, phishing email attacks are often used to trick them into revealing their login credentials or other sensitive information, or downloading attachments containing malware.

CASE EXAMPLES

I

Organisation I's employee clicked on a malicious link in a phishing email, leading to malware infection of the email software. The malware retrieved emails containing customers' personal data from the employee's email account and forwarded the emails to external parties.

J

After **Organisation J**'s ICT staff left the company, system patching was not performed for 12 months, nor were security reviews and checks conducted. The system was subsequently infected by ransomware, and Organisation J could no longer access the personal data in their ICT system.

K

Organisation K had installed anti-malware software in the computers of most of its employees except some who were not working in the main office. When those without anti-malware software fell victim to phishing emails, their computers became infected with malware, resulting in exfiltration of personal data.

The letters marked at each recommendation indicate the case examples where it may have helped the respective organisation prevent the data breach incident.

OBSERVATIONS

- **Phishing is closely linked to malware and ransomware**, as email recipients are often enticed to click on malicious URLs.
- **Threat actors are quick to take advantage of trends and situations** to create phishing emails that users are more likely to fall for (e.g. emails related to the COVID-19 pandemic or to new government initiatives).

RECOMMENDATIONS

I

Conduct regular phishing simulation exercises to train your employees to be alert. This complements any existing employee education. Organisations should put in place processes to regularly monitor the awareness level of their employees.

I K

Educate employees and regularly remind them to be alert to phishing and other forms of social engineering. Even with the most advanced security measures in place, an employee's careless actions can still provide an entry point for cyber-attacks.

I K

Consider restricting Internet access (e.g. via blacklisting or whitelisting), especially where there is direct access from endpoints to large amounts of personal or sensitive data. When these endpoints, such as employee laptops, are compromised, there is a higher risk of personal data being exfiltrated.

I J K

Install endpoint security solutions as defence against malware. It is crucial that these software be kept updated. Some use both antivirus and anti-malware solutions in tandem, with the former (typically signature-based) as defence against known threats, and the latter (typically behavioural-based) against unknown threats. Organisations should keep proper records of the endpoint security solutions and versions installed on all their systems and their employees' computers.

J K

Ensure personal data in your organisation's possession is automatically and regularly backed up. The aim of ransomware is to disrupt business operations by denying access to operational data. Having regular backups can be an effective recovery plan. For better security, backups should be offline, and stored off-site. It is also important to verify that the backup data can be restored.

SECURITY AND RESPONSIBILITY ISSUES

The security of an ICT system needs to be taken into consideration during the design and development phases, and thereafter as part of system maintenance as well. The responsibility of taking care of the ICT system's security needs has to be assigned to someone.

CASE EXAMPLES

L Organisation L placed a text file containing personal data in a test environment for testing purposes but forgot to remove it after testing. Unknown to Organisation L, the test server was publicly accessible and the text file was indexed by search engines. This resulted in the personal data being made easily searchable on the Internet.

M Organisation M contracted a vendor to implement its online resource booking system. After the system was implemented, Organisation M continued to pay the vendor for system maintenance. However, the scope of work only included fixing issues reported by Organisation M. The vendor did not perform any maintenance in terms of ICT security for the system it developed 5 to 10 years ago. Over time, the system's code became increasingly vulnerable as new exploits were developed, and personal data stored in Organisation M's system was found to be easily accessible from the Internet.

N Organisation N's ICT system generated documents containing personal data. These documents were meant for internal use only. Although Organisation N expected its vendor to protect the personal data in possession, it did not instruct the vendor to do so. As a result, the documents were generated in a publicly accessible web folder and were not protected by access control or even a password; the documents were subsequently accessed on the Internet by unintended parties.

The letters marked at each recommendation indicate the case examples where it may have helped the respective organisation prevent the data breach incident.

OBSERVATIONS

- Without proper maintenance, systems generally become more vulnerable over time. **Hence, ICT security needs to be part of the scope of ongoing system maintenance.**
- Out of convenience, many organisations use production data for system testing in their test environments. But as test environments tend to be much less secured, there is a high **risk of data breach in a test environment.**

RECOMMENDATIONS

L

Create synthetic data (i.e. fake personal data or data anonymised from real data) **for development and testing purposes in non-production environments** instead of using real data. Synthetic data can be generated either from scratch using commercial tools[1] or by anonymising production data.[2]

L N

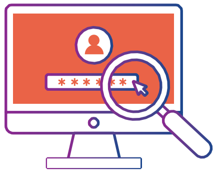
Protect personal data through access control. Without proper access control mechanisms (e.g. requiring user login), any webpage or document in a publicly accessible website/web application can be indexed by search engines and appear in search results, which means it can be easily found by anyone.

M N

Establish clear responsibility for ICT security to the assigned person(s) or team. Examples of ICT security during maintenance include system patching, security scans, and checking of log files for anomalies. This can be performed by either your organisation, a qualified vendor, or with a joint/split arrangement. Where it is to be performed by a vendor, state the scope of work and areas of responsibilities clearly in the contract.

