



OWASP

Open Web Application
Security Project

Lessons Learnt from Past Data Breaches in Singapore & OWASP API Security Artefacts

Wong Onn Chee

OWASP Singapore Chapter Co-Lead

OWASP : Core Mission

- The Open Web Application Security Project (OWASP) is a 501c3 not-for-profit also registered in Europe as a worldwide charitable organization focused on improving the security of software.
- Our mission is to make application security visible, so that people and organizations can make informed decisions about true application security risks.
- Everyone is welcomed to participate in OWASP and all of our materials are available under free and open software licenses.

Full Disclosure

Rajah & Tann Cybersecurity, together with our parent companies, was selected by PDPC on 23 Nov 2020 as one of the 4 vendors to support them in cybersecurity investigation of private sector organisations suspected of violating PDPA.

As of 14 June 2022, RTC became a licensed Penetration Testing Service Provider under Singapore's Cybersecurity Act.

Full Disclosure

Case studies in this slide deck are from published PDPC
Decisions which are publicly available from
<https://www.pdpc.gov.sg>

No OSA-protected content is shared in this presentation.

Agenda

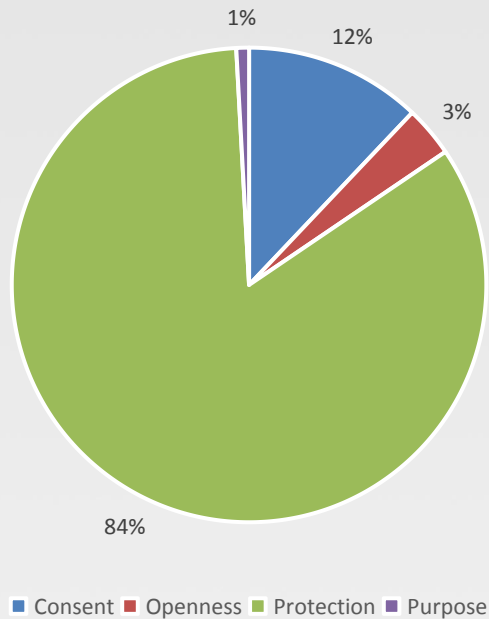
- A look at PDPC Decisions till 24 Sep 2022
- Case Study 1: PDPC Decision on ST Logistics (Ransomware)
- Case Study 2: PDPC Decision on Commeasure (API key)
- Case Study 3: PDPC Decision on Singtel (Mobile) (Insecure API design)
- Review of common causes of data breaches
- OWASP API security artefacts

PDPC Decisions

1. 173 Decisions for confirmed breaches of PDPA from 2016 till Sep 2022
2. Total penalties: **S\$3.073m**
3. Personal data (non-dedup) records: **16.83m**
4. Excluding Singhealth/IHIS case, the next Top 3 offenders by penalties are:
 1. Secure Solutions Group - HSA vendor - leaked blood donor PD - S\$120k - 2020
 2. Ninja Logistics - S\$90K - 2019
 3. Commeasure (reddoorz.com) - S\$74k - 2021
5. All are Breaches of Protection.
6. In fact, all Top 10 breaches by quantum of penalties are Breaches of Protection.

84% of PDPC Decisions are for Breaches of Protection

Enforced Cases by Offence Type



97% of PDPA penalties are for Breaches of Protection



IT firms / vendors account for ~S\$1.156m of penalties



Leakages from web / cloud services are the most common causes



Case Study 1: ST Logistics (Ransomware)

1. Decision published on October 2020.
2. ST Logistics (“STL”) provides logistical services to Singapore’s government and defence sectors, as well as commercial sectors.
3. On 16 Dec 2019, STL notified the PDPC that it had detected an Emotet malware (“Emotet”) in their network which had infected 6 of its users’ laptops (including 4 laptops containing personal data) potentially affecting up to 2,400 individuals in MINDEF and SAF.

Case Study 1: ST Logistics (Ransomware)

4. On 2 Oct 2019, STL's staff received phishing emails
 1. From email addresses with text "Stlogs" in the sender name field (e.g. "Account Executive (Stlogs)" and "Assistant General Manager (Stlogs)")
 2. Each email contained an attachment with the file extension "doc". A total of 13 users from STL opened the malicious attachment ("**Affected Users**")
 3. 7 Affected Users had the Palo Alto Traps software ("**Traps Software**"), an advanced endpoint protection solution, installed in their laptops and were therefore protected from Emotet.
 4. Remaining 6 Affected Users ("**Infected Users**") did not have Traps Software installed in their laptops.