[1] Examples include IBM Infosphere Optim Test Data Management, Informatica Test Data Management Tool and CA Technologies Datamaker.

[2] You may refer to PDPC's *Guide to Basic Data Anonymisation Techniques* for more information.



ACCOUNTS AND PASSWORDS

When accounts and passwords fall into the wrong hands, they can enable unauthorised access to ICT systems without requiring sophisticated attacks at the server end. This can happen, for example, through the use of weak passwords which can easily be guessed by hackers. Hence, accounts and passwords need to be managed securely.

CASE EXAMPLES

O **Organisation O** had an unused account in an internal ICT system. Despite regular user account housekeeping, this unused account was not detected nor removed because Organisation O only checked user accounts used by human users, but ignored all "system" accounts, including the unused account. This, together with a weak password of "admin" (a short, simple and common password) and the relocation of the ICT system to be Internet-facing, resulted in the account being compromised by a brute force attack and the personal data to be accessible to unauthorised parties.

P **Organisation P**'s IT vendor placed their database login credentials in clear text in an "env" file (env files contain configuration values) which was unprotected and publicly accessible. Personal data was subsequently exfiltrated from the database.

Q **Organisation Q**'s Electronic Direct Mail (EDM) software's administrator password was relatively simple and guessable, and it had not been changed for about 10 years. Moreover, it was also shared among multiple users. The administrative account was hacked and spam emails were sent to Organisation Q's customers.

R A hacker gained access to **Organisation R**'s database through the phpMyAdmin database tool. Even though the administrative account had access to the full set of personal data in the database, it had a weak and easily guessable password of "12345".

The letters marked at each recommendation indicate the case examples where it may have helped the respective organisation prevent the data breach incident.

OBSERVATIONS

- In some cases, passwords were a **default value**, or **weak** and easily guessable. This made the accounts vulnerable to brute force and dictionary attacks.
- **Credential stuffing** is another form of attack where a (usually large) set of stolen credentials is used to gain unauthorised access to accounts through automated application.
- In some of the cases, **users kept their passwords in clear text in publicly accessible web folders**, presumably for personal reference so that they did not have to remember the password. This is the digital equivalent of writing a password on a Post-it note and displaying it prominently – a risky practice.
- **Administrative accounts** (or privileged accounts) require even more caution as they may provide access to servers or databases containing personal data.

RECOMMENDATIONS

O

Review user accounts periodically and remove accounts that are no longer needed.

P

Ensure that passwords are not exposed in code or configuration files. State this clearly in your ICT policy and ensure that your team or vendors are aware. Watch out for such risks during security review and scanning.

O Q R

Minimise risk of brute force attacks. Allowing unlimited failed login attempts makes a system more vulnerable to brute force attacks when a hacker can make infinite login attempts (e.g. by using all possible combinations of alphanumeric characters or a list of commonly used passwords). Ways to prevent or slow down brute force attacks include locking the user account upon a pre-defined number of failed login attempts, implementing a delay after a failed login attempt, or using CAPTCHAs.

O P Q R

Adopt and implement a strong password policy. Organisations can adopt the following good practices for passwords:

i. Enforce a **password history policy** to ensure that employees do not reuse their previous passwords.
ii. Encourage users to use **passphrases** such as "Iwant2I@se10kg", which may be long and complex, yet easy to remember.
iii. **Discourage users from using the same passwords across different systems.**

O P Q R

Have stronger requirements for some administrative accounts (e.g. a complex password or 2-Factor Authentication (2FA) / Multi-Factor Authentication (MFA)). With 2FA/MFA in place, access to administrative accounts would involve additional round (s) of authentication, such as a temporary code sent securely to the administrator's mobile phone. Hence, the use of a stolen password alone will not be enough to breach an account. This is important for administrative accounts to systems that hold large volumes of personal data, or personal data of a confidential or sensitive nature (e.g. financial or health records), where a breach of such data could result in adverse impact to the affected individuals.

ACKNOWLEDGEMENT

The Personal Data Protection Commission and Info-communications Media Development Authority express their sincere appreciation to the following organisations for their valuable feedback in the development of this publication:

- ISACA Singapore Chapter - Data Privacy SIG
- SGTech

BROUGHT TO YOU BY

pdpc
PERSONAL DATA
PROTECTION COMMISSION
SINGAPORE

This publication gives a general overview of ways to prevent data breaches by describing common gaps in ICT system management and processes, and corresponding examples and recommendations for prevention of data breach incidents. The contents herein are not intended to be an authoritative statement of the law or a substitute for legal or other professional advice. The PDPC and its members, officers, employees and delegates shall not be responsible for any inaccuracy, error or omission in this publication or liable for any damage or loss of any kind as a result of any use of or reliance on this publication.