Case Study 1: ST Logistics (Ransomware)

5. Emotet harvested the emails in the Infected Users' accounts.
6. Created approximately 100 new phishing emails, and sent these new phishing emails on 3 Oct 2019.
7. Those new phishing emails quoted the bodies of real emails in the email accounts of the Infected Users.

Case Study 1: ST Logistics (Ransomware)

8. Unencrypted files containing personal data were stored in 4 of the Infected Users' laptops. The files were offline working copy files and contained personal data relating to a total of 2,400 MINDEF and SAF personnel:
- (a) Names;
 - (b) Mailing addresses;
 - (c) Email addresses;
 - (d) Telephone numbers; and
 - (e) NRIC numbers (1,320 full NRIC numbers and 1,080 masked (last 3 digits and checksum) NRIC numbers).

Case Study 1: ST Logistics (Ransomware)

9. “The Commission’s investigations revealed that the Organisation failed to conduct periodic security reviews to detect vulnerabilities in its IT systems.”
10. “As stated in the Commission’s previous decisions, organisations are expected to conduct periodic security reviews of its IT systems.

The comprehensiveness of such security reviews should be scoped based on the organisation’s assessment of its data protection needs, and be conducted to a reasonable standard.”

Case Study 1: ST Logistics (Ransomware)

11. “In the present case, a reasonably conducted security review should have included (i) verifying complete installation and proper configuration of the security software on all of the Organisation’s users’ laptops; and (ii) checking that the security software is updated;”
12. “The anti-virus software installed on users’ laptops was not updated because they had not been properly configured to receive updates. This security gap affected all of the Infected Users, whose laptops were not so configured.”
13. STL was found in breach of Protection Obligation

Lessons from ST Logistics (Ransomware)

1. Regular security reviews, including configuration audit, should be performed.

- (a) **2 rounds of Security Architecture Review against CIS controls (start of project)**
 - please refer to the PDPC decisions against Honestbee Pte Ltd, Funding Societies Pte. Ltd., ComGateway (S) Pte. Ltd., JP Pepperdine Group Pte. Ltd., The Cellar Door Pte. Ltd., Smiling Orchid (S) Pte Ltd, and Fu Kwee Kitchen Catering Service.
- (b) **1 round of Security Risk Assessment (start of project)**
 - please refer to the PDPC decisions against EU Holidays Pte. Ltd., Funding Societies Pte. Ltd., ComGateway (S) Pte. Ltd., The Cellar Door Pte. Ltd., and Fu Kwee Kitchen Catering Service.
- (c) **2 rounds of Vulnerability Assessment (end of project)**
 - please refer to the PDPC decisions against The Future of Cooking Pte. Ltd., Honestbee Pte Ltd, Funding Societies Pte. Ltd., Bud Cosmetics, JP Pepperdine Group Pte. Ltd., and K Box Entertainment Group Pte. Ltd.
- (d) **2 rounds of Configuration Security Audit (end of project)**
 - please refer to the PDPC decisions against Honestbee Pte Ltd, COURTS (Singapore) Pte Ltd., Funding Societies Pte. Ltd., Bud Cosmetics, The Cellar Door Pte. Ltd., and K Box Entertainment Group Pte. Ltd.
- (e) **2 rounds of Web Application Penetration Testing of web/cloud services (end of project)**
 - please refer to the PDPC decisions against The Future of Cooking Pte. Ltd., COURTS (Singapore) Pte Ltd., Funding Societies Pte. Ltd., Bud Cosmetics, JP Pepperdine Group Pte. Ltd., The Cellar Door Pte. Ltd., and K Box Entertainment Group Pte. Ltd.
- (f) **2 rounds of Source Code Security Review of applications (end of project)**
 - please refer to the PDPC decisions against Funding Societies Pte. Ltd., Singtel, and GMM Technoworld Pte. Ltd.

Lessons from ST Logistics (Ransomware)

2. Up-to-date and comprehensive IT asset management would have flagged out the inconsistent security implementation to the Infected Users' endpoints.

Asset management should include software assets and libraries, i.e. SBOM.

Remember Log4J/Log4Shell.

Case Study 2: Commeasure (API Key)

1. Decision published on September 2021.
2. Commeasure Pte. Ltd. (“**CPL**”) operates a hotel booking platform *www.reddoorz.com* which serves customers in the Southeast Asian region, such as Indonesia, Singapore, Philippines, Vietnam and Thailand.
3. The Singapore office is primarily engaged in sales, finance and administrative activities, while all IT functions (including the management of the affected application package in this case) were managed by the Organisation’s subsidiary company, Commeasure Solutions India Pvt Ltd (“**CPL India**”)

Case Study 2: Commeasure (API Key)

1. CPL's database containing 5,892,843 customer records which included:
 - a) the customer's name,
 - b) contact number,
 - c) email address,
 - d) date of birth,
 - e) a hashed password (encrypted with one-way BCrypt hash algorithm) used by the customer to access their "RedDoorz" account and their booking informationwas accessed and exfiltrated by unknown TA(s).

Case Study 2: Commeasure (API Key)

1. Threat Actor (“**TA**”) had most likely gained access and exfiltrated the Organisation’s database of customer records hosted in an Amazon RDS cloud database, after they obtained an Amazon Web Services (“**AWS**”) access key.
2. The AWS access key was embedded within an Android application package (“**the affected APK**”) publicly available for download from the Google Play Store.
3. Even though the AWS access key had access to a “live” or production database, the AWS access key was embedded in the APK, and erroneously marked as a “test” key by the then-developers.

Case Study 2: Commeasure (API Key)

1. With the exception of one of the CPL's co-founders and Chief Technology Officer, all the developers have since left the CPL.
2. Even though CPL regarded this APK as “defunct”, the APK remained publicly available for download on the Google Play Store until CPL became aware of the Incident and removed the affected APK.
3. Even though the Organisation had engaged a cybersecurity company to conduct a security review and penetration testing sometime from September 2019 to December 2019, it was not within the scope of the security review or penetration tests. Consequently, the vulnerability was left undetected and exposed until the Organisation found out about the Incident.

Case Study 2: Commeasure (API Key)

1. CPL used “Proguard” on its current Android apps to prevent reverse engineering of APKs, which may have prevented the unknown threat actors from retrieving the AWS access key, the Organisation failed to review and deploy “Proguard” on the affected APK which it regarded as “defunct”.

Case Study 2: Commeasure (API Key)

1. “The data breach occurred because the Organisation embedded the AWS access key, which allowed access to the “live” or production database, in the APK. The root cause was therefore in the application, which was clearly within the Organisation’s responsibility...
2. ...AWS also cautioned users not to “embed access keys directly into code”, which was exactly what the Organisation had done in the present case. We therefore find the Organisation in breach of section 24 of the PDPA for reflecting the AWS access key in the affected APK.”

Case Study 2: Commeasure (API Key)

1. “In the course of investigations, the Organisation explained that its failure to implement sufficiently robust processes to manage its inventory of infrastructure access keys was attributable to the high turnover of its employees from the time of its inception to the discovery of the Incident. This explanation is unacceptable, however sympathetic one might be to the human resource issues that the Organisation had to manage...
2. ...The Commission reiterates that it is necessary for an organisation to “[c]onduct regular ICT security audits, scans and tests to detect vulnerabilities.”

Case Study 2: Commeasure (API Key)

1. “The Organisation’s failure to include the affected APK and the AWS access key within the scope of the security review arose because of the Organisation’s negligence to include them in its inventory of IT assets in production after the Organisation had wrongly labelled the affected APK as “defunct” and the AWS access key as a “test” key.”
2. Accordingly, we are not satisfied that the IT security reviews that the Organisation conducted were sufficiently rigorous, and met the standard required under section 24 of the PDPA.”

Case Study 2: Commeasure (API Key)

1. Up-to-date and comprehensive IT software asset management would have prevented the omission of the “defunct” Affected APK.
2. Proper API key management would have prevented use of defunct API key.

Use Amazon Cognito and limit the rights of the IAM identity used by your mobile app which is accessible to the public. There is no reason why the identity in your mobile app needs admin access to your cloud DB, storage and virtual instances.

Case Study 2: Commeasure (API Key)

2. Besides not embedding API keys in source codes, please **DO NOT** keep credentials or API keys in cloud server environment variables too.

Regardless of cloud or on-prem, it is a **cardinal sin** to keep credentials or API keys in **plaintext storage**, esp in environment variables.

Referring to <https://docs.aws.amazon.com/general/latest/gr/aws-access-keys-best-practices.html>, **do not accept everything** in this page.

From PDPC Decision on MyRepublic (15 Sep 2022): “Thereafter, if the php-info URL is accessed, the browser will display the Portal’s ***operating system environment variable values***. These values included the ***Access Key***, which was used by the Portal to access and transfer documents submitted by customers through the Portal to the Bucket.”

Case Study 2: Commeasure (API Key)

When you use access keys, observe these precautions:

- **Don't embed access keys directly into code.** When you use [AWS SDKs](#) and the [AWS Command Line Tools](#), you can insert access keys in known locations so that you don't have to keep them in code.

Put access keys in one of the following locations:

- **The AWS credentials file.** The AWS SDKs and AWS CLI automatically use the credentials that you store in the AWS credentials file.

For information about using the AWS credentials file, see the documentation for your SDK. Examples include [Set AWS Credentials and Region](#) in the *AWS SDK for Java Developer Guide* and [Configuration and credential files](#) in the *AWS Command Line Interface User Guide*.

To store credentials for the AWS SDK for .NET and the AWS Tools for Windows PowerShell, we recommend that you use the SDK Store. For more information, see [Using the SDK Store](#) in the *AWS SDK for .NET Developer Guide*.

- **Environment variables.** On a multitenant system, choose user environment variables, not system environment variables.

For more information about using environment variables to store credentials, see [Environment Variables](#) in the *AWS Command Line Interface User Guide*.

Source:
<https://docs.aws.amazon.com/general/latest/gr/aws-access-keys-best-practices.html>

Case Study 2: Commeasure (API Key)

INFOTECT SECURITY						
Home Incidents Content Type Whitelist Users Websites System Audit Logs superadmin						
Incidents						
Q Search Reset Export						
ID	Requestor IP	Protocol	Hostname	Path	Content Type	Occurred On
11074967057182943	88.214.43.215	http	66.96.198.108	/.aws/config	Exception	17 Sep 2022 7:17 PM SGT
11074966302291578	88.214.43.215	http	66.96.198.108	/.aws/config	Exception	17 Sep 2022 7:17 PM SGT
11074965285593346	88.214.43.215	http	66.96.198.108	/.aws/credentials	Exception	17 Sep 2022 7:17 PM SGT
11074964560533404	88.214.43.215	http	66.96.198.108	/.aws/credentials	Exception	17 Sep 2022 7:17 PM SGT
10691855134831590	88.214.43.215	http	66.96.198.108	/.aws/config	Exception	13 Sep 2022 8:52 AM SGT
10691854160510117	88.214.43.215	http	66.96.198.108	/.aws/config	Exception	13 Sep 2022 8:52 AM SGT
106918546101117	88.214.43.215	http	66.96.198.108	/.aws/config	Exception	13 Sep 2022 8:52 AM SGT
106918531601117	88.214.43.215	http	66.96.198.108	/.aws/config	Exception	13 Sep 2022 8:52 AM SGT
1013263596	88.214.43.215	http	66.96.198.108	/.aws/config	Exception	13 Sep 2022 8:52 AM SGT
1013263504	88.214.43.215	http	66.96.198.108	/.aws/config	Exception	13 Sep 2022 8:52 AM SGT

Threat Actor(s) also can read
<https://docs.aws.amazon.com/general/latest/gr/aws-access-keys-best-practices.html>

and catch you if you follow instructions blindly.



Case Study 2: Commeasure (API Key)


3. Red-teaming (beyond PT), which should include reconnaissance of an organisation's publicly available resources, might have helped.

For red-teaming of all tech firms, especially start-ups, used public resources, such as those from Github, mobile app stores, Slack, MongoDB Atlas, Bitbucket, Confluence and etc, must be included.

Any red teaming firm which does not recommend such reconnaissance coverage should be reported to certification / licensing authorities for professional negligence.

If the red teaming firm had recommended, but you refused due to budget constraints, then you will bear the full brunt of PDPC decisions.

Case Study 3: Singtel (Mobile) (Insecure API Design)

 PERSONAL DATA
PROTECTION COMMISSION
SINGAPORE

Who We Are

Overview of PDPA

Enforcement
Decisions

News & Events

Home / All Commission's Decisions / Breach of the Protection Obligation by Singtel

Breach of the Protection Obligation by Singtel

04 Nov 2019

A financial penalty of \$25,000 was imposed on Singtel for failing to put in place reasonable security arrangements to protect the personal data of users on its My Singtel mobile application.

Click [here](#) to find out more.

Tags: [Protection](#), [Financial Penalty](#)

Case Study 3: Singtel (Mobile) (Insecure API Design)

Singapore Telecommunications Limited

[2019] SGPDPC 36

10 During the investigation, the Organisation admitted that the Data Breach was caused by a design issue in the API – the application input⁴ was not validated against the login credential used to access the Mobile App before performing the requested operation (the “**Direct Object Reference Vulnerability**”). Because all request parameters sent by the Mobile App to the Organisation’s server during a valid login session were assumed to be valid, once a user was legitimately authenticated to initiate a valid login session on the device (via the MSISDN, OTP or OnePass login methods), the user would be able to intercept and change the field parameters in the API requests between the Mobile App and the server. Notwithstanding, the Organisation asserted that such an action was “not something that a normal user of the App would attempt” and the attacker must be “technically competent” as the changing of the parameters could only be performed on a workstation.

Common mistakes in data breaches

1

WEAK IDENTITY & ACCESS MGT

1. Victims did not have MFA enabled or did not know their credentials or API keys were leaked.
2. No geolocation access restriction, esp for administrator access.
3. Insecure design of access mgt, e.g. IDOR

2

WEAK PATCHING

1. Victims did not have formal patch management standards, such as patching SLA. (Combined with weak vendor mgt)
2. Unpatched systems, such as FW, Exchange Servers, were common conduits through which threat actors had intruded into victims.

3

NO REGULAR EVALUATION

1. Most victims did not perform regular full security review, esp configuration audit and compliance review against any internal policies or standards.

Common mistakes in data breaches

4

WEAK VENDOR MGT

1. Victims did not have good governance of their outsourced IT vendors, especially SaaS vendors. Not all IT vendors are security minded.
2. Contract management was a common weak spot.

5

WEAK AUDIT LOGGING

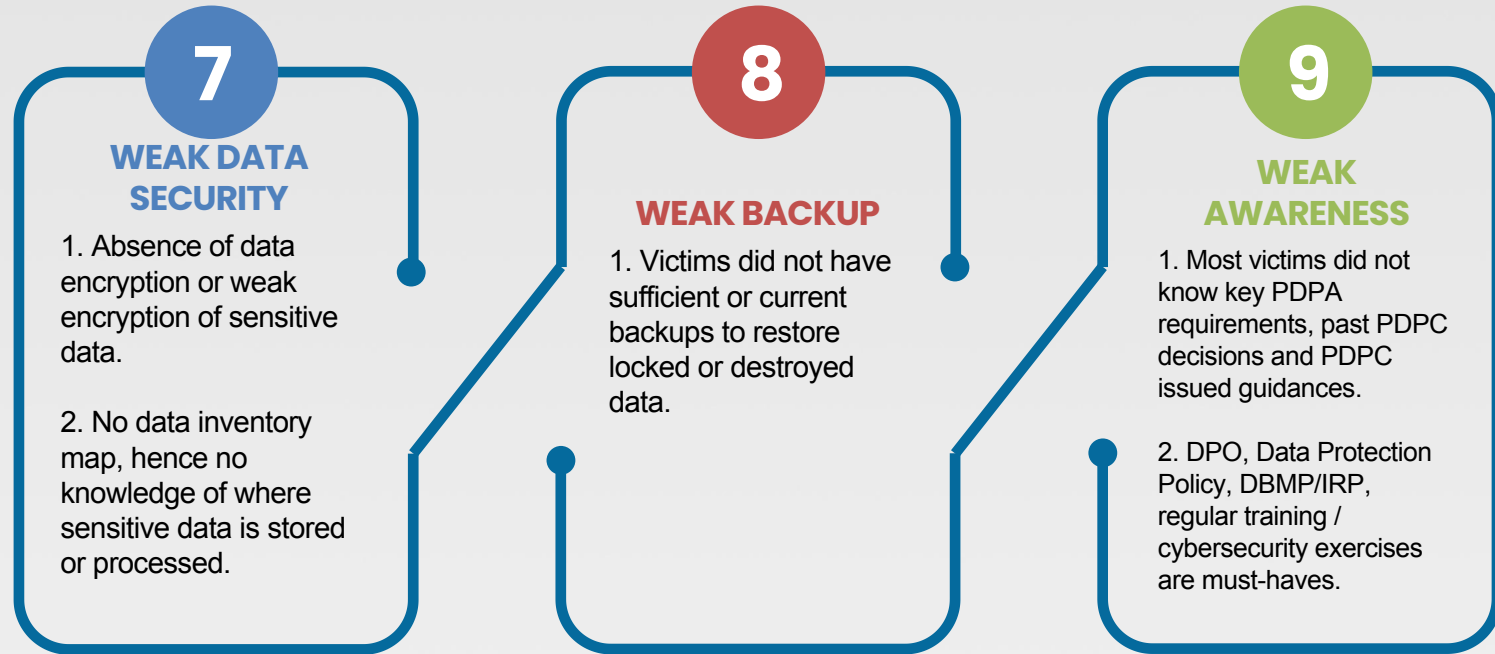
1. Victims did not have sufficient logs. This was often due to insufficient logging level or insufficient retention.
2. Problem is more endemic in cloud environment.

6

NO OUTBOUND MONITORING

1. Most victims did not have outbound monitoring capabilities to detect unauthorised disclosure or leakage of sensitive data from their public-facing web or cloud services.

Common mistakes in data breaches



Don't forget to issue cable locks to your laptop users.

Protection of personal data

24. An organisation must protect personal data in its possession or under its control by making reasonable security arrangements to prevent —

- (a) unauthorised access, collection, use, disclosure, copying, modification or disposal, or similar risks; and
- (b) the loss of any storage medium or device on which personal data is stored.

[40/2020]

**You are still guilty
of s.24(b) violation
even if your storage
device is encrypted.**

Source:
<https://sso.agc.gov.sg/Act/POPA2012?Provs=P16-#pr24->

Public officers are now subjected to the penalties as private sector personnel

Unauthorised disclosure of personal data

Private Sector

48D.—(1) If —

- (a) an individual discloses, or the individual's conduct causes disclosure of, personal data in the possession or under the control of an organisation or a public agency to another person;
- (b) the disclosure is not authorised by the organisation;
- (c) the individual does so —
 - (i) knowing that the disclosure is not authorised, the case may be; or
 - (ii) reckless as to whether the disclosure is or is not authorised by the organisation or public agency, as the case may be,

the individual shall be guilty of an offence and shall be liable on conviction to a fine not exceeding 2 years or to both.

Source:
<https://sso.agc.gov.sg/Act/PSGA2018?ProvlDs=P12-#pr7->

Source:
<https://sso.agc.gov.sg/Act/PDPA2012?ProvlDs=P19B-#pr48D->

Unauthorised disclosure and improper use of information

Public Sector

7.—(1) If —

- (a) an individual discloses, or the individual's conduct causes disclosure of, information under the control of a Singapore public sector agency to another person (whether or not a Singapore public sector agency);
- (b) the disclosure is not authorised by any data sharing direction given to the Singapore public sector agency;
- (c) the individual is a relevant public official of the Singapore public sector agency at the time of the disclosure; and
- (d) the individual does so —
 - (i) knowing that the disclosure is not in accordance with that direction; or
 - (ii) reckless as to whether the disclosure is or is not in accordance with that direction,

the individual shall be guilty of an offence and shall be liable on conviction to a fine not exceeding \$5,000 or to imprisonment for a term not exceeding 2 years or to both.



OWASP API Security Artefacts

- OWASP API Security Top 10 2019 (<https://owasp.org/www-project-api-security/>)
- OWASP Enterprise Security API (ESAPI) (<https://owasp.org/www-project-enterprise-security-api/>)

OWASP API Security Top 10 2019

API1:2019 Broken Object Level Authorization	API6:2019 Mass Assignment
API2:2019 Broken User Authentication	API7:2019 Security Misconfiguration
API3:2019 Excessive Data Exposure	API8:2019 Injection
API4:2019 Lack of Resources & Rate Limiting	API9:2019 Improper Assets Management
API5:2019 Broken Function Level Authorization	API10:2019 Insufficient Logging & Monitoring

OWASP ESAPI

- ESAPI (The OWASP Enterprise Security API) is
 - a free, open source, web application security control library that consists of a set of security control interfaces.
 - makes it easier for programmers to write lower-risk applications
 - currently available for Java



OWASP

Open Web Application
Security Project

Lessons Learnt from Past Data Breaches in Singapore & OWASP API Security Artefacts

Thank you for your attention.